Reverse Engineering ℓ_p attacks: A block-sparse optimization approach with recovery guarantees

Darshan Thaker [*1] **Paris Giampouras** [*1] **René Vidal** [1]

Abstract

Deep neural network-based classifiers have been shown to be vulnerable to imperceptible perturbations to their input, such as ℓ_p-bounded norm adversarial attacks. This has motivated the development of many defense methods, which are then broken by new attacks, and so on. This paper focuses on a different but related problem of reverse engineering adversarial attacks. Specifically, given an attacked signal, we study conditions under which one can determine the type of attack (ℓ_1, ℓ_2 or ℓ_∞) and recover the clean signal. We pose this problem as a block-sparse recovery problem, where both the signal and the attack are assumed to lie in a union of subspaces that includes one subspace per class and one subspace per attack type. We derive geometric conditions on the subspaces under which any attacked signal can be decomposed as the sum of a clean signal plus an attack. In addition, by determining the subspaces that contain the signal and the attack, we can also classify the signal and determine the attack type. Experiments on digit and face classification demonstrate the effectiveness of the proposed approach.

1. Introduction

Deep neural network based classifiers have been shown to be vulnerable to imperceptible perturbations to their inputs, which can cause the classifier to incorrectly classify a datapoint. (Biggio et al., 2013; Szegedy et al., 2014). Examples of such adversarial attacks include the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) and the Projected Gradient Method (PGD) (Madry et al., 2018a), where small additive perturbations are made to the input that are bounded in ℓ_p norm and designed to maximize the loss of the classifier. In response to these attacks, many defense methods have been developed, including Randomized Smoothing (Cohen et al., 2019). However, such defenses have been broken by new attacks, and so on, leading to a cat and mouse game between new attacks (Athalye et al., 2018a;b; Carlini & Wagner, 2017a; Uesato et al., 2018; Athalye & Carlini, 2018) and new defenses (Madry et al., 2018b; Samangouei et al., 2018; Zhang et al., 2019; Papernot et al., 2016; Kurakin et al., 2016; Miyato et al., 2017; Zheng et al., 2016).

This paper focuses on the less well studied problem of *reverse engineering adversarial attacks*. Specifically, given a corrupted signal $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$, where $\mathbf{x}$ is a "clean" signal and $\boldsymbol{\delta}$ is an ℓ_p-norm bounded attack, the goal is to determine the attack type (ℓ_1, ℓ_2 or ℓ_∞) as well as the original signal $\mathbf{x}$.

Challenges. In principle, this problem might seem impossible to solve since there could be many pairs $(\mathbf{x}, \boldsymbol{\delta})$ that yield the same $\mathbf{x}'$. A key challenge is hence to derive conditions under which this problem is well posed. We propose to address this challenge by leveraging results from the sparse recovery literature, which show that one can perfectly recover a signal $\mathbf{x}$ from a corrupted version $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}_0$ when both $\mathbf{x}$ and $\boldsymbol{\delta}_0$ are sparse in a meaningful basis. Specifically, it is shown in (Wright & Ma, 2010) that if $\mathbf{x}$ is sparse with respect to some signal dictionary $\mathbf{D}_s$, i.e., if $\mathbf{x} = \mathbf{D}_s \mathbf{c}_s$ for a sparse vector $\mathbf{c}_s$, and $\boldsymbol{\delta}_0$ is also sufficiently sparse, then the solution $(\mathbf{c}^*, \boldsymbol{\delta}^*)$ to the convex problem

$$\min_{\mathbf{c}} \|\mathbf{c}\|_1 + \|\boldsymbol{\delta}\|_1 \quad \text{s.t.} \quad \mathbf{x}' = \mathbf{D}_s \mathbf{c} + \boldsymbol{\delta} \tag{1}$$

is such that $\mathbf{c}^* = \mathbf{c}_s$ and $\boldsymbol{\delta}^* = \boldsymbol{\delta}_0$. In other words, one can perfectly recover the clean signal as $\mathbf{x} = \mathbf{D}_s \mathbf{c}^* = \mathbf{D}_s \mathbf{c}_s$ and the corruption $\boldsymbol{\delta}_0$ by solving the convex problem in (1).

Unfortunately, these classical results from sparse recovery are not directly applicable to the problem of reverse engineering adversarial attacks due to several challenges:

1. An attack $\boldsymbol{\delta}$ may not be sparse. Indeed, $\boldsymbol{\delta}$ is usually assumed to be bounded in ℓ_p norm, where $p = 1, 2, \infty$. While results from sparse recovery can be extended to bounded ℓ_2 errors, e.g. (Candès et al., 2006) considers the case where $\boldsymbol{\delta}$ is ℓ_2-bounded, such results only

[*]Equal contribution [1]Mathematical Institute for Data Science, Johns Hopkins University, Baltimore, MD USA. Correspondence to: Darshan Thaker <dbthaker@jhu.edu>, Paris Giampouras <parisg@jhu.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

guarantee stable recovery, instead of exact recovery, of *sparse* vectors $\mathbf{c}_s$ close to $\mathbf{c}^*$.

2. One of our goals is to determine the attack type. To do so, we need to exploit the fact that $\boldsymbol{\delta}$ is not an arbitrary vector, but rather a function of the attack type, the loss, the neural network and $\mathbf{x}$ (e.g., in the PGD method $\boldsymbol{\delta}$ is the projection of the gradient of the loss with respect to $\mathbf{x}$ onto the ℓ_2 ball). The challenge is to devise an attack model that, despite these complex dependencies, is amenable to results from sparse recovery. In particular, we wish to impose structure on $\boldsymbol{\delta}$ that correlates its sparsity pattern to the attack type.

3. Another goal is to correctly classify $\mathbf{x}'$ despite the attack $\boldsymbol{\delta}$. This is at odds with most sparse recovery results, which focus on reconstruction rather than classification. The main exceptions are the sparse and block-sparse representation classifiers (Wright et al., 2009; Elhamifar & Vidal, 2012), which divide the dictionary $\mathbf{D}_s$ into blocks corresponding to different classes and exploit the sparsity pattern of $\mathbf{c}_s$ to determine the class of $\mathbf{x}$. But such classifiers are different from the neural network classifier given to us.

Paper contributions. This paper proposes a framework based on structured block-sparsity for addressing some of these challenges. Our key contributions are the following.

First, we develop a structured block-sparse model, a condition we show holds in a variety of settings, for decomposing attacked signals under three main assumptions about the signal and underlying network:

1. We assume that the signal $\mathbf{x}'$ to be classified is the sum of a clean signal $\mathbf{x}$ plus an ℓ_p-norm bounded adversarial attack $\boldsymbol{\delta}$, i.e., $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$, i.e., additive attacks.

2. We assume that the clean signal $\mathbf{x}$ is *block sparse* with respect to a dictionary of signals $\mathbf{D}_s$, i.e. $\mathbf{x} = \mathbf{D}_s\mathbf{c}_s$, where $\mathbf{D}_s$ can be decomposed into multiple blocks, each one corresponding to one class, and $\mathbf{c}_s$ is *block-sparse* i.e. $\mathbf{c}_s$ is only supported on a sparse number of blocks, but not necessarily sparse within those blocks.

3. We assume that the the ℓ_p-norm bounded adversarial attack also admits a block-sparse representation in the columnspace of a dictionary $\mathbf{D}_a$, which contains blocks corresponding to different ℓ_p bounded attacks.

Second, we study conditions under which the aforementioned assumptions are feasible. In particular, we prove that ℓ_p attacks can be expressed as a structured block-sparse combination of other attacks for general loss functions when the attacked deep classifier satisfies some local linearity assumptions (e.g. ReLU networks).

Third, to determine the attack type and reconstruct the clean signal, we solve a convex optimization problem of the form:

$$\min_{\mathbf{c}_s,\mathbf{c}_a} \quad \|\mathbf{c}_s\|_{1,2}+\|\mathbf{c}_a\|_{1,2} \quad \text{s.t.} \quad \mathbf{x}' = \mathbf{D}_s\mathbf{c}_s+\mathbf{D}_a\mathbf{c}_a. \quad (2)$$

Here, $\|\cdot\|_{1,2}$ is the ℓ_1/ℓ_2 norm that promotes structured block-sparsity on $\mathbf{c}_s$ and $\mathbf{c}_a$ exploiting the structure of $\mathbf{D}_s$ and $\mathbf{D}_a$. For this optimization problem, we derive geometric data-dependent conditions under which the attack type and the clean signal can be recovered. These conditions rely on a special covering radius of $\mathbf{D}_s$ and $\mathbf{D}_a$ and a generalization of angular distance induced by the ℓ_1/ℓ_2 norm.

Fourth, since solving (2) can be computationally expensive due to the potentially large size of dictionaries $\mathbf{D}_s$ and $\mathbf{D}_a$, we develop an efficient active set homotopy algorithm by first relaxing the constrained problem to a regularized problem instead and solving a sequence of subproblems restricted to certain blocks of $\mathbf{D}_s$ and $\mathbf{D}_a$.

Finally, we perform experiments on digit and face classification datasets to complement our theoretical results and demonstrate not only the robustness of block-sparse models on attacks arising from a union of ℓ_p perturbations, but also the effectiveness of our models in classifying the attack family.

2. Related Work

Structured representations for data classification. Sparse representation of signals has achieved great success in applications such as image classification (Yang et al., 2009b; Mairal et al., 2008), action recognition (Yang et al., 2009a; Castrodad & Sapiro, 2012), and speech recognition (Gemmeke et al., 2011; Sainath et al., 2011) (see (Wright et al., 2010; Julien Mairal & Ponce, 2012) for more examples). These works rely on the assumption that data from a specific class lie in a low-dimensional subspace spanned by training samples of the same class. Hence, correct classification of amounts to recovering the correct sparse representation of the signal on the columnspace of a certain dictionary. However, these works do not account for adversarially corrupted inputs, which pose significant challenges and are studied in this work.

Structured representations for adversarial defenses. In the adversarial learning community, denoising-based defense strategies that leverage structured data representations have been recently proposed e.g. the work of (Samangouei et al., 2018) and (Moosavi-Dezfooli et al., 2018) (see (Niu et al., 2020) for a comprehensive survey). However, these approaches do not perform attack classification, and the key advantage of our approach is joint recovery of the signal and attack. To the best of our knowledge, our work is the

first one to study this problem from a theoretical perspective. Additionally, even though our main goal is not to develop simply a stronger defense, we can compare the signal classification stage of our approach to defenses for a union of perturbation families simultaneously. Work such as (Tramer & Boneh, 2019; Maini et al., 2020b; Croce & Hein, 2019) develop adversarial training variants to tackle this problem. Our approach is distinct from adversarial training in that it requires no retraining of the neural network and can be applied post-hoc to adversarial examples.

Detection of Adversarial Attacks. There is a vast literature on the detection of adversarial attacks, which work on the problem of detecting whether any example is an adversarial example or a clean example. These methods can be categorized into unsupervised and supervised methods, e.g. (Metzen et al., 2017) and (Grosse et al., 2017). We refer the reader to (Bulusu et al., 2020) for a comprehensive survey on these methods. Our task is fundamentally different in that given an attacked image, we aim to classify the type of attack used to corrupt the image, and moreover provide theoretical guarantees under which this recovery is feasible. For this problem, (Maini et al., 2020a) provide a method to classify attack perturbations similar to our work; however, they classify between ℓ_1, ℓ_2 attacks vs. ℓ_∞ attacks only.

3. Block-sparse model of ℓ_p attacked signals

Assume we are given a deep classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the input space, $\mathcal{Y}$ is the output space and θ are the classifier weights. Assume the classifier is trained using a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$. Assume also an additive attack model $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$, where the attack $\boldsymbol{\delta}$ is a small perturbation to the input $\mathbf{x}$ that causes the classifier to make a wrong prediction, i.e., $f_\theta(\mathbf{x}') \neq f_\theta(\mathbf{x})$.

We restrict our attention to ℓ_p-norm bounded attacks, i.e. $\boldsymbol{\delta} \in \Delta_p = \{\boldsymbol{\delta} \in \mathbb{R}^n : \|\boldsymbol{\delta}\|_p \leq \epsilon\}$, which are crafted by finding a perturbation to $\mathbf{x}$ that maximizes the loss, i.e.:

$$\max_{\boldsymbol{\delta} \in \Delta_p} L(f_\theta(\mathbf{x} + \boldsymbol{\delta}), y). \tag{3}$$

Since solving this problem can be costly, a common practice is to maximize a first-order approximation of the loss. Letting $\mathbf{g} = \nabla_{\mathbf{x}} L(f_\theta(\mathbf{x}), y)$, we obtain the following gradient-based attacks for $p = 1, 2, \infty$, respectively:

$$\boldsymbol{\delta}_1 = \epsilon\, \mathbf{a}, \quad \boldsymbol{\delta}_2 = \epsilon\, \frac{\mathbf{g}}{\|\mathbf{g}\|_2}, \quad \boldsymbol{\delta}_\infty = \epsilon\, \text{sign}(\mathbf{g}), \tag{4}$$

where $\mathbf{a}$ denotes a unit norm vector where $\mathbf{a}_{i^\star} = \text{sign}(g_{i^\star})$ for $i^\star := \arg\max_i |g_i|$. Note that ℓ_p attacks depend on the gradient of the loss with respect to the classifier input, the classifier output, the value of $p \geq 1$ used in the ℓ_p-norm, and the attack strength $\epsilon > 0$.

3.1. Validity of the block-sparse signal model

We assume that the clean signal $\mathbf{x}$ (or features extracted from it) can be expressed in terms of a dictionary of signals $\mathbf{D}_s$ with coefficients $\mathbf{c}_s$, i.e. $\mathbf{x} = \mathbf{D}_s \mathbf{c}_s$. We also assume that $\mathbf{D}_s$ can be decomposed into r blocks, one per class, and that its columns are unit norm. Letting $\mathbf{D}_s[i] \in \mathbb{R}^{n \times m_i}$ denote the dictionary for the ith class and $\mathbf{c}_s[i] \in \mathbb{R}^{m_i}$ the corresponding set of coefficients, we can write the clean signal as

$$\mathbf{x} = \sum_{i=1}^{r} \mathbf{D}_s[i] \mathbf{c}_s[i]. \tag{5}$$

A priori this might seem like a strong assumption, which is violated by many datasets. However, we argue that the validity of this model depends on the choice of the dictionary (fixed or learned), the choice of additional structure on the coefficients (e.g, sparse, block-sparse), and the choice of a data embedding (e.g., fixed features such as SIFT, or unsupervised learned deep features).

For example, as is common in image denoising, the dictionary $\mathbf{D}_s$ could be chosen as a Fourier or wavelet basis and the coefficients sparse with respect to such basis. Alternatively, as is common in face classification (Belhumeur & Kriegman, 1998; Basri & Jacobs, 2003; Ho et al., 2003) where each class can be described by a low-dimensional subspace, $\mathbf{D}_s$ could be chosen as the training set and different blocks of the dictionary could correspond to different classes (subspaces). These results will motivate us to report experiments on face classification datasets, such as YaleB (Lee et al., 2005), for which our modeling assumptions are satisfied.

Even when the data from one class cannot be well approximated by a linear subspace, we note that the model $\mathbf{x} = \mathbf{D}_s \mathbf{c}_s$ is actually nonlinear with (structured) sparsity constraints on $\mathbf{c}_s$. Indeed, in manifold learning, data is often approximated locally by a subspace of nearest neighbors (Roweis & Saul, 2000; Elhamifar & Vidal, 2011). When $\mathbf{c}_s$ is block sparse, the model $\mathbf{D}_s \mathbf{c}_s$ thus generates data on a manifold by stitching locally linear approximations. This will motivate our experiments on the MNIST dataset, where the set of all images of one digit is not a linear subspace, but our model still performs well.

3.2. Structured block-sparse attack model

We assume that the attack $\boldsymbol{\delta} \in \Delta_p$ can be expressed in terms of an attack dictionary $\mathbf{D}_a$ with coefficients $\mathbf{c}_a$, i.e., $\boldsymbol{\delta} = \mathbf{D}_a \mathbf{c}_a$. We also assume that the columns of $\mathbf{D}_a$ are unit norm and that the dictionary can be decomposed into a blocks, one per attack type. Moreover, we assume that the columns of $\mathbf{D}_a$ are chosen as ℓ_p attacks evaluated at points in the training set. Therefore, each block of $\mathbf{D}_a$ can be further subdivided into r subblocks, one per class. As

a consequence, the dictionary $\mathbf{D}_a$ is composed of $r \times a$ blocks, $\mathbf{D}_a[i][j] \in \mathbb{R}^{n \times k_{ij}}$, each one corresponding to data points from the ith class and jth attack type. Decomposing $\mathbf{c}_a$ according to the block structure of $\mathbf{D}_a$ so that $\mathbf{c}_a[i][j] \in \mathbb{R}^{k_{ij}}$ is the vector of coefficients corresponding to block $\mathbf{D}_a[i][j]$, we obtain the following threat model:

$$\boldsymbol{\delta} = \sum_{i=1}^{r} \sum_{j=1}^{a} \mathbf{D}_a[i][j]\mathbf{c}_a[i][j]. \tag{6}$$

Observe that when $\boldsymbol{\delta}$ is an ℓ_p attack evaluated at one of the points in the training set, the vector of coefficients $\mathbf{c}_a$ is 1-sparse. In general, however, $\boldsymbol{\delta}$ will be evaluated at a test data point. In the next section, we will show in this case, we still expect $\boldsymbol{\delta}$ to be 1-block sparse for attacks on neural networks with ReLU activations. That is, we expect an attack of a certain type evaluated at a test point from one of the classes to be well approximated as a sparse linear combination of attacks of the same type but evaluated at other training data points from the same class.

3.3. Validity of the attack model for ReLU networks

Consider a ReLU network $f_\theta \colon \mathcal{X} \to \mathbb{R}^r$, mapping the input to a point in $\mathbb{R}^r$, where r is the number of classes. The network is composed of k layers, each consisting of an affine transformation followed by a ReLU non-linearity, i.e.

$$f_\theta(\mathbf{x}) = \mathbf{W}_k(\dots(\mathbf{W}_2(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)_+ + \mathbf{b}_2)_+ \dots)_+ + \mathbf{b}_k, \tag{7}$$

where $\theta = (\mathbf{W}_k, \dots, \mathbf{W}_2, \mathbf{W}_1)$ denotes the parameters and $(\cdot)_+$ is the pointwise ReLU operation. The classification decision is then an argmax operation given by $\arg\max_{i=1\dots r} |z_i^{\mathbf{x},k}|$, where $\mathbf{z}_\mathbf{x}^k = f_\theta(\mathbf{x})$ is the network output.

ReLU networks partition the input space into several polyhedral regions, inside each of which the network behaves like an affine map (Balestriero & Baraniuk, 2020). Specifically, the affine region around $\mathbf{x}$ is given by the set of all points $\mathbf{x}'$ that produce the same *sign pattern* as $\mathbf{x}$ after the ReLU activations at all the intermediate layers. More formally, defining $\boldsymbol{\zeta}^l(\mathbf{x}) = \operatorname{sgn}(\mathbf{W}_l(\dots(\mathbf{W}_2(\mathbf{W}_1\mathbf{x}+\mathbf{b}_1)_+ + \mathbf{b}_2)_+ \dots) + b_l)$ to be the sign pattern at layer l for the input $\mathbf{x}$, the neural network f_θ behaves as an affine function $f_\theta(\mathbf{x}) = \mathbf{P}_S^\top \mathbf{x} + \mathbf{q}$ in the region $S = \{\mathbf{x}' \colon (\boldsymbol{\zeta}^1(\mathbf{x}'), \boldsymbol{\zeta}^2(\mathbf{x}'), \dots, \boldsymbol{\zeta}^k(\mathbf{x}')) = (\boldsymbol{\zeta}^1(\mathbf{x}), \boldsymbol{\zeta}^2(\mathbf{x}), \dots, \boldsymbol{\zeta}^k(\mathbf{x}))\}$. Therefore, the gradient of a loss $L(f_\theta(\mathbf{x}), y)$ in for all $\mathbf{x}' \in S$ is equal to

$$\begin{aligned} \nabla_\mathbf{x} L(f_\theta(\mathbf{x}'), y) &= \frac{\partial f_\theta(\mathbf{x}')}{\partial \mathbf{x}'} \nabla_{\mathbf{z}_\mathbf{x}^k} L(\mathbf{z}_\mathbf{x}^k, y) \\ &= \mathbf{P}_S \nabla_{\mathbf{z}_x^k} L(\mathbf{z}_\mathbf{x}^k, y) \end{aligned} \tag{8}$$

where the last part of (8) comes by the affine approximation of the output of the ReLU network i.e., $f_\theta(\mathbf{x}) = \mathbf{P}_S^\top \mathbf{x} + \mathbf{q}$

Thus, the gradient of the loss function of a ReLU network at a test point in S lives in the columnspace of a matrix $\mathbf{P}_S$. That being said, the gradient of the loss at a test point can be expressed as a linear combination of the gradients at training samples in the same region S. Note that this property holds true for popular loss functions e.g. cross-entropy loss, mean squared loss, etc. Moreover, we further assume that training samples in S belong to the same class, which is a reasonable assumption to make for ReLU networks (Sattelberg et al., 2020).

Hence, for any test point lying in region S, if there exists a training datapoint of the same class and also in S, then there will exist a vector $\mathbf{c}_a$ that is a feasible solution of (2) and block-sparse in the columnspace of $\mathbf{D}_a$ with only one non-zero block (assuming that the $\boldsymbol{\delta}$ comes from a single attack from the family). Recent works (Lee et al., 2019) show that ReLU networks can be trained to have large linear regions, hence it is reasonable to expect that S contains a training point.

4. Reverse engineering of ℓ_p-bounded attacks

In Section 3 we introduced a block-sparse model of attacked signals, $\mathbf{x}' = \mathbf{D}_s\mathbf{c}_s + \mathbf{D}_a\mathbf{c}_a$, where $\mathbf{D}_s$ is a dictionary of clean signals (typically the training set), $\mathbf{D}_a$ is a dictionary of attacks (typically ℓ_p attacks on the training set), and $\mathbf{c}_s$ and $\mathbf{c}_a$ are block-sparse vectors whose nonzero coefficients indicate the class and the attack type. In this section, we show how to reverse engineer the attack and clean signal.

4.1. Block sparse optimization approach

Assume that test sample $\mathbf{x}'$ has been corrupted by an attack of a single type. Since $\mathbf{x}'$ belongs to only one of the classes, we expect vectors $\mathbf{c}_s$ and $\mathbf{c}_a$ to be 1-block-sparse. Therefore, the problem of reverse engineering ℓ_p attacks can be cast as a standard block-sparse optimization problem, where we minimize the total number of nonzero blocks in $\mathbf{c}_s$ and $\mathbf{c}_a$ needed to generate $\mathbf{x}'$, i.e.

$$\begin{aligned} \min_{\mathbf{c}_s, \mathbf{c}_a} \quad & \sum_{i=1}^{r} \left(I(\|\mathbf{c}_s[i]\|_2) + \sum_{j=1}^{a} I(\|\mathbf{c}_a[i][j]\|_2) \right) \\ \text{s.t.} \quad & \mathbf{x}' = \mathbf{D}_s\mathbf{c}_s + \mathbf{D}_a\mathbf{c}_a, \end{aligned} \tag{9}$$

where $I(\cdot)$ is the indicator function, i.e., $I(x) = 1$ if $x \neq 0$ and $I(x) = 0$ if $x = 0$. As is common in block-sparse recovery (Elhamifar & Vidal, 2012), a convex relaxation of the problem of minimizing the number of nonzero blocks is given by:

$$\begin{aligned} \min_{\mathbf{c}_s, \mathbf{c}_a} \quad & \sum_{i=1}^{r} \|\mathbf{c}_s[i]\|_2 + \sum_{j=1}^{a} \|\mathbf{c}_a[i][j]\|_2 \\ \text{s.t.} \quad & \mathbf{x}' = \mathbf{D}_s\mathbf{c}_s + \mathbf{D}_a\mathbf{c}_a, \end{aligned} \tag{10}$$

where the sum of the ℓ_2 norms of the blocks, also known as the ℓ_1/ℓ_2 norm, is a convex surrogate for the number of nonzero blocks.

4.2. Active Set Homotopy Algorithm

In practice, we further relax the problem and solve the regularized noisy version of problem (10), which can be written in the form,

$$\min_{\mathbf{c}_s,\mathbf{c}_a} \frac{1}{2}\|\mathbf{x}' - \mathbf{D}_s\mathbf{c}_s - \mathbf{D}_a\mathbf{c}_a\|_2^2 + \lambda_s \sum_{i=1}^{r} \|\mathbf{c}_s[i]\|_2 + \lambda_a \sum_{j=1}^{a} \|\mathbf{c}_a[i][j]\|_2. \tag{11}$$

Because the size of the dictionaries $\mathbf{D}_s$ and $\mathbf{D}_a$ can be large, it is crucial to develop scalable algorithms to solve the above optimization problem. Furthermore, the choice of λ_s and λ_a play a crucial role in enforcing the correct level of block-sparsity. High values of these parameters will drive the solutions $\mathbf{c}_s, \mathbf{c}_a$ to the zero vector, and too low values will result in a solution that is not block-sparse as desired. To address both issues, we develop an active-set based homotopy algorithm. The main insight is that instead of solving an optimization problem using the full data matrix, we can restrict $\mathbf{D}_s$ and $\mathbf{D}_a$ to the blocks that correspond to non-zero blocks of the optimal $\mathbf{c}_s$ and $\mathbf{c}_a$. Using the optimality conditions of problem (11), we can derive an algorithm that maintains a list of block indices for both $\mathbf{c}_s$ and $\mathbf{c}_a$, denoted as the *active sets* T_s and T_a, and solve reduced subproblems based on these indices, significantly reducing runtime. The details of the derivation can be found in the Appendix.

Additionally, to pick proper values of λ_s and λ_a, we employ techniques from homotopy methods in sparse optimization to construct a sequence of decreasing values for λ_s and λ_a (Malioutov et al., 2005). Traditionally, homotopy methods for ℓ_1 minimization use the fact that the solution path as a function of regularization strength is piecewise linear, with breakpoints when the support of the solution changes. However, with the block-sparsity constraint, the path is nonlinear (Yau & Hui, 2017), and thus we approximate this path using a sequence of λ values. The initial value of λ_s and λ_a is chosen to be a hyperparameter $\gamma \in (0, 1)$ times the value that produces the all-zeros vector based on the optimality conditions. In Algorithm 1 in the Appendix, we provide the details of the active set homotopy algorithm.

5. Theoretical analysis of the block-sparse minimization problem

In this section, we provide geometrically interpretable conditions under which the true signal class and attack type, which is generated by a single ℓ_p perturbation type, can be recovered from the nonzero blocks of $\mathbf{c}_s, \mathbf{c}_a$ using the proposed block-sparse minimization approach given in (10).

At first sight, one may think that existing conditions for block-sparse recovery in a union of low-dimensional subspaces (Elhamifar & Vidal, 2012), which require the subspaces to be disjoint and sufficiently separated, might be directly applicable to our problem. However, the adversarial setting presents several additional challenges. First, we do not need conditions for all block pairs, but only for the block pair formed by one signal subspace and one signal-attack subspace. Second, the two non-zero blocks are not independent from each other, because if we determine the signal-attack block (i^*, j^*), then we also determine the signal block i^*. Third, we need to disentangle not only one signal class from another, but also one attack type from another, and signals from attacks.

In the following, we address these three challenges. Specifically, we significantly improve on the block-sparse recovery results of (Elhamifar & Vidal, 2012) by getting rid of the strong assumption of disjointness among *all* pairs of subspaces spanned by the blocks of the dictionaries. Moreover, our analysis goes one step beyond previous efforts (e.g. (Wang et al., 2017)) to generalize the subspace-sparse recovery results (You & Vidal, 2015a;b) in the following ways. First, we focus on block-sparse recovery in a union of dictionaries as opposed to (Wang et al., 2017), which focuses on a single dictionary. Second, our problem has more specific structure i.e., the dependency of non-zero blocks of the signal (see Remark 5.2). Third, our conditions are based on different newly introduced geometric measures i.e. covering radius and angular distances induced by the ℓ_1/ℓ_2 norm. This leads to our first main result (Theorem 5.5). Finally, we provide an additional result (see Theorem 5.8), which relaxes Theorem 5.5 by involving the angular distances between points in a set of Lebesgue measure zero (instead of all points in the direct sum of the signal and attack subspaces) and a finite set.

Proposition 5.1 gives a necessary and sufficient condition for recovering the correct signal and attack by solving problem (10). Let $\mathcal{I} = \{1, 2, \ldots, r\}$ and $\mathcal{J} = \{1, 2, \ldots, a\}$ denote the indices for the blocks of $\mathbf{D}_s$ and $\mathbf{D}_a$ and vectors $\mathbf{c}_s$ and $\mathbf{c}_a$ respectively. We define the *correct-class minimum* ℓ_1/ℓ_2 *vectors* $\hat{\mathbf{c}}_s^*, \hat{\mathbf{c}}_a^*$ with non-zero blocks $\hat{\mathbf{c}}_s^*[i^*]$ and $\hat{\mathbf{c}}_a^*[i^*][j^*]$ as

$$\{\hat{\mathbf{c}}_s^*, \hat{\mathbf{c}}_a^*\} \equiv \arg\min_{\mathbf{c}_s,\mathbf{c}_a} \|\mathbf{c}_s[i^*]\|_2 + \|\mathbf{c}_a[i^*][j^*]\|_2 \quad \text{s.t. } \mathbf{x}' = \mathbf{D}_s[i^*]\mathbf{c}_s[i^*] + \mathbf{D}_a[i^*][j^*]\mathbf{c}_a[i^*][j^*], \tag{12}$$

and the *wrong-class minimum* ℓ_1/ℓ_2 *norm vectors* $\tilde{\mathbf{c}}_s^*, \tilde{\mathbf{c}}_a^*$ as,

$$\{\tilde{\mathbf{c}}_s^*, \tilde{\mathbf{c}}_a^*\} \equiv$$
$$\arg\min_{\mathbf{c}_s, \mathbf{c}_a} \sum_{i \in \mathcal{I} \setminus \{i^*\}} \|\mathbf{c}_s[i]\|_2 + \sum_{i \in \mathcal{I}, j \in \mathcal{J} \setminus \{j^*\}} \|\mathbf{c}_a[i][j]\|_2$$
$$\text{s.t.}$$
$$\mathbf{x}' = \sum_{i \in \mathcal{I} \setminus \{i^*\}} \mathbf{D}_s[i]\mathbf{c}_s[i] + \sum_{i \in \mathcal{I}, j \in \mathcal{J} \setminus \{j^*\}} \mathbf{D}_a[i][j]\mathbf{c}_a[i][j] \tag{13}$$

Note that the non-zero blocks of $\tilde{\mathbf{c}}_s^*$ and $\tilde{\mathbf{c}}_a^*$ do not correspond to the correct signal and attack.

Proposition 5.1. *The correct classes of the signal $\mathbf{x} \in \mathcal{S}_{i^*}^{\mathbf{x}}$ and the attack $\boldsymbol{\delta} \in \mathcal{S}_{i^*,j^*}^{\boldsymbol{\delta}}$, with $\mathcal{S}_{i^*}^{\mathbf{x}} \cap \mathcal{S}_{i^*j^*}^{\boldsymbol{\delta}} = \emptyset$, can be recovered by solving (10) if and only if, $\forall i^*, j^*, \forall \mathbf{x}' \in (\mathcal{S}_{i^*}^{\mathbf{x}} \oplus \mathcal{S}_{i^*,j^*}^{\boldsymbol{\delta}}), \mathbf{x} \neq \mathbf{0}$, the ℓ_1/ℓ_2 norm of the correct-class minimum ℓ_1/ℓ_2 vectors $\hat{\mathbf{c}}_s^*, \hat{\mathbf{c}}_a^*$ is strictly less that of the wrong-class minimum ℓ_1/ℓ_2 norm vectors $\tilde{\mathbf{c}}_s^*, \tilde{\mathbf{c}}_a^*$, i.e.,*

$$\|\hat{\mathbf{c}}_s^*\|_{1,2} + \|\hat{\mathbf{c}}_a^*\|_{1,2} < \|\tilde{\mathbf{c}}_s^*\|_{1,2} + \|\tilde{\mathbf{c}}_a^*\|_{1,2}. \tag{14}$$

Remark 5.2. Note that disjointness between $\mathcal{S}_{i^*}^{\mathbf{x}}$ and $\mathcal{S}_{i^*j^*}^{\boldsymbol{\delta}}$ is necessary if we want to recover the class of the signal *and* the attack. However, in case that disjointness is violated, we can still guarantee the recovery of the correct class of the signal since we know that attacks depend on the signal class.

Next, we aim to provide more geometrically interpretable conditions for the recovery of the signal and attack classes. First, we generalize the standard angular distance originally used for ℓ_1 norm minimization problems, (You & Vidal, 2015a), to the particular case of ℓ_1/ℓ_2 norm minimization.

Definition 5.3. Let $\mathcal{D}$ be a set of unit ℓ_2 norm columns of the dictionary $\mathbf{D} = [\mathbf{D}[1], \mathbf{D}[2], \ldots, \mathbf{D}[c]]$ with $\mathbf{D}[i] \in \mathbb{R}^{n \times m}$. The angular distance between the atoms in $\mathcal{D}$ and a vector $\mathbf{v} \in \mathbb{R}^n$ is defined as,

$$\theta_{1,2}(\mathbf{v}, \pm\mathcal{D}) = \cos^{-1}\left(\frac{1}{\sqrt{m}}\|\mathbf{D}^\top \frac{\mathbf{v}}{\|\mathbf{v}\|_2}\|_{\infty,2}\right). \tag{15}$$

This definition can also be extended to a set of vectors $\mathcal{V}$ as

$$\theta_{1,2}(\mathcal{V}, \pm\mathcal{D}) = \inf_{\mathbf{v} \in \mathcal{V}} \cos^{-1}\left(\frac{1}{\sqrt{m}}\|\mathbf{D}^\top \frac{\mathbf{v}}{\|\mathbf{v}\|_2}\|_{\infty,2}\right). \tag{16}$$

Similarly, we define a generalized version of the covering radius of a set induced by the ℓ_1/ℓ_2 norm.

Definition 5.4. The covering radius $\gamma_{1,2}(\mathcal{D})$ of a set $\mathcal{D}$ consisting of the columns of matrix $\mathbf{D}$ is defined as,

$$\gamma_{1,2}(\pm\mathcal{D}) = \sup\{\theta_{1,2}(\mathbf{v}, \pm\mathcal{D}), \mathbf{v} \in \mathbb{S}^{n-1} \cap \text{span}(\mathcal{D})\}. \tag{17}$$

Note that the covering radius captures how well-separated the atoms of *the blocks* of $\mathbf{D}[i]$ are, and is a decreasing function of the distance between atoms.

Let $\mathcal{D}_{i^*j^*}$ be the set that contains the columns of $\mathbf{D}_s[i^*], \mathbf{D}_a[i^*][j^*]$, and $\mathcal{D}_{i^*j^*}^-$ the set with all remaining columns of the blocks $\mathbf{D}_s[i], \forall i \in \mathcal{I} \setminus i^*$ and $\mathbf{D}_a[i][j]$ for $i \in \mathcal{I}$ and $j \in \mathcal{J} \setminus j^*$. We now provide a sufficient condition, which ensures the recovery of the correct classes of the signal and the attack.

Theorem 5.5. *The correct classes of the signal and the attack of an adversarially perturbed signal in $\mathcal{S}_{i^*} \oplus \mathcal{S}_{i^*j^*}$ can be recovered by solving problem (10), if the following primary recovery condition (PRC)*

$$\gamma_{1,2}(\pm\mathcal{D}_{i^*j^*}) < \theta_{1,2}(\mathcal{S}_{i^*}^{\mathbf{x}} \oplus \mathcal{S}_{i^*j^*}^{\boldsymbol{\delta}}, \pm\mathcal{D}_{i^*j^*}^-) \tag{18}$$

holds for the dictionaries $\mathbf{D}_s$ and $\mathbf{D}_a$.

Theorem 5.5 offers a geometric intuition for recovery guarantees. Note that (18) depends only on the properties of the dictionaries $\mathbf{D}_s$ and $\mathbf{D}_a$. Specifically, (18) is easier satisfied when a) the covering radius of $\mathcal{D}_{i^*j^*}$ is small, meaning that columns of *both* $\mathbf{D}_s[i^*]$ and $\mathbf{D}_a[i^*][j^*]$ are well distributed in $\mathcal{S}_{i^*}^{\mathbf{x}}$ and $\mathcal{S}_{i^*j^*}^{\boldsymbol{\delta}}$ respectively or b) the atoms of the remaining blocks of $\mathbf{D}_s, \mathbf{D}_a$ are sufficiently away from $\mathcal{S}_{i^*} \oplus \mathcal{S}_{i^*j^*}$.

Next, we derive the dual recovery condition (DRC), which only needs to hold a subset of points in $\mathcal{S}_{i^*} \oplus \mathcal{S}_{i^*j^*}$ called as *dual points*. Before illustrating the DRC, we first define the polar set induced by the $\ell_{1,2}$ norm and the dual points.

Definition 5.6. The polar of the set $\mathcal{D}$ containing the columns of matrix $\mathbf{D}$ induced by the $\ell_{1,2}$ norm is defined as

$$\mathcal{K}_{\ell_{1,2}}^o(\mathcal{D}) = \{\mathbf{v} \in \mathcal{R}(\mathbf{D}) : \frac{1}{\sqrt{m}}\|\mathbf{D}^\top \mathbf{v}\|_{\infty,2} \leq 1\}. \tag{19}$$

where $\mathcal{R}(\mathbf{D})$ is the range of $\mathbf{D}$.

Definition 5.7. The set of dual points of matrix $\mathbf{D}$, denoted as $\mathcal{A}(\mathcal{D})$, is the set of extreme points of $\mathcal{K}_{\ell_{1,2}}^o(\mathcal{D})$, which is the polar set of $\mathcal{D}$.

Theorem 5.8. *The correct classes of the signal and the attack can be recovered by solving problem (10), if the following dual recovery condition (DRC) is satisfied*

$$\gamma_{1,2}(\mathcal{D}_{i^*j^*}) < \theta_{1,2}\left(\mathcal{A}\left(\mathcal{D}_{i^*j^*}\right), \pm\mathcal{D}_{i^*j^*}^-\right), \tag{20}$$

Theorem 5.8 requires the covering radius of $\mathcal{D}_{i^*j^*}$ to be smaller than the minimum angular distance between the dual points of $\mathcal{D}_{i^*j^*}$, which form a set of Lebesgue measure zero, and elements of the set $\mathcal{D}_{i^*j^*}^-$.

6. Experiments

In this section, we present experiments on the Extended YaleB Face dataset and the MNIST dataset.

6.1. Experimental Setup

Network architectures and attack evaluation. For the YaleB Face dataset, we train a simple 3-layer fully-connected ReLU neural network with 256 hidden units per layer, which already serves as a strong baseline obtaining 96.3% accuracy. For the MNIST dataset, we train a 4-layer convolutional network, identical to the architecture from (Carlini & Wagner, 2017c). All networks are trained with a cross-entropy loss.

We consider the family of $\{\ell_1, \ell_2, \ell_\infty\}$ PGD attacks. ℓ_1 PGD refers to the Sparse ℓ_1 PGD attack from (Tramer & Boneh, 2019). For optimization, we use the active set homotopy algorithm developed in Section 4.2. The Appendix contains full experimental details.

Metrics. We choose $\mathbf{D}_s$ as a dictionary whose columns are the flattened training images, and $\mathbf{D}_a$ is a dictionary whose columns are the ℓ_p perturbations for each training image. For each block in $\mathbf{D}_s$ and $\mathbf{D}_a$, we subsample 200 training datapoints to limit the dictionary size, and we normalize the columns of the dictionary to unit ℓ_2 norm to keep the same scaling for all blocks. For a given perturbed image $\mathbf{x}'$, we run Algorithm 1 to obtain the output coefficients $\hat{\mathbf{c}}_s$ and $\hat{\mathbf{c}}_a$. We define the predicted block indices for the signal and attack dictionaries to be:

$$\hat{i} = \arg\min_i \|\mathbf{x}' - \mathbf{D}_s[i]\hat{\mathbf{c}}_s[i] - \mathbf{D}_a\hat{\mathbf{c}}_a\|_2 \tag{21}$$

$$\hat{j} = \arg\min_j \|\mathbf{x}' - \mathbf{D}_s\hat{\mathbf{c}}_s - \mathbf{D}_a[\hat{i}][j]\hat{\mathbf{c}}_a[\hat{i}][j]\|_2 \tag{22}$$

Using these indices, we define two classification methods and one attack detection method:

1. *SBSC (Structured Block-Sparse Classifier)*: This method predicts the class of the test image as $\hat{i}$.
2. *SBSC+CNN (Denoiser)*: From $\hat{\mathbf{c}}_s$, this method computes a denoised image as $\hat{\mathbf{x}} = \mathbf{D}_s[\hat{i}]\hat{\mathbf{c}}_s[\hat{i}]$ and then predicts the class of the test image from the output of the original network at the denoised datapoint, i.e. $f_\theta(\hat{\mathbf{x}})$
3. *SBSAD (Structured Block-Sparse Attack Detector)*: This method returns $\hat{j}$, which represents the predicted attack type of the test image.

For each method, we report the accuracy of prediction with respect to the ground truth label. For the SBSC and SBSC+CNN methods, the label is the correct label of the test image, while for SBSAD, the label is the true ℓ_p perturbation type that was applied to the test image. As a naive block-sparse classifier baseline, we denote BSC as a block-sparse classifier which does not model the structure of the attack perturbation, but simply models $\mathbf{x}' = \mathbf{D}_s\mathbf{c}_s$ (Elhamifar & Vidal, 2012). We also consider a BSC+CNN baseline, which predicts the class from $f_\theta(\mathbf{D}_s[\hat{i}]\hat{\mathbf{c}}_s[\hat{i}])$ as above, except $\hat{\mathbf{c}}_s$ is obtained from the BSC problem.

6.2. YaleB Face Dataset

We first evaluate our method on images from the Extended YaleB Face Dataset (Lee et al., 2005), a 38-way classification task. While the adversarial learning literature does not usually evaluate attacks on this dataset, we choose it because it exhibits the self-expressiveness property. Indeed, face images of an individual under varying lighting conditions have been shown to lie in a low-dimensional subspace (Belhumeur & Kriegman, 1998; Basri & Jacobs, 2003; Ho et al., 2003). Our goal is to complement our theoretical recovery guarantees by demonstrating the effectiveness of our approach in determining the correct signal and attack type, which is illustrated in Table 1. For all perturbation types, we observe the SBSC approach significantly improves upon the accuracy of the undefended model, indicating the successful decoupling of the signal and attack. One phenomenon we see is the remarkable robustness of block-sparse classifiers, even without attack modelling. The BSC baseline consistently improves the adversarial accuracy of the undefended model; however, the low BSC+CNN accuracy indicates that there is still significant noise in the data modelling. On the other hand, the SBSC and SBSC+CNN are able to improve over the BSC baseline by around 20%, indicating that explicitly modelling the attack helps signal classification for both the block-sparse classifier as well as the original classification network.

6.3. MNIST

Despite the simplicity of the MNIST dataset, networks trained on MNIST are still brittle to attacks that arise from a union of perturbations. Specifically, in (Maini et al., 2020b), the authors observe that most state-of-the-art adversarial training defenses for MNIST are only robust to one type of ℓ_p attack.

Baselines. While we emphasize that our approach is not primarily a defense, but rather a principled attack classification and signal decoupling algorithm, we can still compare our signal classification accuracy to a variety of state of the art methods for defending against a union of attacks. First, we consider classifiers M_1, M_2, M_∞ trained with adversarial training (Madry et al., 2017) against ℓ_1, ℓ_2, or ℓ_∞ perturbations, respectively. Next, we compare against variants of adversarial training: the MAX, AVG and MSD approaches

Table 1. Adversarial image and attack classification accuracy on YaleB dataset. BSC denotes the block-sparse classifier baseline, SBSC denotes the structured block-sparse signal classifier, SBSC+CNN denotes the denoised model, and SBSAD denotes the structured block-sparse attack detector.

Yale-B	CNN	BSC	BSC+CNN	SBSC	SBSC+CNN	SBSAD
ℓ_∞ PGD ($\epsilon = 0.02$)	15.1%	79%	2%	**97%**	93%	52%
ℓ_2 PGD ($\epsilon = 0.75$)	4.2%	51%	2%	**96%**	87%	76%
ℓ_1 PGD ($\epsilon = 15$)	53.7%	81%	3%	**96%**	93%	39%
Average	24.3%	70.3%	2.3%	**96.3%**	91%	55.7%

Table 2. Adversarial image and attack classification accuracy on digit classification of MNIST dataset. See above table for column descriptions. The clean accuracy represents the accuracy of the method with unperturbed test inputs.

MNIST	CNN	M_∞	M_2	M_1	MAX	AVG	MSD	BSC	SBSC	SBSC+CNN	SBSAD
Clean accuracy	98.99%	99.1%	99.2%	99.0%	98.6%	98.1%	98.3%	92%	94%	99%	-
ℓ_∞ PGD ($\epsilon = 0.3$)	0.03%	**90.3%**	0.4%	0.0%	51.0%	65.2%	62.7%	54%	77.27%	76.83%	73.2%
ℓ_2 PGD ($\epsilon = 2.0$)	44.13%	68.8%	69.2%	38.7%	64.1%	67.9%	70.2%	76%	**85.34%**	85.17%	46%
ℓ_1 PGD ($\epsilon = 10.0$)	41.98%	61.8%	51.1%	74.6%	61.2%	66.5%	70.4%	75%	**85.97%**	85.85%	36.6%
Average	28.71%	73.63%	40.23%	37.77%	58.66%	66.53%	67.76%	68.33%	**82.82%**	82.61%	51.93%
Unseen Attacks											
ℓ_∞ MIM ($\epsilon = 0.3$)	0.02%	**92.3%**	11.2%	0.1%	70.7%	76.7%	71.0%	59.5%	74.3%	74.2%	79.0%
ℓ_2 C-W ($\epsilon = 2.0$)	0%	79.6%	74.5%	44.8%	72.1%	72.4%	74.5%	**89.1%**	87.1%	87.1%	60.4%
ℓ_2 DDN ($\epsilon = 2.0$)	0%	63.9%	70.5%	40.0%	62.5%	64.6%	69.5%	**88.8%**	87.2%	87.1%	57.8%
Average	0%	78.6%	52.06%	28.3%	68.43%	71.23%	71.66%	79.13%	**82.86%**	82.8%	65.73%

(Maini et al., 2020b; Tramer & Boneh, 2019). Finally, we compare against the BSC baseline.

Quality of Defense. Table 2 summarizes our results on the MNIST dataset. The top half demonstrates that our proposed block-sparse approach improves upon state of the art adversarial training defenses against a union of attacks from ℓ_2, ℓ_1, and ℓ_∞ PGD attacks by about 15% on average. The high accuracy of the Denoiser+CNN model also shows that $\mathbf{D}_s[\hat{i}]\hat{\mathbf{c}}_s[\hat{i}]$ is a good model of the denoised data. Surprisingly, even though the block-sparse classifier is not the strongest baseline for MNIST, as indicated by the relatively low clean accuracy of 94%, we observe that it is much more robust to ℓ_p perturbations than the neural network models as the strength of the attack increases.

Performance on unseen test-time attacks. Our dictionary $\mathbf{D}_a$ consists of $\{\ell_1, \ell_2, \ell_\infty\}$ PGD attacks, so the SBSAD must predict one of these three classes. However, we can evaluate our method on test-time attacks that are non-PGD ℓ_p attacks for $p \in \{1, 2, \infty\}$. The second half of Table 2 demonstrates the accuracy of our method on the ℓ_∞ Momentum Iterative Method (MIM) (Dong et al., 2018), the ℓ_2 Carlini-Wagner (C-W) attack (Carlini & Wagner, 2017b), and the ℓ_2 Decoupled Direction and Norm (DDN) attack (Rony et al., 2019). The block-sparse baseline performs remarkably well at denoising even though it does not model the attack structure. In principle, it does not make sense for our method to capture these attacks through $\mathbf{D}_a$; however on average, we still observe a slight increase in accuracy by modelling some portion of the perturbation through the SBSC method. Perhaps more surprisingly, our method still has high attack classification accuracy, indicating that for the purposes of determining the attack family, the attacks can be well-approximated by a linear combination of PGD attacks.

7. Conclusion

In this paper, we studied the conditions under which we can reverse engineer adversarial attacks by determining the type of attack from a corrupted signal. We provided a structured block-sparse optimization approach to model not only the signal as a block-sparse combination of datapoints, but also the attack perturbation as a block-sparse combination of attacks. Under this optimization approach, we derived theoretical conditions under which recovery of the correct signal and attack type is feasible. Finally, we experimentally verified the validity of the structured block-sparse optimization approach on the YaleB and MNIST datasets. We believe there are many directions to further study the properties of block-sparse classifiers, such as introducing non-linear embedding dictionaries.

Acknowledgements

This work is partially supported by by the European Union under the Horizon 2020 Marie-Skłodowska- Curie Global Fellowship program: HyPPOCRATES— H2020- MSCA-IF-2018, Grant Agreement Number: 844290, and DARPA RED Contract DARPA HR00112090132. DT also thanks Naren Manoj for helpful discussions.

References

Agrawal, A., Verschueren, R., Diamond, S., and Boyd, S. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

Athalye, A. and Carlini, N. On the robustness of the cvpr 2018 white-box adversarial example defenses. *ArXiv*, abs/1804.03286, 2018.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *ICML*, 2018a.

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018b.

Balestriero, R. and Baraniuk, R. G. Mad max: Affine spline insights into deep learning. *Proceedings of the IEEE*, 2020.

Basri, R. and Jacobs, D. Lambertian reflection and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, 2003.

Belhumeur, P. and Kriegman, D. What is the set of images of an object under all possible illumination conditions? *International Journal of Computer Vision*, 28(3):1–16, 1998.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., vSrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.

Bulusu, S., Kailkhura, B., Li, B., Varshney, P. K., and Song, D. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347, 2020.

Candès, E., Romberg, J., and Tao, T. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8): 1207–1223, 2006.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, pp. 3–14, New York, NY, USA, 2017a. ACM. ISBN 978-1-4503-5202-4. doi: 10.1145/3128572.3140444. URL http://doi.acm.org/10.1145/3128572.3140444.

Carlini, N. and Wagner, D. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. IEEE, 2017b.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. IEEE, 2017c.

Castrodad, A. and Sapiro, G. Sparse modeling of human actions from motion imagery. *International Journal of Computer Vision*, 100(1):1–15, 2012.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. *International Conference on Machine Learning*, 2019.

Croce, F. and Hein, M. Provable robustness against all adversarial l_p-perturbations for $p \geq 1$. *arXiv preprint arXiv:1905.11213*, 2019.

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

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., and Li, J. Boosting adversarial attacks with momentum. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9185–9193. IEEE, 2018.

Elhamifar, E. and Vidal, R. Sparse manifold clustering and embedding. In *Neural Information Processing and Systems*, 2011.

Elhamifar, E. and Vidal, R. Block-sparse recovery via convex optimization. *IEEE Transactions on Signal Processing*, 60(8):4094–4107, 2012.

Elhamifar, E. and Vidal, R. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11): 2765–2781, 2013.

Gemmeke, J. F., Virtanen, T., and Hurmalainen, A. Exemplar-based sparse representations for noise robust automatic speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2067–2080, 2011.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *ICLR*, 2015.

Grosse, K., Manoharan, P., Papernot, N., Backes, M., and McDaniel, P. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

Ho, J., Yang, M. H., Lim, J., Lee, K., and Kriegman, D. Clustering appearances of objects under varying illumination conditions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11–18, 2003.

Julien Mairal, F. B. and Ponce, J. Sparse modeling for image and vision processing. *Foundations and Trends® in Computer Graphics and Vision*, 8(2-3):85–283, 2012.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial machine learning at scale. *ArXiv*, abs/1611.01236, 2016.

Lee, G.-H., Alvarez-Melis, D., and Jaakkola, T. S. Towards robust, locally linear deep networks. *arXiv preprint arXiv:1907.03207*, 2019.

Lee, K.-C., Ho, J., and Kriegman, D. J. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on pattern analysis and machine intelligence*, 27(5):684–698, 2005.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

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

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018b. URL https://openreview.net/forum?id=rJzIBfZAb.

Maini, P., Chen, X., Li, B., and Song, D. Perturbation type categorization for multiple ell_p bounded adversarial robustness. 2020a.

Maini, P., Wong, E., and Kolter, Z. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, pp. 6640–6650. PMLR, 2020b.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. Discriminative learned dictionaries for local image analysis. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

Malioutov, D. M., Cetin, M., and Willsky, A. S. Homotopy continuation for sparse signal representation. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 5, pp. v–733. IEEE, 2005.

Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

Miyato, T., ichi Maeda, S., Koyama, M., and Ishii, S. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41: 1979–1993, 2017.

Moosavi-Dezfooli, S.-M., Shrivastava, A., and Tuzel, O. Divide, denoise, and defend against adversarial attacks. *arXiv preprint arXiv:1802.06806*, 2018.

Niu, Z., Chen, Z., Li, L., Yang, Y., Li, B., and Yi, J. On the limitations of denoising strategies as adversarial defenses. *arXiv preprint arXiv:2012.09384*, 2020.

Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, May 2016. doi: 10.1109/SP.2016.41.

Robinson, D. P., Vidal, R., and You, C. Basis pursuit and orthogonal matching pursuit for subspace-preserving recovery: Theoretical analysis. *arXiv preprint arXiv:1912.13091*, 2019.

Rony, J., Hafemann, L. G., Oliveira, L. S., Ayed, I. B., Sabourin, R., and Granger, E. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4322–4330, 2019.

Roweis, S. and Saul, L. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Sainath, T. N., Ramabhadran, B., Picheny, M., Nahamoo, D., and Kanevsky, D. Exemplar-based sparse representation features: From timit to lvcsr. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8):2598–2613, 2011.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BkJ3ibb0-.

Sattelberg, B., Cavalieri, R., Kirby, M., Peterson, C., and Beveridge, R. Locally linear attributes of relu neural networks. *arXiv preprint arXiv:2012.01940*, 2020.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing Properties

of Neural Networks. In *International Conference on Learning Representations*, 2014.

Tramer, F. and Boneh, D. Adversarial training and robustness for multiple perturbations. *arXiv preprint arXiv:1904.13000*, 2019.

Uesato, J., O'Donoghue, B., Kohli, P., and van den Oord, A. Adversarial risk and the dangers of evaluating against weak attacks. In *ICML*, 2018.

Wang, Y., Tang, Y. Y., Li, L., Chen, H., and Pan, J. Atomic representation-based classification: theory, algorithm, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):6–19, 2017.

Wright, J. and Ma, Y. Dense error correction via ℓ^1-minimization. *IEEE Transactions on Information Theory*, 56(7):3540–3560, 2010.

Wright, J., Yang, A., Ganesh, A., Sastry, S., and Ma, Y. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, Feb. 2009.

Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T. S., and Yan, S. Sparse representation for computer vision and pattern recognitiong. *Proceedings of the IEEE*, 98(6): 1031–1044, 2010.

Yang, A. Y., Jafari, R., Sastry, S., and Bajcsy, R. Distributed recognition of human actions using wearable motion sensor networks. *JAISE*, 1(2):103–115, 2009a. doi: 10.3233/AIS-2009-0016.

Yang, J., Yu, K., Gong, Y., and Huang, T. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009b.

Yau, C. Y. and Hui, T. S. Lars-type algorithm for group lasso. *Statistics and Computing*, 27(4):1041–1048, 2017.

You, C. and Vidal, R. Geometric conditions for subspace-sparse recovery. In *International Conference on Machine learning*, pp. 1585–1593, 2015a.

You, C. and Vidal, R. Subspace-sparse representation. *Arxiv*, abs/1507.01307, 2015b.

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

Zheng, S., Song, Y., Leung, T., and Goodfellow, I. J. Improving the robustness of deep neural networks via stability training. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4480–4488, 2016.

A. Theoretical Results

We first give the proof of Proposition 5.1, which is based on the proofs of relevant results in subspace-sparse recovery (Theorem 2, (Elhamifar & Vidal, 2013; You & Vidal, 2015b;a)) and atomic representation-based recovery (Lemma 2, (Wang et al., 2017).

A.1. Proof of Proposition 5.1.

Proof. ($\Longrightarrow$)

We first prove that (14) is a sufficient condition for recovering the correct class i^* of the signal $\mathbf{x}$ and (i^*j^*) of the attack $\boldsymbol{\delta}$. Let $\mathbf{c}_s^*, \mathbf{c}_a^*$ be optimal solutions of the problem

$$\begin{aligned} \{\mathbf{c}_s^*, \mathbf{c}_a^*\} \equiv \underset{\mathbf{c}_s, \mathbf{c}_a}{\arg\min} \|\mathbf{c}_s\|_2 + \|\mathbf{c}_a\|_2 \\ \text{s.t. } \mathbf{x}' = \mathbf{D}_s\mathbf{c}_s + \mathbf{D}_a\mathbf{c}_a, \end{aligned} \tag{23}$$

The correct classes of the signal and the attack can be recovered when $\mathbf{c}_s^*[i] = 0$ for $i \neq i^*$ and $\mathbf{c}_a^*[i][j] = 0$ for $i \neq i^*$ and $j \neq j^*$. We prove the sufficiency of condition of (14) for correct recovery of the classes of signal and attack by contradiction. Let us assume that there exist $\mathbf{c}_s[i] \neq 0$ for $i \neq i^*$ and $\mathbf{c}_a^*[i][j] \neq 0$ for $i \neq^*, j \neq j^*$ and define a vector $\tilde{\mathbf{x}}$ as,

$$\tilde{\mathbf{x}} = \mathbf{x}' - \mathbf{D}_s[i^*]\mathbf{c}_s^*[i^*] + \mathbf{D}_a[i^*][j^*]\mathbf{c}_a[i^*][j^*] = \sum_{i \neq i^*} \mathbf{D}_s[i]\mathbf{c}_s^*[i] + \sum_{i \in \mathcal{I}, j \neq j*} \mathbf{D}_a[i][j]\mathbf{c}_a^*[i][j]. \tag{24}$$

Since $\mathbf{x}' \in \mathcal{S}_{i^*}^{\mathbf{x}} \oplus \mathcal{S}_{i^*j^*}^{\boldsymbol{\delta}}$, from (24) we deduce that $\tilde{\mathbf{x}}$ will have a representation on $\mathcal{S}_{i^*}^{\mathbf{x}} \oplus \mathcal{S}_{i^*j^*}^{\boldsymbol{\delta}}$, which will be a feasible solution of the following block-sparse optimization problem.

$$\begin{aligned} \{\hat{\mathbf{c}}_s^{*,\tilde{x}}, \hat{\mathbf{c}}_a^{*,\tilde{x}}\} \equiv \underset{\mathbf{c}_s, \mathbf{c}_a}{\arg\min} \|\mathbf{c}_s[i^*]\|_2 + \|\mathbf{c}_a[i^*][j^*]\|_2 \\ \text{s.t. } \tilde{\mathbf{x}} = \mathbf{D}_s[i^*]\mathbf{c}_s[i^*] + \mathbf{D}_a[i^*][j^*]\mathbf{c}_a[i^*][j^*], \end{aligned} \tag{25}$$

where $\hat{\mathbf{c}}_s^*, \hat{\mathbf{c}}_a^*$ are the correct-class minimum ℓ_1/ℓ_2 norm vectors supported on $\hat{\mathbf{c}}_s^*[i^*]$ and $\hat{\mathbf{c}}_a^*[i^*][j^*]$. Moreover, from the (24) we can also see that $\tilde{\mathbf{x}}$ is also belong to the span of the union of subspaces of remaining blocks of the dictionaries and hence the following problem

$$\begin{aligned} \{\tilde{\mathbf{c}}_s^{*,\tilde{x}}, \tilde{\mathbf{c}}_a^{*,\tilde{x}}\} \equiv \underset{\mathbf{c}_s, \mathbf{c}_a}{\arg\min} \sum_{i \in \mathcal{I} \setminus \{i^*\}} \|\mathbf{c}_s[i]\|_2 + \sum_{i \in \mathcal{I}, j \in \mathcal{J} \setminus \{j^*\}} \|\mathbf{c}_a[i][j]\|_2 \\ \text{s.t. } \tilde{\mathbf{x}} = \sum_{i \in \mathcal{I} \setminus \{i^*\}} \mathbf{D}_s[i]\mathbf{c}_s[i] + \sum_{i \in \mathcal{I}, j \in \mathcal{J} \setminus \{j^*\}} \mathbf{D}_a[i][j]\mathbf{c}_a[i][j] \end{aligned} \tag{26}$$

with $\mathcal{I} = \{1, 2, \ldots, r\}$ and $\mathcal{J} = \{1, 2, \ldots, a\}$, will have feasible solutions. From (24), we can get,

$$\begin{aligned} \mathbf{x}' &= \tilde{\mathbf{x}} + \mathbf{D}_s[i^*]\mathbf{c}_s^*[i^*] + \mathbf{D}_a[i^*][j^*]\mathbf{c}_a^*[i^*][j^*] \\ &= \mathbf{D}_s[i^*]\left(\hat{\mathbf{c}}_s^{*,\tilde{x}}[i^*] + \mathbf{c}_s^*[i^*]\right) + \mathbf{D}_a[i^*][j^*]\left(\hat{\mathbf{c}}_a^{*,\tilde{x}}[i^*][j^*] + \mathbf{c}_a^*[i^*][j^*]\right) \end{aligned} \tag{27}$$

From (27) we can see that vectors the pair of vectors $\hat{\mathbf{c}}_s^{*,\tilde{x}} + \mathbf{c}_s^*$ supported on the i^*th block and $\hat{\mathbf{c}}_a^{*,\tilde{x}} + \mathbf{c}_a^*$, supported on the (i^*, j^*)th block will be a feasible solution of (23). We will have,

$$\begin{aligned} &\|\hat{\mathbf{c}}_s^{*,\tilde{x}}[i^*] + \mathbf{c}_s^*[i^*]\|_2 + \|\hat{\mathbf{c}}_a^{*,\tilde{x}}[i^*][j^*] + \mathbf{c}_a^*[i^*][j^*]\|_2 \leq \\ &\|\hat{\mathbf{c}}_s^{*,\tilde{x}}[i^*]\|_2 + \|\mathbf{c}_s^*[i^*]\|_2\|_2 + \|\hat{\mathbf{c}}_s^{*,\tilde{x}}[i^*][j^*]\|_2 + \|\mathbf{c}_a^*[i^*][j^*]\|_2 \\ &< \|\tilde{\mathbf{c}}_s^{*,\tilde{x}}\|_{1,2} + \|\mathbf{c}_s^*[i^*]\|_2 + \|\tilde{\mathbf{c}}_a^{*,\tilde{x}}\|_{1,2} + \|\mathbf{c}_a^*[i^*][j^*]\|_2 \leq \\ \sum_{i \neq i^*} \|\mathbf{c}_s^*[i]\|_2 + \|\mathbf{c}_s^*[i^*]\|_2\|_2 + \sum_{i \in \mathcal{I}, j \neq j*} \|\mathbf{c}_a^*[i][j]\|_2 + \|\|\mathbf{c}_a^*[i^*][j^*]\|_2 &= \|\mathbf{c}_s^*\|_{1,2} + \|\mathbf{c}_a^*\|_{1,2}, \end{aligned} \tag{28}$$

where the second to the last inequality comes for the condition (14) of the Proposition. The last inequality in (28) appears due to optimality of $\tilde{\mathbf{c}}_s^*, \tilde{\mathbf{c}}_a^*$ in (26) and the fact that a vector supported on the blocks of $\mathbf{c}_s^*[i]$ for $i \neq i^*$ and $\mathbf{c}_a^*[i][j]$ for $i \neq i^*$,

$j \neq j^*$ is also a feasible solution of (26), yet not optimal. Hence we have arrived at a contradiction since by optimality of $\mathbf{c}_s^*, \mathbf{c}_a^*$ the inequality $\|\hat{\mathbf{c}}_s^{*,\tilde{x}} + \mathbf{c}_s^*\|_{1,2} + \|\hat{\mathbf{c}}_a^{*,\tilde{x}} + \mathbf{c}_a^*\|_{1,2} < \|\mathbf{c}_s^*\|_{1,2} + \|\mathbf{c}_a^*\|_{1,2}$ can not be true. We thus proved that the sufficiency of the condition (14) for correct recovery of signal and attack classes.

($\Longleftarrow$) Let first define $\hat{\mathbf{c}}_s^{*,\mathbf{x}'}, \hat{\mathbf{c}}_a^{*,\mathbf{x}'}, \tilde{\mathbf{c}}_s^{*,\mathbf{x}'}, \tilde{\mathbf{c}}_a^{*,\mathbf{x}'}$ as

$$\begin{aligned} \{\hat{\mathbf{c}}_s^{*,\tilde{x}}, \hat{\mathbf{c}}_a^{*,\tilde{x}}\} \equiv \underset{\mathbf{c}_s,\mathbf{c}_a}{\arg\min} \, & \|\mathbf{c}_s[i^*]\|_2 + \|\mathbf{c}_a[i^*][j^*]\|_2 \\ \text{s.t. } \mathbf{x}' = & \mathbf{D}_s[i^*]\mathbf{c}_s[i^*] + \mathbf{D}_a[i^*][j^*]\mathbf{c}_a[i^*][j^*], \end{aligned} \tag{29}$$

and the *wrong-class minimum* ℓ_1/ℓ_2 *norm vectors* $\tilde{\mathbf{c}}_s^*, \tilde{\mathbf{c}}_a^*$ as,

$$\begin{aligned} \{\tilde{\mathbf{c}}_s^{*,\mathbf{x}'}, \tilde{\mathbf{c}}_a^{*,\mathbf{x}'}\} \equiv \underset{\mathbf{c}_s,\mathbf{c}_a}{\arg\min} & \sum_{i \in \mathcal{I}\setminus\{i^*\}} \|\mathbf{c}_s[i]\|_2 + \sum_{i \in \mathcal{I}, j \in \mathcal{J}\setminus\{j^*\}} \|\mathbf{c}_a[i][j]\|_2 \\ \text{s.t. } \mathbf{x}' = & \sum_{i \in \mathcal{I}\setminus\{i^*\}} \mathbf{D}_s[i]\mathbf{c}_s[i] + \sum_{i \in \mathcal{I}, j \in \mathcal{J}\setminus\{j^*\}} \mathbf{D}_a[i][j]\mathbf{c}_a[i][j] \end{aligned} \tag{30}$$

Recall that the correct classes of the signal and the attack for an $\mathbf{x}' \in \mathcal{S}_{i^*}^{\mathbf{x}} \oplus \mathcal{S}_{i^*j^*}^{\boldsymbol{\delta}}$ can be recovered when the optimal $\mathbf{c}_s^*, \mathbf{c}_a^*$ are non-zero only at blocks $\mathbf{c}_s^*[i^*]$ and $\mathbf{c}_a^*[i^*][j^*]$. In that case, it also holds that $\|\mathbf{c}_s^*\|_{1,2} + \|\mathbf{c}_a^*\|_{1,2} = \|\hat{\mathbf{c}}_s^{*,\mathbf{x}'}\|_{1,2} + \|\hat{\mathbf{c}}_a^{*,\mathbf{x}'}\|_{1,2}$. We will show that if the correct classes of the signal and the attack can be recovered for $\mathbf{x}'$ then the condition (14) is true.

For that we assume that the solution $\tilde{\mathbf{c}}_s^{*,\mathbf{x}'}, \tilde{\mathbf{c}}_a^{*,\mathbf{x}'}$ is also feasible for problem (23) otherwise condition (14) is trivially satisfied since the RHS of (14) becomes $+\infty$.

Assume now that condition (14) is not true, i.e.,

$$\|\hat{\mathbf{c}}_s^{*,\tilde{x}}\|_{1,2} + \|\hat{\mathbf{c}}_a^{*,\tilde{x}}\|_{1,2} \geq \|\tilde{\mathbf{c}}_s^{*,\mathbf{x}'}\|_{1,2} + \|\tilde{\mathbf{c}}_a^{*,\mathbf{x}'}\|_{1,2} \tag{31}$$

that will imply,

$$\|\mathbf{c}_s^*\|_{1,2} + \|\mathbf{c}_a^*\|_{1,2} \geq \|\tilde{\mathbf{c}}_s^{*,\mathbf{x}'}\|_{1,2} + \|\tilde{\mathbf{c}}_a^{*,\mathbf{x}'}\|_{1,2} \tag{32}$$

and from optimality of $\mathbf{c}_s^*, \mathbf{c}_a^*$ and feasibility of $\hat{\mathbf{c}}_s^{*,\mathbf{x}'}, \hat{\mathbf{c}}_a^{*,\mathbf{x}'}$ at problem (23), we will have that equality will hold, i.e.,

$$\|\mathbf{c}_s^*\|_{1,2} + \|\mathbf{c}_a^*\|_{1,2} = \|\tilde{\mathbf{c}}_s^{*,\mathbf{x}'}\|_{1,2} + \|\tilde{\mathbf{c}}_a^{*,\mathbf{x}'}\|_{1,2}. \tag{33}$$

The latter means that there will be an optimal solution $\{\tilde{\mathbf{c}}_s^{*,\mathbf{x}'}, \tilde{\mathbf{c}}_a^{*,\mathbf{x}'}\}$ of (23) with non-zero blocks at indices corresponding to wrong classes of the signal and the attack when (31) holds true (i.e. condition (14) is false) which contradicts the initial assumption for the correct recovery of the classes of the signal and the attack. □

Let $\mathcal{D}_{i^*j^*}$ be the set of atoms, which contains the columns of the blocks of dictionaries of the signal and the attack that correspond to the correct classes i.e., $[\mathbf{D}_s[i^*], \mathbf{D}_a[i^*][j^*]]$. Recall from (19) that the relative polar set of $\pm\mathcal{D}_{i^*j^*}$ induced by the $\ell_{1,2}$ norm is given as,

$$\mathcal{K}_{\ell_{1,2}}^o(\pm\mathcal{D}_{i^*j^*}) = \{\mathbf{v} \in \operatorname{span}(\mathcal{S}_{i^*} \cup \mathcal{S}_{i^*j^*}) : \frac{1}{\sqrt{m}}\|[\mathbf{D}_s[i^*], \mathbf{D}_a[i^*][j^*]]^\top \mathbf{v}\|_{\infty,2} \leq 1\} \tag{34}$$

where $\operatorname{span}(\mathcal{S}_{i^*} \cup \mathcal{S}_{i^*j^*})$ is the column-space of $[\mathbf{D}_s[i^*], \mathbf{D}_a[i^*][j^*]]^\top]$. Next we define the circumradius of a convex body.

Definition A.1. (Circumradius) The circumradius of a convex body $\mathcal{P}$ denoted as $R(\mathcal{P})$ is defined as the radius of the smallest euclidean ball containing $\mathcal{P}$.

In our case, we will use the circumradius of the convex hull of the set $\mathcal{K}_{\ell_{1,2}}^o(\pm\mathcal{D}_{i^*j^*})$, denoted as $R(\mathcal{K}_{\ell_{1,2}}^o(\pm\mathcal{D}_{i^*j^*}))$.

Lemma A.2 shows the relationship between the covering radius of a set induced be $\ell_{1,2}$ norm and corresponding circumradius of its relative polar set.

Lemma A.2. *It holds that* $\cos(\gamma_{1,2}(\pm\mathcal{D})) = \frac{1}{R(\mathcal{K}_{\ell_{1,2}}^o(\pm\mathcal{D}))}$.

Proof. Our proof is based on that of the relevant result for the sparse recovery given in (Robinson et al., 2019). The covering radius $\gamma_{1,2}(\mathcal{D})$ is defined as

$$\gamma_{1,2}(\pm\mathcal{D}) = \sup\{\theta_{1,2}(\mathbf{v}, \pm\mathcal{D}), \mathbf{v} \in \mathbb{S}^{n-1} \cap \text{span}(\mathcal{D})\} = \sup_{\mathbf{v}\in\mathbb{S}^{n-1}\cap\text{span}(\mathcal{D})} \{\cos^{-1}\left(\frac{1}{\sqrt{m}}\|\mathbf{D}^\top\mathbf{v}\|_{\infty,2}\right)\} \tag{35}$$

Getting the cosine of $\gamma_{1,2}(\pm\mathcal{D})$ we have,

$$\cos\left(\gamma_{1,2}(\pm\mathcal{D})\right) = \cos\left(\sup_{\mathbf{v}\in\mathbb{S}^{n-1}\cap\text{span}(\mathcal{D})} \{\cos^{-1}\left(\frac{1}{\sqrt{m}}\|\mathbf{D}^\top\mathbf{v}\|_{\infty,2}\right)\}\right) = \inf_{\mathbf{v}\in\mathbb{S}^{n-1}\cap\text{span}(\mathcal{D})} \frac{1}{\sqrt{m}}\|\mathbf{D}^\top\mathbf{v}\|_{\infty,2} \tag{36}$$

The circumradius $R(\mathcal{K}^o_{\ell_{1,2}}(\pm\mathcal{D}))$ of the relative polar set $\mathcal{K}^o_{\ell_{1,2}}(\pm\mathcal{D})$ is given by,

$$R(\mathcal{K}^o_{\ell_{1,2}}(\pm\mathcal{D})) = \sup\{\|\mathbf{v}\|_2 : \frac{1}{\sqrt{m}}\|\mathbf{D}^\top\mathbf{v}\|_{\infty,2} \leq 1, \mathbf{v} \in \text{span}(\mathcal{D})\} \tag{37}$$

We want to prove that,

$$\inf_{\mathbf{v}\in\mathbb{S}^{n-1}\cap\text{span}(\mathcal{D})} \frac{1}{\sqrt{m}}\|\mathbf{D}^\top\mathbf{v}\|_{\infty,2} = \frac{1}{\sup\{\|\mathbf{v}\|_2 : \frac{1}{\sqrt{m}}\|\mathbf{D}^\top\mathbf{v}\|_{\infty,2} \leq 1, \mathbf{v} \in \text{span}(\mathcal{D})\}} \tag{38}$$

Let $\mathbf{w}^*$ and $\mathbf{v}^*$ be optimal solutions of the optimization problems appearing at the LHS and RHS of (38), respectively. Let us now define $\bar{\mathbf{v}} = \frac{\sqrt{m}\mathbf{w}^*}{\|\mathbf{D}^\top\mathbf{w}^*\|_{\infty,2}}$ and $\bar{\mathbf{w}} = \frac{\mathbf{v}^*}{\|\mathbf{v}^*\|_2}$. We have that $\|\mathbf{w}^*\|_2 = 1$ and $\bar{\mathbf{v}}$ satisfies the constraints appearing the optimization problem at the RHS of (38) i.e., $\frac{1}{\sqrt{m}}\|\mathbf{D}^\top\bar{\mathbf{v}}\|_{\infty,2} \leq 1$ and $\bar{\mathbf{v}} \in \text{span}(\mathcal{D})$. Hence, we will have,

$$\|\bar{\mathbf{v}}\|_2 = \frac{\sqrt{m}\|\mathbf{w}^*\|_2}{\|\mathbf{D}^\top\mathbf{w}^*\|_{\infty,2}} = \frac{\sqrt{m}}{\|\mathbf{D}^\top\mathbf{w}^*\|_{\infty,2}} \leq \|\mathbf{v}^*\|_2 \tag{39}$$

where the last inequality arises by the fact that $\bar{\mathbf{v}}$ is a feasible but not optimal solution of the problem at the RHS of (38). Moreover, for $\bar{\mathbf{w}} = \frac{\mathbf{v}^*}{\|\mathbf{v}^*\|_2}$ we have that $\bar{\mathbf{w}} \in \mathbb{S}^{n-1}$ and $\bar{\mathbf{w}} \in \text{span}(\mathcal{D})$. Therefore, $\bar{\mathbf{w}}$ satisfies the constraints and it will be a feasible solution of of the optimization problem at the LHS of (38). From that we can deduce that

$$\frac{1}{\sqrt{m}}\|\mathbf{D}^\top\bar{\mathbf{w}}\|_{\infty,2} = \frac{1}{\sqrt{m}}\frac{\|\mathbf{D}^\top\mathbf{v}^*\|_{\infty,2}}{\|\mathbf{v}^*\|_2} \leq \frac{1}{\|\mathbf{v}^*\|_2} \tag{40}$$

From optimality of $\mathbf{w}^*$ at the LHS of (38) we will have

$$\frac{1}{\|\mathbf{v}^*\|_2} \geq \frac{1}{\sqrt{m}}\|\mathbf{D}^\top\mathbf{w}^*\|_{\infty,2} \rightarrow \frac{\sqrt{m}}{\|\mathbf{D}^\top\mathbf{w}^*\|_{\infty,2}} \geq \|\mathbf{v}^*\|_2 \tag{41}$$

By combining (39) and (41) we get the result. □

A.2. Proof of Theorem 5.5

Without loss of generality for the proofs of theorems 5.5 and 5.8 we scale the dictionaries $\mathbf{D}_s, \mathbf{D}_a$ by $\frac{1}{\sqrt{m}}$, where m is the size of the blocks. The primal problem denoted as $P(\frac{1}{\sqrt{m}}\mathbf{D}_s, \frac{1}{\sqrt{m}}\mathbf{D}_a, \mathbf{x}')$ is given as,

$$P(\frac{1}{\sqrt{m}}\mathbf{D}_s, \frac{1}{\sqrt{m}}\mathbf{D}_a, \mathbf{x}') \coloneqq \arg\min \|\mathbf{c}_s\|_{1,2} + \|\mathbf{c}_a\|_{1,2} \quad \text{s.t. } \mathbf{x}' = \frac{1}{\sqrt{m}}\mathbf{D}_s\mathbf{c}_s + \frac{1}{\sqrt{m}}\mathbf{D}_a\mathbf{c}_a \tag{42}$$

and the dual of (42),

$$D(\frac{1}{\sqrt{m}}\mathbf{D}_s, \frac{1}{\sqrt{m}}\mathbf{D}_a, \mathbf{x}') \coloneqq \arg\max\langle\mathbf{w}, \mathbf{x}'\rangle \quad \text{s.t. } \|[\frac{1}{\sqrt{m}}\mathbf{D}_s, \frac{1}{\sqrt{m}}\mathbf{D}_a]^\top\mathbf{w}\|_{\infty,2} \leq 1 \tag{43}$$

where $\mathbf{w}$ is the dual variable.

Proof. We will prove the theorem by showing that the condition

$$\gamma_{1,2}(\pm\mathcal{D}_{i^*j^*}) < \theta_{1,2}(\mathcal{S}_{i^*} \cup \mathcal{S}_{i^*j^*}, \pm\mathcal{D}^-_{i^*j^*}) \tag{44}$$

implies the necessary and sufficient condition of Proposition 1, i.e.,

$$\|\hat{\mathbf{c}}^*_s\|_{1,2} + \|\hat{\mathbf{c}}^*_a\|_{1,2} < \|\tilde{\mathbf{c}}^*_s\|_{1,2} + \|\tilde{\mathbf{c}}^*_a\|_{1,2} \tag{45}$$

Let us focus on $\frac{1}{\sqrt{m}}\mathbf{D}_s[i^*]$, $\frac{1}{\sqrt{m}}\mathbf{D}_a[i^*][j^*]$ and denote as $p(\frac{1}{\sqrt{m}}\mathbf{D}_s[i^*], \frac{1}{\sqrt{m}}\mathbf{D}_a[i^*][j^*], \mathbf{x}')$, $d(\frac{1}{\sqrt{m}}\mathbf{D}_s[i^*], \frac{1}{\sqrt{m}}\mathbf{D}_a[i^*][j^*], \mathbf{x}')$ the values of the objective functions of the primal and dual problems, respectively. Due to convexity, strong duality holds, hence we have,

$$p(\frac{1}{\sqrt{m}}\mathbf{D}_s[i^*], \frac{1}{\sqrt{m}}\mathbf{D}_a[i^*][j^*]\mathbf{x}') = d(\frac{1}{\sqrt{m}}\mathbf{D}_s[i^*], \frac{1}{\sqrt{m}}\mathbf{D}_a[i^*][j^*], \mathbf{x}') = \langle \mathbf{w}, \mathbf{x}' \rangle \tag{46}$$

Let us now decompose the dual variable $\mathbf{w} \in \mathbb{R}^n$ as $\mathbf{w} = \mathbf{w}^\perp + \mathbf{w}^\parallel$, where $\mathbf{w}^\parallel \in \mathcal{S}_{i^*} \cup \mathcal{S}_{i^*j^*}$ and $\mathbf{w}^\perp \perp \mathbf{w}^\parallel$. For (46) we have,

$$\begin{aligned} p(\frac{1}{\sqrt{m}}\mathbf{D}_s[i^*], \frac{1}{\sqrt{m}}\mathbf{D}_a[i^*][j^*], \mathbf{x}') = d(\frac{1}{\sqrt{m}}\mathbf{D}_s[i^*], \frac{1}{\sqrt{m}}\mathbf{D}_a[i^*][j^*], \mathbf{x}') = \\ \langle \mathbf{w}, \mathbf{x}' \rangle = \langle \mathbf{w}^\parallel, \mathbf{x}' \rangle \leq \|\mathbf{w}^\parallel\|_2 \|\mathbf{x}'\|_2 \leq \|\mathbf{x}'\|_2 \frac{1}{\cos(\gamma_{1,2}(\pm\mathcal{D}_{i^*j^*}))} \end{aligned} \tag{47}$$

where the last inequality follows from Lemma (A.2), by taking into account that $\mathbf{w}^\parallel$ a) belongs to the dual polar set $\mathcal{K}^o_{\ell_{1,2}}(\mathcal{D}_{i^*j^*})$ b) $\mathbf{w}^\parallel \in \mathcal{S}_{i^*} \cup \mathcal{S}_{i^*j^*}$ and hence is a feasible solution of the optimization problem at the RHS of (38).

Let us now focus the primal problem,

$$p(\frac{1}{\sqrt{m}}\mathbf{D}^-_s, \frac{1}{\sqrt{m}}\mathbf{D}^-_a\mathbf{x}') = \min_{\mathbf{c}_s, \mathbf{c}_s} \|\mathbf{c}_s\|_{1,2} + \|\mathbf{c}_a\|_{1,2} \;\text{ s.t. }\; \mathbf{x}' = \frac{1}{\sqrt{m}}\mathbf{D}^-_s\mathbf{c}_s + \frac{1}{\sqrt{m}}\mathbf{D}^-_a\mathbf{c}_a \tag{48}$$

and assume that there exist solutions $\mathbf{c}^*_s, \mathbf{c}^*_a \in P(\frac{1}{\sqrt{m}}\mathbf{D}^-_s, \frac{1}{\sqrt{m}}\mathbf{D}^-_a\mathbf{x}'))$ such that $\mathbf{x}' = \mathbf{D}^-_s\mathbf{c}^*_s + \mathbf{D}^-_a\mathbf{c}^*_a$. We will have,

$$\begin{aligned} \|\mathbf{x}'\|_2^2 = \mathbf{x}'^\top \left(\frac{1}{\sqrt{m}}\mathbf{D}^-_s\mathbf{c}^*_s + \frac{1}{\sqrt{m}}\mathbf{D}^-_a\mathbf{c}^*_a\right) \\ \leq \|\mathbf{D}^{-,\top}_s \frac{\mathbf{x}'}{\|\mathbf{x}'\|_2}\|_{\infty,2}\|\mathbf{x}'\|_2\|\mathbf{c}^*_s\|_{1,2} + \|\mathbf{D}^{-,\top}_a \frac{\mathbf{x}'}{\|\mathbf{x}'\|_2}\|_{\infty,2}\|\mathbf{x}'\|_2\|\mathbf{c}^*_a\|_{1,2} \\ \leq \cos(\theta_{1,2}(\frac{\mathbf{x}'}{\|\mathbf{x}'\|_2}, \mathcal{D}^-_{i^*,j^*}))\|\mathbf{x}'\|_2 \underbrace{(\|\mathbf{c}_s\|_{1,2} + \|\mathbf{c}_a\|_{1,2})}_{p(\frac{1}{\sqrt{m}}\mathbf{D}^-_s, \frac{1}{\sqrt{m}}\mathbf{D}^-_a\mathbf{x}')} \\ \rightarrow p(\frac{1}{\sqrt{m}}\mathbf{D}^-_s, \frac{1}{\sqrt{m}}\mathbf{D}^-_a\mathbf{x}') \geq \frac{\|\mathbf{x}'\|_2}{\cos(\theta_{1,2}(\frac{\mathbf{x}'}{\|\mathbf{x}'\|_2}, \mathcal{D}^-_{i^*,j^*}))} \end{aligned} \tag{49}$$

By combining (47) with (49) we get,

$$\begin{aligned} \|\mathbf{x}'\|_2 \frac{1}{\cos(\gamma_{1,2}(\pm\mathcal{D}_{i^*j^*}))} < \frac{\|\mathbf{x}'\|_2}{\cos(\theta_{1,2}(\frac{\mathbf{x}'}{\|\mathbf{x}'\|_2}, \mathcal{D}^-_{i^*,j^*}))} \rightarrow \\ \gamma_{1,2}(\pm\mathcal{D}_{i^*j^*})) < \theta_{1,2}(\frac{\mathbf{x}'}{\|\mathbf{x}'\|_2}, \mathcal{D}^-_{i^*,j^*})) \end{aligned} \tag{50}$$

and hence the last inequality is a sufficient condition for (45). □

A.3. Proof of Theorem 5.8

We first prove prove the following Lemma.

Lemma A.3. *If the Dual Recovery Condition holds i.e.,* $\gamma_{1,2}(\mathcal{D}_{i^*j^*}) < \theta_{1,2}\left(\mathcal{A}\left(\mathcal{D}_{i^*j^*}\right), \pm\mathcal{D}^-_{i^*j^*}\right)$ *then* $\forall \mathbf{v} \in \mathcal{A}(\mathcal{D}_{i^*j^*})$ *it holds* $\frac{1}{\sqrt{m}}\|[\mathbf{D}_s^{-,\top}, \mathbf{D}_a^{-,\top}]\mathbf{v}\|_{\infty,2} < 1$.

Proof. We have that $\forall \mathbf{v} \in \mathcal{A}(\mathcal{D}_{i^*j^*})$ it holds $\frac{1}{\sqrt{m}}\|[\mathbf{D}_s[i^*], \mathbf{D}_a[i^*][j^*]]^\top \mathbf{v}\|_{\infty,2} \leq 1$. Hence, due to Lemma A.2 we have that $\|\mathbf{v}\|_2 \leq \frac{1}{\cos(\gamma_{1,2}(\mathcal{D}_{i^*j^*}))}$. We will have,

$$\frac{1}{\sqrt{m}}\|[\mathbf{D}_s^-, \mathbf{D}_a^-]^\top \mathbf{v}\|_{\infty,2} = \frac{1}{\sqrt{m}}\|[\mathbf{D}_s^-, \mathbf{D}_a^-]^\top \frac{\mathbf{v}}{\|\mathbf{v}\|_2}\|_{\infty,2}\|\mathbf{v}\|_2 \leq \frac{\cos(\theta_{1,2}(\mathbf{v}, \mathcal{D}^-_{i^*j^*}))}{\cos(\gamma_{1,2}(\mathcal{D}_{i^*j^*}))} < 1 \tag{51}$$

□

Next we will prove the following Lemma,

Lemma A.4. *If* $\frac{1}{\sqrt{m}}\|[\mathbf{D}_s^{-,\top}, \mathbf{D}_a^{-,\top}]\mathbf{v}\|_{\infty,2} < 1 \ \forall \mathbf{v} \in \mathcal{A}(\mathcal{D}_{i^*j^*})$ *then the necessary and sufficient condition for successful recovery of the correct class of the signal and the attack given in* (14) *i.e.,* $p(\frac{1}{\sqrt{m}}\mathbf{D}_s[i^*], \frac{1}{\sqrt{m}}\mathbf{D}_a[i^*][j^*], \mathbf{x}') < p(\frac{1}{\sqrt{m}}\mathbf{D}_s^-, \frac{1}{\sqrt{m}}\mathbf{D}_a^-, \mathbf{x}')$ *holds.*

Proof. Let us define the following constrained optimization problem,

$$\max\langle \mathbf{x}', \mathbf{w}\rangle \quad \text{s.t.} \ \frac{1}{\sqrt{m}}\|[\mathbf{D}_s[i^*], \mathbf{D}_a[i^*][j^*]]^\top, \mathbf{w}\|_{\infty,2} \leq 1, \mathbf{w} \in \text{span}(\mathcal{D}_{i^*j^*}) \tag{52}$$

Using standard convex optimization arguments we can deduce that the optimal solution of the above problem $\mathbf{w}$ will be an extreme point of the convex set defined by $\{\mathbf{w} : \frac{1}{\sqrt{m}}\|[\mathbf{D}_s[i^*], \mathbf{D}_a[i^*][j^*]]^\top \mathbf{w}\|_{\infty,2} \leq 1, \mathbf{w} \in \text{span}(\mathcal{D}_{i^*j^*})\}$ hence $\mathbf{w}$ will belong to the set of dual points $\mathcal{A}(\mathcal{D}_{i^*j^*})$. Let us now state the following problem,

$$\max\langle \mathbf{x}', \mathbf{w}\rangle \quad \text{s.t.} \ \frac{1}{\sqrt{m}}\|[\mathbf{D}_s[i^*], \mathbf{D}_a[i^*][j^*]]^\top \mathbf{w}\|_{\infty,2} \leq 1, \tag{53}$$

Note that (53) does not constrain $\mathbf{w}$ to belong in $\text{span}(\mathcal{D}_{i^*j^*})$. As a result, there might be optimal solutions $\mathbf{w}$ not in $\text{span}(\mathcal{D}_{i^*j^*})$. However, we can deduce that there will always exist a $\mathbf{w}$ $\text{span}(\mathcal{D}_{i^*j^*})$ that will be an optimal solution and a dual point. This can be deduced if we express a candidate solution $\mathbf{w}^*$ as $\mathbf{w}^* = \mathbf{w}^\perp + \mathbf{w}^\parallel$ where $\mathbf{w}^\parallel \in \text{span}(\mathcal{D}_{i^*j^*})$.

Let us now assume that there exists a $\{\mathbf{c}_s, \mathbf{c}_a\} \in P(\frac{1}{\sqrt{m}}\mathbf{D}_s^-, \frac{1}{\sqrt{m}}\mathbf{D}_a^-, \mathbf{x}')$. We will have $\mathbf{x}' = \frac{1}{\sqrt{m}}\mathbf{D}_s^-\mathbf{c}_s + \frac{1}{\sqrt{m}}\mathbf{D}_a^-\mathbf{c}_a$. On the other hand, there will be $\mathbf{w}^* \in \mathcal{A}(\mathcal{D}_{i^*j^*})$ that will be a dual optimal solution of $D(\frac{1}{\sqrt{m}}\mathbf{D}_s[i^*], \frac{1}{\sqrt{m}}\mathbf{D}_a[i^*][j^*], \mathbf{x}')$ i.e.,

$$p(\frac{1}{\sqrt{m}}\mathbf{D}_s[i^*], \frac{1}{\sqrt{m}}\mathbf{D}_a[i^*][j^*], \mathbf{x}') = d(\frac{1}{\sqrt{m}}\mathbf{D}_s[i^*], \frac{1}{\sqrt{m}}\mathbf{D}_a[i^*][j^*], \mathbf{x}') = \langle \mathbf{w}^*, \mathbf{x}'\rangle = \langle \mathbf{w}^*, \frac{1}{\sqrt{m}}(\mathbf{D}_s^-\mathbf{c}_s + \mathbf{D}_a^-\mathbf{c}_a)\rangle \leq$$
$$\frac{1}{\sqrt{m}}\|[\mathbf{D}_s^-, \mathbf{D}_a^-]^\top \mathbf{w}^*\|_{\infty,2}\left(\|\mathbf{c}_s\|_{1,2} + \|\mathbf{c}_a\|_{1,2}\right) < p(\frac{1}{\sqrt{m}}\mathbf{D}_s^-, \frac{1}{\sqrt{m}}\mathbf{D}_a^-, \mathbf{x}') \tag{54}$$

□

Theorem 5.8 is proved by combining Lemmas A.3 and A.4.

A.4. Derivation of the Active Set Homotopy Algorithm and Algorithm Details

Consider the optimization problem in Equation (11). We denote the objective as $L(\mathbf{x}', \mathbf{D}_s, \mathbf{D}_a, \lambda_s, \lambda_a)$. We can write the optimality conditions with respect to $\mathbf{c}_s$ and $\mathbf{c}_a$ for this problem. For any block i of $\mathbf{D}_s$, the optimality conditions with respect to $\mathbf{c}_s$ are:

$$\mathbf{D}_s[i]^T(\mathbf{x}' - \mathbf{D}_s\mathbf{c}_s^* - \mathbf{D}_a\mathbf{c}_a^*) = \lambda_s \frac{\mathbf{c}_s^*}{\|\mathbf{c}_s^*\|_2} \qquad \text{if } \mathbf{c}_s^*[i] \neq 0 \tag{55}$$

$$\left\|\mathbf{D}_s[i]^T(\mathbf{x}' - \mathbf{D}_s\mathbf{c}_s^* - \mathbf{D}_a\mathbf{c}_a^*)\right\|_2 \leq \lambda_s \qquad \text{if } \mathbf{c}_s^*[i] = 0 \tag{56}$$

Algorithm 1 Active Set Homotopy Algorithm

Results:$\hat{\mathbf{c}}_a^*, \hat{\mathbf{c}}_s^*$
Initialize : $\hat{\mathbf{c}}_s^0, \hat{\mathbf{c}}_a^0 \leftarrow \mathbf{0}, \mathbf{0}, T_s^0, T_a^0 \leftarrow \emptyset, \emptyset, k \leftarrow 1$
Set : $\gamma \in (0,1)$
while $T_s^{k+1} \not\subseteq T_s^k$ and $T_a^{k+1} \not\subseteq T_a^k$ **do**
  $\mathbf{o}^k \leftarrow \mathbf{x}' - \mathbf{D}_s[T_s^k]\hat{\mathbf{c}}_s^k[T_s^k] - \mathbf{D}_a[T_a^k]\hat{\mathbf{c}}_a^k[T_a^k]$
  $\lambda_s^k \leftarrow \gamma \cdot \max_i \left\|\mathbf{D}_s[i]^T\mathbf{o}^k\right\|_2$
  $\lambda_a^k \leftarrow \gamma \max_{i,j} \left\|\mathbf{D}_a[i][j]^T\mathbf{o}^k\right\|_2$
  $\hat{i}^k \leftarrow \arg\max_i \left\|\mathbf{D}_a[i]^T\mathbf{o}^k\right\|_2$
  $\hat{j}^k \leftarrow \arg\max_j \max_i \left\|\mathbf{D}_a[i][j]^T\mathbf{o}^k\right\|_2$
  Add $\hat{i}^k$ and $\hat{j}^k$ to T_s^k and T_a^k respectively.
  Solve problem (11) with any solver using $\mathbf{D}_s[T_s^k], \mathbf{D}_a[T_a^k], \lambda_s^k, \lambda_a^k$ and compute $\hat{\mathbf{c}}_s^{k+1}$ and $\hat{\mathbf{c}}_a^{k+1}$.
  $k \leftarrow k+1$
end while

For any block (i,j) of $\mathbf{D}_a$, the optimality conditions with respect to $\mathbf{c}_a$ are:

$$\mathbf{D}_a[i][j]^T(\mathbf{x}' - \mathbf{D}_s\mathbf{c}_s^* - \mathbf{D}_a\mathbf{c}_a^*) = \lambda_2 \frac{\mathbf{c}_a^*}{\|\mathbf{c}_a^*\|_2} \qquad \text{if } \mathbf{c}_a^*[i][j] \neq 0 \tag{57}$$

$$\left\|\mathbf{D}_a[i][j]^T(\mathbf{x}' - \mathbf{D}_s\mathbf{c}_s^* - \mathbf{D}_a\mathbf{c}_a^*)\right\|_2 \leq \lambda_2 \qquad \text{if } \mathbf{c}_a^*[i][j] = 0 \tag{58}$$

First, we derive a value of λ_s and λ_s such that the optimal $\mathbf{c}_s$ and $\mathbf{c}_a$ are the all-zero vectors.

Lemma A.5. *Let* $\lambda_s \geq \left\|\mathbf{D}_s^T\mathbf{x}'\right\|_{\infty,2} = \sup_i \left\|\mathbf{D}_s[i]^T\mathbf{x}'\right\|_2$ *and* $\lambda_a \geq \left\|\mathbf{D}_a^T\mathbf{x}'\right\|_{\infty,2} = \sup_{i,j} \left\|\mathbf{D}_a[i][j]^T\mathbf{x}'\right\|_2$. *Then, the values of* $\mathbf{c}_s^*$ *and* $\mathbf{c}_a^*$ *that minimize* $L(x', \mathbf{D}_s, \mathbf{D}_a, \lambda_s, \lambda_a))$ *are the all-zero vectors.*

Proof. We begin with the proof of showing that $\lambda_s = \left\|\mathbf{D}_s^T\mathbf{x}'\right\|_{\infty,2}$ is sufficient so that $\mathbf{c}_s^*$ is the all-zeroes vector. Looking at Equation (56), we see that for a block of $\mathbf{c}_s^*$ to be 0, a sufficient condition is that the norm of the gradient of the fitting term of the objective is less than λ_s. This immediately gives that if for all blocks i,

$$\left\|\mathbf{D}_s[i]^T(\mathbf{x}' - \mathbf{D}_s\mathbf{c}_s^* - \mathbf{D}_a\mathbf{c}_a^*)\right\|_2 \leq \lambda_s \tag{59}$$

then the optimal $\mathbf{c}_s^*$ must be 0 based on the optimality conditions. For simplicity in the proof, we will assume that the sufficient condition for $\mathbf{c}_a$ to be the zero vector holds, which will be shown after. This implies that if $\lambda_s \geq \left\|\mathbf{D}_s^T\mathbf{x}'\right\|_{\infty,2} = \sup_i \left\|\mathbf{D}_s[i]^T\mathbf{x}'\right\|_2$, then the $\mathbf{c}_s^*$ that minimizes $L(\mathbf{x}', \mathbf{D}_s, \mathbf{D}_a, \lambda_s, \lambda_a))$ is the all-zeros vector. The same argument applies for λ_a, for which we have that if $\lambda_a \geq \left\|\mathbf{D}_a^T\mathbf{x}'\right\|_{\infty,2} = \sup_{i,j} \left\|\mathbf{D}_a[i][j]^T\mathbf{x}'\right\|_2$, then $\mathbf{c}_a^*$ is the all-zeros vector. Jointly fixing both λ_s and λ_a, we have a sufficient condition for $\mathbf{c}_s^*$ and $\mathbf{c}_a^*$ being zero. □

The proof strategy of the above lemma suggests that if we knew the value of $\mathbf{c}_s^*$ and $\mathbf{c}_a^*$, then we can find a value of λ_s and λ_a such that minimizing $L(\mathbf{x}', \mathbf{D}_s, \mathbf{D}_a, \lambda_s, \lambda_a))$ yields $\mathbf{c}_s^*$ and $\mathbf{c}_a^*$; however, obviously, we do not know the value of $\mathbf{c}_s^*$ and $\mathbf{c}_a^*$. Namely, the value of the regularization parameters depends on the residual $\mathbf{x}' - \mathbf{D}_s\mathbf{c}_s^* - \mathbf{D}_a\mathbf{c}_a^*$, which we denote as $\mathbf{o}^*$ or the *oracle point*. The homotopy algorithm for solving LASSO ℓ_1 minimization problems proceeds by starting from the all-zeros solution and calculating the decrease in λ that results in one non-zero element added to the support of the optimal solution. This works because the optimal solution plotted as a function of the regularization parameter is piecewise linear. For the block-sparse optimization problem, also known as group-LASSO, it is well-known that the solution path is nonlinear (Yau & Hui, 2017). Thus, we use the natural heuristic of starting with the value of λ_s and λ_a that produces the all-zero vector, scaling the value by some hyperparameter $\gamma \in (0,1)$, and estimating the oracle point $\mathbf{o}$ by solving $L(\mathbf{x}', \mathbf{D}_s, \mathbf{D}_a, \gamma\lambda_s, \gamma\lambda_a)$. From $\mathbf{o}$, we can then again calculate a value of λ_s and λ_a and iterate. This alternating algorithm forms the basis of the active set homotopy algorithm. Since we begin from the all-zero vector and reduce λ_s and λ_a, we can maintain an active set of non-zero coordinates and only solve subproblems restricted to these non-zero blocks for efficiency

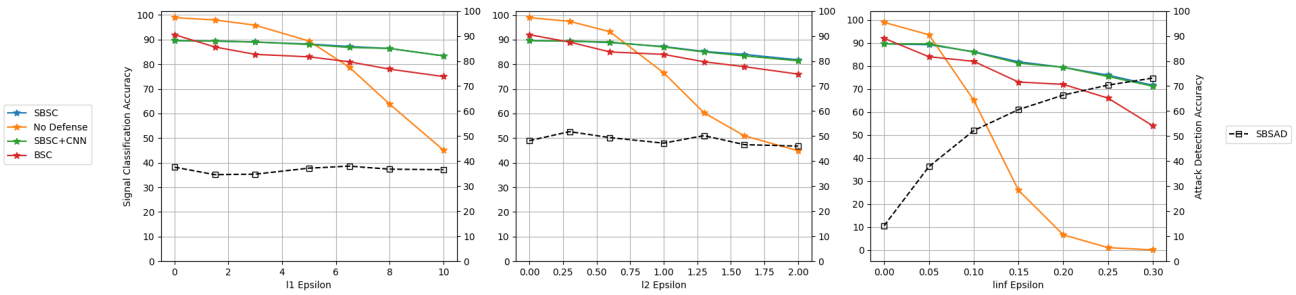

Figure 1. Results on the MNIST dataset with varying attack strength ϵ. The SBSC and SBSC+CNN curves show the accuracy of the structured block-sparse classifier and denoiser at predicting the correct class. Note that these curves overlap almost fully, thus both are not clearly visible. BSC denotes the naive block-sparse baseline. The SBSAD curve denotes the accuracy of the attack detector at predicting the correct attack type. Best viewed in color.

purposes. In Algorithm 1, we see the full algorithm detailed. Note that we overload the notation $\mathbf{D}_s[T_s]$ and $\mathbf{D}_a[T_a]$ to denote the submatrices of $\mathbf{D}_s$ and $\mathbf{D}_a$ corresponding to the block indices in the sets T_s and T_a.To solve the subproblems, we use the cvxpy package (Diamond & Boyd, 2016; Agrawal et al., 2018) with the SCS solver run for a maximum of 50 iterations.

A.5. Experimental Details

Layer Type	Size
Convolution + ReLU	$3 \times 3 \times 32$
Convolution + ReLU	$3 \times 3 \times 32$
Max Pooling	2×2
Convolution + ReLU	$3 \times 3 \times 64$
Convolution + ReLU	$3 \times 3 \times 64$
Max Pooling	2×2
Fully Connected + ReLU	200
Fully Connected + ReLU	200
Fully Connected + ReLU	10

Table 3. Network Architecture for the MNIST dataset

A.5.1. MNIST

The network architecture for the MNIST dataset is given in Table 3. The network on MNIST is trained using SGD for 50 epochs with learning rate 0.1, momentum 0.5, and batch size 128.

All PGD adversaries were generated using the Advertorch library. The ℓ_∞ PGD adversary ($\epsilon = 0.3$) used a step size $\alpha = 0.01$ and was run for 100 iterations. The ℓ_2 PGD adversary ($\epsilon = 2$) used a step size $\alpha = 0.1$ and was run for 200 iterations. The ℓ_1 PGD adversary ($\epsilon = 10$) used a step size $\alpha = 0.8$ and was run for 100 iterations. These hyperparameters are identical to the hyperparameters for the adversarial training baselines, to enable a fair comparison.

A.5.2. YALEB

For the YaleB dataset, we train a three-layer fully-connected network, where each hidden layer contains 256 neurons followed by a ReLU activation. We train this network using SGD for 75 epochs with learning rate 0.05, momentum 0.5, and batch size 128. All PGD adversaries were generated using the Advertorch library. The ℓ_∞ PGD adversary ($\epsilon = 0.1$) used a step size $\alpha = 0.003$ and was run for 100 iterations. The ℓ_2 PGD adversary ($\epsilon = 5$) used a step size $\alpha = 0.02$ and was run for 200 iterations. The ℓ_1 PGD adversary ($\epsilon = 15$) used a step size $\alpha = 1.0$ and was run for 100 iterations.

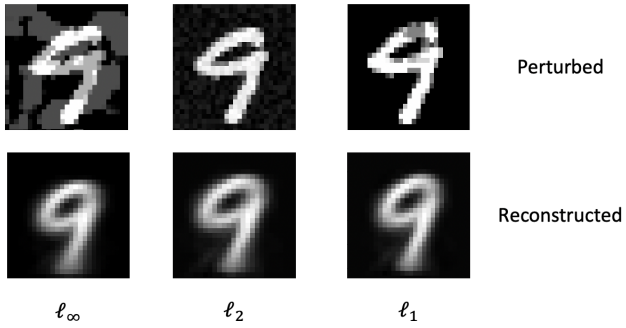

Figure 2. Qualitative experiments on the reconstruction of a digit from the MNIST dataset.

A.6. Extended Experiments on MNIST Dataset

Tradeoff between signal and attack detection. For ℓ_∞ attacks, the SBSC method does not serve as an effective defense compared to the baselines. One possible explanation for this phenomenon is the relationship between the SBSC and SBSAD methods, since both are predicted jointly from Algorithm 1. Specifically, in Figure 1, we see an explicit tradeoff between the choice of ϵ in terms of the signal classification and attack detection accuracy. As ϵ increases, we expect the attacks to be easier to distinguish among the family of attacks; however, as the noise increases, classifying the correct label becomes harder regardless of the accuracy of the predicted attack type. Note that in this figure, the dictionaries $\mathbf{D}_s$ and $\mathbf{D}_a$ are kept fixed using the same ϵ values as in Table 2, and only the ϵ of the attacked test images is varied. Additionally, Figure 1 demonstrates that explicitly modeling the perturbation and adding further structure to block-sparse optimization methods helps improve the accuracy of the block-sparse classifier, as our method outperforms the BSC method that only models $\mathbf{x}' = \mathbf{D}_s\mathbf{c}_s$. As ϵ increases, the method remains robust to perturbations, while the accuracy of the undefended network continues to degrade.

Qualitative results. Figure 2 shows an example of the reconstruction of one digit from the MNIST dataset using the SBSC+CNN method perturbed using all perturbation types. The SBSC method performs a smoothing that is able to denoise across various perturbation types and remove visible noise patterns in the corrupted image. The CNN classifier then correctly classifies the resulting denoised image.