# Structural Extensions of Basis Pursuit: Guarantees on Adversarial Robustness

Dávid Szeghy[2,3][a], Mahmoud Aslan[1][b], Áron Fóthi[1][c], Balázs Mészáros[1][d],
Zoltán Ádám Milacski[4][e] and András Lőrincz[1][f]

[1]*Department of Artificial Intelligence, Faculty of Informatics, ELTE Eötvös Loránd University,*
*1/A. Pázmány Péter sétány, Budapest, 1117, Hungary*

[2]*Department of Geometry, Faculty of Natural Sciences, ELTE Eötvös Loránd University,*
*1/C. Pázmány Péter sétány, Budapest, 1117, Hungary*

[3]*AImotive Inc., 18-22 Szépvölgyi út, Budapest, 1025, Hungary*

[4]*Former Member of Department of Artificial Intelligence, Faculty of Informatics, ELTE Eötvös Loránd University,*
*1/A. Pázmány Péter sétány, Budapest, 1117, Hungary*

Keywords: Sparse Coding, Group Sparse Coding, Stability Theory, Adversarial Attack.

Abstract: While deep neural networks are sensitive to adversarial noise, sparse coding using the Basis Pursuit (BP) method is robust against such attacks, including its multi-layer extensions. We prove that the stability theorem of BP holds upon the following generalizations: (i) the regularization procedure can be separated into disjoint groups with *different* weights, (ii) *neurons* or *full layers* may form groups, and (iii) the regularizer takes various generalized forms of the $\ell_1$ norm. This result provides the proof for the architectural generalizations of (Cazenavette et al., 2021) including (iv) an approximation of the complete architecture as a shallow sparse coding network. Due to this approximation, we settled to experimenting with shallow networks and studied their robustness against the Iterative Fast Gradient Sign Method on a synthetic dataset and MNIST. We introduce classification based on the $\ell_2$ norms of the groups and show numerically that it can be accurate and offers considerable speedups. In this family, linear transformer shows the best performance. Based on the theoretical results and the numerical simulations, we highlight numerical matters that may improve performance further. The proofs of our theorems can be found in the supplementary material*.

## 1 INTRODUCTION

Considerable effort has been devoted to overcoming the vulnerability of deep neural networks against '*white box*' adversarial attacks. These attacks have access to the network structure and the loss function. They work by modifying the input towards the sign of the gradient of the loss function (Goodfellow et al., 2014) that can spoil classification at very low levels of perturbations. Furthermore, this white box attack gives rise to successful transferable attacking samples to other networks of similar kinds (Liu et al., 2016), called '*black box attack*'. This underlines the need for network structures exhibiting robustness against white box adversarial attacks.

Sparse methods exploiting $\ell_1$ norm regularization and the Basis Pursuit (BP) algorithm (Figs. 1(a) and 1(c)) exhibit robustness against such attacks, including their multilayer Layered Basis Pursuit (LBP) extensions (Romano et al., 2020) (Fig. 1(d)). (Cazenavette et al., 2021) (Cazenavette et al., 2021) found a solution to the LBP's drawback that layered basis pursuit accumulates errors: they put forth an architectural generalization of LBP to modify the cascade of layered basis pursuit steps of the deep neural network in such a way that the entire network becomes an approximation to a single structured sparse coding problem that they call deep pursuit (Figs. 1(e) and 1(e$^*$)). Note that their generalization goes beyond the structure depicted in Fig. 1(e). This architectural generalization points to the relevance of a single sparse layer BP that we study

[a] https://orcid.org/0000-0002-2934-7732
[b] https://orcid.org/0000-0003-4844-1860
[c] https://orcid.org/0000-0002-1662-7583
[d] https://orcid.org/0000-0002-1261-4523
[e] https://orcid.org/0000-0002-3135-2936
[f] https://orcid.org/0000-0002-1280-3447
*https://arxiv.org/pdf/2205.08955.pdf

Szeghy, D., Aslan, M., Fóthi, Á., Mészáros, B., Milacski, Z. and Lőrincz, A.
Structural Extensions of Basis Pursuit: Guarantees on Adversarial Robustness.
DOI: 10.5220/0011138900003277
In *Proceedings of the 3rd International Conference on Deep Learning Theory and Applications (DeLTA 2022)*, pages 77-85
ISBN: 978-989-758-584-5; ISSN: 2184-9277

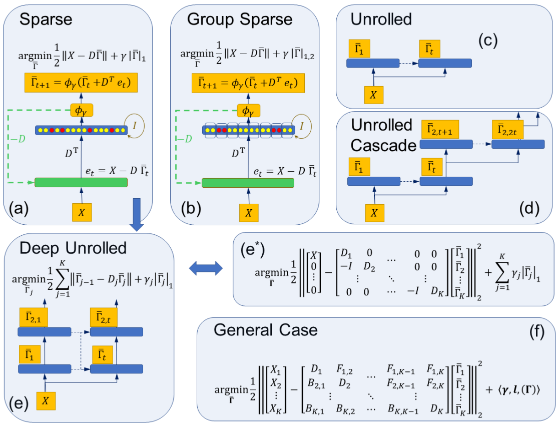

Sparse
$\underset{\bar{\Gamma}}{\mathrm{argmin}} \frac{1}{2}\|X - D\bar{\Gamma}\| + \gamma\,|\bar{\Gamma}|_1$
$\bar{\Gamma}_{t+1} = \phi_\gamma(\bar{\Gamma}_t + D^T e_t)$
$\phi_\gamma$
$I$
$-D$
$D^T$
$e_t = X - D\,\bar{\Gamma}_t$
(a)
$X$
Group Sparse
$\underset{\bar{\Gamma}}{\mathrm{argmin}} \frac{1}{2}\|X - D\bar{\Gamma}\| + \gamma\,|\bar{\Gamma}|_{1,2}$
$\bar{\Gamma}_{t+1} = \phi_\gamma(\bar{\Gamma}_t + D^T e_t)$
$\phi_\gamma$
$I$
$-D$
$D^T$
$e_t = X - D\,\bar{\Gamma}_t$
(b)
$X$
Unrolled
(c)
$\bar{\Gamma}_1$
$\bar{\Gamma}_t$
$X$
Unrolled Cascade
$\bar{\Gamma}_{2,t+1}$
$\bar{\Gamma}_{2,2t}$
$\bar{\Gamma}_1$
$\bar{\Gamma}_t$
$X$
(d)
Deep Unrolled
$\underset{\bar{\Gamma}_j}{\mathrm{argmin}} \frac{1}{2}\sum_{j=1}^{K}\|\bar{\Gamma}_{j-1} - D_j\bar{\Gamma}_j\| + \gamma_j|\bar{\Gamma}_j|_1$
$\bar{\Gamma}_{2,1}$
$\bar{\Gamma}_{2,t}$
$\bar{\Gamma}_1$
$\bar{\Gamma}_t$
(e)
$X$
(e*)
$\underset{\bar{\Gamma}}{\mathrm{argmin}} \frac{1}{2}\left\|\begin{bmatrix}X\\0\\\vdots\\0\end{bmatrix} - \begin{bmatrix}D_1 & 0 & \cdots & 0 & 0\\-I & D_2 & & 0 & 0\\\vdots & & \ddots & & \vdots\\0 & 0 & \cdots & -I & D_K\end{bmatrix}\begin{bmatrix}\bar{\Gamma}_1\\\bar{\Gamma}_2\\\vdots\\\bar{\Gamma}_K\end{bmatrix}\right\|_2^2 + \sum_{j=1}^{K}\gamma_j|\bar{\Gamma}_j|_1$
General Case
(f)
$\underset{\bar{\Gamma}}{\mathrm{argmin}} \frac{1}{2}\left\|\begin{bmatrix}X_1\\X_2\\\vdots\\X_K\end{bmatrix} - \begin{bmatrix}D_1 & F_{1,2} & \cdots & F_{1,K-1} & F_{1,K}\\B_{2,1} & D_2 & & F_{2,K-1} & F_{2,K}\\\vdots & & \ddots & & \vdots\\B_{K,1} & B_{K,2} & \cdots & B_{K,K-1} & D_K\end{bmatrix}\begin{bmatrix}\bar{\Gamma}_1\\\bar{\Gamma}_2\\\vdots\\\bar{\Gamma}_K\end{bmatrix}\right\|_2^2 + \langle\boldsymbol{\gamma}, \boldsymbol{l}, (\boldsymbol{\Gamma})\rangle$

Figure 1: Steps of Basis Pursuit (BP) generalizations. Equations with argmin: the minimization tasks. **(a):** Recurrent BP with sparse representation. Blue (light green) rectangle: representation (input) layer. Blue (dashed light green) arrows: channels that deliver quantities in the actual (in the previous) time step. Red (light yellow) circles: active (non-active) units of the sparse representation. $X$: input. $\bar{\Gamma}_t$ and $e_t$: representation and error at the $t^{th}$ iteration. $\bar{\Gamma}_{j,t}$: same at the $j^{th}$ layer of the deep unrolled network. Matrices $D$, $D_j$: dictionaries, $I$: identity matrix, $\phi_\gamma$ softmax with $\gamma$ bias. **(b):** group sparse case: $\ell_1$ norm is replaced with the $\ell_{1,2}$ norm. **(c):** Unrolled feedforward network with finite number of iterations. **(d):** Cascaded unrolled deep network. **(e):** Non-cascaded modification of the unrolled deep *sparse* cascade. **(e*):** The minimization task of **(e)**. **(f):** The general case still having warranties against adversarial attacks. Within layer groups are not shown. More details: text and supplementary material.

here.

A long-standing problem is that sparse coding is slow. An early effort utilized an associative correlation matrix (Gregor and LeCun, 2010). Recent efforts, put forth the first approximation of BP combined with specific loss terms during training (see (Murdock and Lucey, 2021) and the references therein). Although the approach is attractive, theoretical stability warranties are missing.

We propose group sparse coding as an additional means for the resolution. Sparse coding that exploits $\ell_1$ norm regularization to optimize the hidden representation can be generalized to group sparse coding that uses the $\ell_{1,2}$ norm or the elastic $\ell_{\beta,1,2}$ norm instead.

We present theoretical results on the stability of a family of group sparse coding that alike to its sparse variant can robustly recover the underlying representations under adversarial attacks. Yet, group sparse coding offers fast and efficient feedforward estimations of the groups either by traditional networks or by transformers that the classification step can follow. Previous work (Lőrincz et al., 2016) suggested the feedforward estimation of the groups to be followed by the pseudoinverse estimation of the group activities for learning and finding a group sparse code but without targeting classification or adversarial considerations.

Our feedforward method estimates the $\ell_2$ norms of the active groups followed by the classification step, achieving further computational gains by eliminating the pseudoinverse computations. We consider how to combine the fast estimation with the robust BP computations based on our theoretical and numerical results. However, the speed considerations and test will be presented in a separate paper, now will focus on the robustness results.

Our contributions are as follows:

- we extend the theory of adversarial robustness of Basis Pursuit to a family of networks, including groups, layers, and skip connections between the layers both to deeper and to more superficial layers,
- we introduce group norm based classification and its group pooled variant,
- suggest and study gap regularization,
- execute numerical computations and test feedforward shallow, deep, transformer networks trained on sparse and group sparse layers with a synthetic and the MNIST dataset,study the performance of these fast algorithms, and
- we point to bottlenecks in the training procedures.

We present our theoretical results in Sect. 2. It is followed by the experimental studies (Sect. 3). We examine the properties of the group sparse structures outside of the scope of the theory to foster further works. Section 4 contains the discussions of our results. We conclude in Section 5. Details of the theoretical derivations are in the supplementary material of Footnote*.

## 2 THEORY

We start with the background of the theory including the notations. It is followed by our theoretical results.

### 2.1 Background and Notation

We denote the Sparse Coding (SC) problem by $\boldsymbol{X} = \boldsymbol{D\Gamma}$, where given the *signal* $\boldsymbol{X} \in \mathbb{R}^N$ and the unit-normed *dictionary* $\boldsymbol{D} \in \mathbb{R}^{N \times M}$, the task is to recover the *sparse vector representation* $\boldsymbol{\Gamma} \in \mathbb{R}^M$.

$$\min \|\boldsymbol{\Gamma}\|_0 \text{ subject to } \boldsymbol{X} = \boldsymbol{D\Gamma}, \qquad (P_0)$$

where $\|.\|_0$ denotes the $\ell_0$ norm. For an excellent book on the topic, see (Elad, 2010) and the references therein.

One may try to approximate the solution of Eq. ($P_0$) via the unconstrained version of the Basis Pursuit (BP, or LASSO) method (Tibshirani, 1996; Chen et al., 2001; Donoho and Elad, 2003):

$$\underset{\bar{\mathbf{\Gamma}}}{\operatorname{argmin}} L\left(\bar{\mathbf{\Gamma}}\right) \overset{def}{=} \underset{\bar{\mathbf{\Gamma}}}{\operatorname{argmin}} \frac{1}{2}\left\|\boldsymbol{X}-\boldsymbol{D}\bar{\mathbf{\Gamma}}\right\|_2^2+\gamma\cdot\left\|\bar{\mathbf{\Gamma}}\right\|_1, \tag{BP}$$

where $\gamma > 0$.

Given $\boldsymbol{X} = \boldsymbol{D}\mathbf{\Gamma}$, we may assume that $\mathbf{\Gamma}$ can be further decomposed in a way similar to $\boldsymbol{X}$:

$$\begin{aligned} \boldsymbol{X} &= \boldsymbol{D}_1\mathbf{\Gamma}_1, \\ \mathbf{\Gamma}_1 &= \boldsymbol{D}_2\mathbf{\Gamma}_2, \\ &\vdots \\ \mathbf{\Gamma}_{K-1} &= \boldsymbol{D}_K\mathbf{\Gamma}_K. \end{aligned} \tag{1}$$

The layered problem then tries to recover $\mathbf{\Gamma}_1,\ldots,\mathbf{\Gamma}_K$.

**Definition 1.** *The Layered Basis Pursuit (LBP) (Papyan et al., 2017a) first solves the Sparse Coding problem $\boldsymbol{X} = \boldsymbol{D}_1\mathbf{\Gamma}_1$ via Eq.* (BP) *with parameter $\gamma_1$, obtaining $\hat{\mathbf{\Gamma}}_1$. Next, it solves another Sparse Coding problem $\hat{\mathbf{\Gamma}}_1 = \boldsymbol{D}_2\mathbf{\Gamma}_2$ again by Eq.* (BP) *with parameter $\gamma_2$, denoting the result by $\hat{\mathbf{\Gamma}}_2$, and so on. The final vector $\hat{\mathbf{\Gamma}}_K$ is the solution of LBP. The vector $\boldsymbol{\gamma}^{LBP}$ contains the weights $\gamma_i$ in Eq.* (BP) *for each layer i.*

It was shown in (Papyan et al., 2016) and (Papyan et al., 2017b) that there is strong relationship between the LBP and the CNN, showing that the forward pass of the CNN is in fact identical to a layered pursuit thresholding algorithm, moreover the layered version can improve the system. There was also shown that LBP suffers from error accumulation. To alleviate this obstacle, (Cazenavette et al., 2021) rewrote LBP into a single joint Eq. (BP)-like minimization scheme (i.e., all layers are processed simultaneously) that can be equipped with skip connections. However, the solutions of the two programs differ, and the stability has not been proven for the latter that we do in the supplementary material of Footnote*, see Figs. 1(e*), and (f).

We want to extend these methods to allow different norms on different parts of Γ with different γ weights (as in the layered case) and prove a stability result for this more general case. This will also allow to relieve the condition on the dictionary $\boldsymbol{D}$ that its columns have unit length in the $\ell_2$ norm.

Let us introduce a slightly modified version of the notation used by (Papyan et al., 2016) and (Papyan et al., 2017b). Let Λ be a subset of $\{1,\ldots,M\}$ which is called a *subdomain*, and the components, or *atoms* corresponding to Λ form the *subdictionary* $\boldsymbol{D}_\Lambda$. Let $\boldsymbol{d}_\omega$, $\omega \in \{1,\ldots,M\}$ denote the atom corresponding to the index ω.

If $\Lambda_i(\boldsymbol{D}) \overset{def}{=} \{\omega \mid \langle \boldsymbol{d}_\omega, \boldsymbol{d}_i\rangle \neq 0\}$ and $|\Lambda_i(\boldsymbol{D})|$ is its cardinality, then the restriction $\mathbf{\Gamma}_{\Lambda_i(\boldsymbol{D})} \in \mathbb{R}^{|\Lambda_i(\boldsymbol{D})|}$ of $\mathbf{\Gamma} \in \mathbb{R}^M$ to the indices in $\Lambda_i(\boldsymbol{D})$ is given by,

$$\left(\mathbf{\Gamma}_{\Lambda_i(\boldsymbol{D})}\right)_\theta \overset{def}{=} \begin{cases} \mathbf{\Gamma}_\theta, & \text{if } \theta \in \Lambda_i(\boldsymbol{D}), \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Now let

$$\|\mathbf{\Gamma}\|_{0,st,\boldsymbol{D}} \overset{def}{=} \max_i \left\|\mathbf{\Gamma}_{\Lambda_i(\boldsymbol{D})}\right\|_0 \tag{3}$$

be the *stripe norm with respect to $\boldsymbol{D}$*, a generalization of the definition in (Papyan et al., 2017b).

If $\boldsymbol{D}$ is fixed, then we will use the shorter form $\|\mathbf{\Gamma}\|_{0,st} = \|\mathbf{\Gamma}\|_{0,st,\boldsymbol{D}}$. Further, let $\mu(\boldsymbol{D}) = \max_{i\neq j}\left\langle \boldsymbol{d}_i, \boldsymbol{d}_j\right\rangle$ be the *mutual coherence* of the dictionary (since $\boldsymbol{D}$ is unit-normed the division by $\|\boldsymbol{d}_i\|_2 \cdot \left\|\boldsymbol{d}_j\right\|_2$ is dropped).

We want to use 4 different norms the $\ell_1$, $\ell_2$ and the elastic $\ell_{\beta,1,2}$ norm defined as $\|\boldsymbol{Z}\|_{\beta,1,2} \overset{def}{=} \beta\cdot\|\boldsymbol{Z}\|_1 + (1-\beta)\|\boldsymbol{Z}\|_2$, i.e., it is the convex combination of the $\ell_1$ and $\ell_2$ norms, and finally, the $\ell_{1,2}$ group-norm, sometimes referred to as the Group LASSO (Yuan and Lin, 2006; Bach et al., 2011). To define this we need a group partition of the index set.

If the index set $\{1,\ldots,M\}$ is partitioned into groups $\mathcal{G}_i$, $i \in \{1,\ldots,k\}$ (i.e., $\bigcup_{i=1}^k \mathcal{G}_i = \{1,\ldots,M\}$ and $\mathcal{G}_i \cap \mathcal{G}_j = \emptyset$ for $i \neq j$), then the $\ell_{1,2}$ norm ( see, e.g., (Bach et al., 2011) and the references therein) is

$$\|\boldsymbol{Z}\|_{1,2} \overset{def}{=} \sum_{i=1}^k \left\|\boldsymbol{Z}_{\mathcal{G}_i}\right\|_2, \tag{4}$$

where $\boldsymbol{Z}_{\mathcal{G}_i} = \sum_{j\in\mathcal{G}_i} z_j \cdot \mathbf{e}_j$ with the standard basis vectors $\mathbf{e}_j \in \mathbb{R}^M$, i.e. $z_j$-s are the coordinates of $\boldsymbol{Z}$.

To extend the regularizer of Eq. (BP), if $\mathcal{G}_i$, $i \in \{1,\ldots,k\}$ is a partition of the index set $\{1,\ldots,M\}$ then let

$$\boldsymbol{l}: \mathbb{R}^M \to \mathbb{R}^k, \boldsymbol{l}(\mathbf{\Gamma}) \overset{def}{=} \left(l_{\alpha_1}\left(\mathbf{\Gamma}_{\mathcal{G}_1}\right),\ldots,l_{\alpha_k}\left(\mathbf{\Gamma}_{\mathcal{G}_k}\right)\right), \tag{5}$$

where $l_{\alpha_i}$ is one of the $\ell_1$, $\ell_2$, $\ell_{\beta,1,2}$ norm. For different groups the parameter β can be different as well. So this is a vector which elements are norms evaluated on different parts of $\mathbf{\Gamma}$ corresponding to the different groups and for each group, we can individually decide which norm to use. Let $\boldsymbol{\gamma} \overset{def}{=} (\gamma_1,\ldots,\gamma_k)$ be a weight vector for the different groups (more precisely for the norms of the different groups), where $\gamma_i > 0$, $\forall i$. We want use the regulariser

$$\langle \boldsymbol{\gamma}, \boldsymbol{l}(\mathbf{\Gamma})\rangle = \sum_{i=1}^k \gamma_i \boldsymbol{l}_{\alpha_i}\left(\mathbf{\Gamma}_{\mathcal{G}_i}\right). \tag{6}$$

Note that if for some groups we use the $\ell_2$ norm with the same weight $\gamma$, then we think of this as using the $\ell_{1,2}$ group norm for this group of groups with the weight $\gamma$ being a special case.

Now if we fix a partition $\mathcal{G}_i$ and a regularizer $\boldsymbol{l}$ (i.e. norms for the groups), then let $\boldsymbol{\chi}_{\boldsymbol{\Gamma},\mathcal{G}} \in \mathbb{R}^M$ be the 2-*norm group characteristic vector* of $\boldsymbol{\Gamma}$, i.e.,

$$\left(\boldsymbol{\chi}_{\boldsymbol{\Gamma},\mathcal{G}}\right)_j \overset{def}{=} \begin{cases} 1, & \text{if } j \in \operatorname{supp}\boldsymbol{\Gamma}, \text{ or} \\ & j \in \mathcal{G}_i, \mathcal{G}_i \cap \operatorname{supp}\boldsymbol{\Gamma} \neq \emptyset \text{ and } l_{\alpha_i} = \ell_2, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where $\operatorname{supp}\boldsymbol{\Gamma} \overset{def}{=} \{\omega \mid \boldsymbol{\Gamma}_\omega \neq 0\}$ is the support of $\boldsymbol{\Gamma}$.

For $\boldsymbol{Z} \in \mathbb{R}^N$, we define

$$\left(\boldsymbol{Z}_{\operatorname{supp}\boldsymbol{d}_i}\right)_\theta \overset{def}{=} \begin{cases} z_\theta, & \text{if } \theta \in \operatorname{supp}\boldsymbol{d}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

We call

$$\|\boldsymbol{Z}\|_{L,\boldsymbol{D}} \overset{def}{=} \max_i \left\|\boldsymbol{Z}_{\operatorname{supp}\boldsymbol{d}_i}\right\|_2 \tag{9}$$

the *local amplitude of* $\boldsymbol{Z}$ *with respect to the dictionary* $\boldsymbol{D}$.

For a fixed $D$, we use the shorthand $\|\boldsymbol{Z}\|_L = \|\boldsymbol{Z}\|_{L,\boldsymbol{D}}$.

Both the stripe norm defined previously, and the local amplitude seem difficult to calculate. However, as in (Papyan et al., 2017b) if $\boldsymbol{D}$ corresponds to a CNN architecture, then both become quite natural and the calculation is easy. Moreover, it is easier to keep mutual coherence of the dictionary low.

## 2.2 Theoretical Results

The proofs of the results can be found in the supplementary material of Footnote*.

Here, we will investigate the stability of Eq. (BP) and two closely related algorithms. To unify the several different cases, we introduce the following definition.

**Definition 2.** *First, fix a partition* $\mathcal{G}_i$, $i \in \{1,\ldots,k\}$, *norms for this partition* $\boldsymbol{l}(\boldsymbol{\Gamma})$ *and the weights* $\boldsymbol{\gamma}$ *for the norms. The unconstrained Group Basis Pursuit (GBP) is the solution of the problem:*

$$\underset{\bar{\boldsymbol{\Gamma}}}{\operatorname{argmin}} L\left(\bar{\boldsymbol{\Gamma}}\right) \overset{def}{=} \underset{\bar{\boldsymbol{\Gamma}}}{\operatorname{argmin}} \frac{1}{2}\left\|\boldsymbol{X} - \boldsymbol{D}\bar{\boldsymbol{\Gamma}}\right\|_2^2 + \left\langle \boldsymbol{\gamma}, \boldsymbol{l}\left(\bar{\boldsymbol{\Gamma}}\right)\right\rangle, \tag{GBP}$$

**Theorem 3.** *Let* $\boldsymbol{X} = \boldsymbol{D}\boldsymbol{\Gamma}$ *be a clean signal and* $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}$ *be its perturbed variant. Let* $\boldsymbol{\Gamma}_{GBP}$ *be the minimizer of Eq. GBP where* $\boldsymbol{\gamma}$ *is the weight vector. If among the norms of* $\boldsymbol{l}$ *we used the elastic norm, let* $\{\beta_1,\ldots,\beta_r\}$ *be the set of the parameters used in the elastic norms and* $\lambda \overset{def}{=} \min\{1,\beta_1,\ldots,\beta_r\}$. *Moreover, let* $\gamma_{\max} \overset{def}{=} \max\{\gamma_1,\ldots,\gamma_k\}$ *and* $\gamma_{\min} \overset{def}{=} \min\{\gamma_1,\ldots,\gamma_k\}$ *for the weight vector* $\boldsymbol{\gamma}$ *and* $\theta \overset{def}{=} \frac{\lambda\gamma_{\min}}{\gamma_{\max}}$. *Assume that*

*a)* $\left\|\boldsymbol{\chi}_{\boldsymbol{\Gamma},\mathcal{G}}\right\|_{0,st} \leq c\frac{\theta}{1+\theta}\left(1 + \frac{1}{\mu(\boldsymbol{D})}\right)$,

*b)* $\frac{1}{\lambda(1-c)}\|\boldsymbol{E}\|_L \leq \gamma_{\min}$,

*where* $0 < c < 1$. *If* $\boldsymbol{D}_{\operatorname{supp}\boldsymbol{\chi}_{\boldsymbol{\Gamma},\mathcal{G}}}$ *has full column rank, then*

*1)* $\operatorname{supp}\boldsymbol{\Gamma}_{GBP} \subseteq \operatorname{supp}\boldsymbol{\chi}_{\boldsymbol{\Gamma},\mathcal{G}}$,

*2) the minimizer of Eq. GBP is unique.*

*If we set* $\gamma_{\min} = \frac{1}{\lambda(1-c)}\|\boldsymbol{E}\|_L$, *then*

*3)* $\|\boldsymbol{\Gamma}_{GBP} - \boldsymbol{\Gamma}\|_\infty < \frac{1+\theta}{(1+\mu(\boldsymbol{D}))\theta(1-c)}\|\boldsymbol{E}\|_L$,

*4)* $\left\{i \,\middle|\, |\boldsymbol{\Gamma}_i| > \frac{1+\theta}{(1+\mu(\boldsymbol{D}))\theta(1-c)}\|\boldsymbol{E}\|_L\right\} \subseteq \operatorname{supp}\boldsymbol{\Gamma}_{GBP}$,

*where* $\frac{1+\theta}{(1+\mu(\boldsymbol{D}))\theta(1-c)}\|\boldsymbol{E}\|_L \leq \frac{1+\theta}{\theta(1-c)}\|\boldsymbol{E}\|_L$ *yields a weaker bound in 3) and 4) without the mutual coherence.*

Roughly speaking, if the perturbation is not too large, the support of the noisy representation stays within its clean equivalent, and the indices that are above the threshold level in 4) are recovered. Moreover, we can compare our result to the original Eq. BP, Theorem 6 in (Papyan et al., 2016), as in the pure $\ell_1$ norm case $\lambda = 1$ and if we set $c = \frac{2}{3}$, we get the same bound $\|\boldsymbol{\Gamma}\|_{0,st} < \frac{1}{3}\left(1 + \frac{1}{\mu(\boldsymbol{D})}\right)$, but we have $3\|\boldsymbol{E}\|_L \leq \gamma$ instead of the original $4\|\boldsymbol{E}\|_L$ in b). Similarly, our weaker bound in 3) and 4) is $6\|\boldsymbol{E}\|_L$ instead of their $7.5\|\boldsymbol{E}\|_L$.

Interestingly, this single sparse layer theorem for Eq. GBP extends to multiple layers, where on each layer we can add group partitioning, can choose norms and weights. The precise convergence theorem can be found in the supplementary material of Footnote*. It is a generalized version of Theorem 12 in (Papyan et al., 2017a), but that suffers from error accumulation (Romano et al., 2020).

As mentioned earlier, we can rewrite a layered GBP into a single sparse layer GBP. The solution will differ a bit, but the error accumulation is not present, see the supplementary material of Footnote* for the details. However, the new dictionary describing all the layers won't have unit normalization being a problem in the 'classical' case but not in ours. This is because if the dictionary $\boldsymbol{D}$ is not unit-normed, but the columns belonging to a group $\mathcal{G}_i$ (where we choose the $\ell_2$ or the $\ell_{\beta,1,2}$ norm) have the same $\ell_2$ norm, then we can push the "normalization weights" of the columns of $\boldsymbol{D}$ to the weight $\gamma_i$ in $\boldsymbol{\gamma}$ through the solutions of the

(GBP). The problem and the solution change, but the solution will be equivalent to the original problem, see the supplementary material of Footnote* for further details. This allows us to extend our result for more general sparse coding problems, see Fig. 1f and the supplementary material of Footnote*.

Now, if we stack a linear classifier onto the top of GBP (or onto a layered GBP) as it was done in (Romano et al., 2020), we have several classification stability results, see in the supplementary material of Footnote*.

Also if we solve Eq. (GBP) with positive coding, i.e. restrict the problem to non-negative $\bar{\boldsymbol{\Gamma}}$ vectors, and the solution $\boldsymbol{\Gamma}_{+GBP}$ is group-full (i.e. $\operatorname{supp}\boldsymbol{\Gamma}_{+GBP} = \operatorname{supp}\boldsymbol{\chi}_{\boldsymbol{\Gamma}_{+GBP},\mathcal{G}}$ ) then a weak stability theorem holds for $\boldsymbol{\Gamma}_{+GBP}$, more in the supplementary material of Footnote*.

# 3 EXPERIMENTAL STUDIES

We turn to the description of our numerical studies. We want to explore the limitations of Group Basis Pursuit (GBP) methods and our experiments are outside of the scope of the present theory. We first review the methods. It is followed by the description of the datasets and the experimental results. Throughout these studies we used fully connected (dense) networks implemented in PyTorch (Paszke et al., 2019).

## 3.1 Methods

### 3.1.1 Architectures

To evaluate the empirical robustness of our GBP with $\ell_2$ norm regularization, we compared two variants of it with Basis Pursuit (BP) and 3 Feedforward networks.

For our BP experiments, we used a single BP layer to compute the hidden representation $\boldsymbol{\Gamma}_{BP}$, then stacked a classifier $\boldsymbol{w}$ on top.

Next, for GBP, we considered two scenarios. First, we applied GBP on its own to compute a full $\boldsymbol{\Gamma}_{GBP}$ code. Second, we introduced *Pooled GBP (PGBP)*: after computing $\boldsymbol{\Gamma}_{GBP}$ with GBP, we compressed it with a per group $\ell_2$ norm calculation into $\boldsymbol{\Gamma}_{PGBP}$, and used this smaller code as input to a smaller classifier $\boldsymbol{w}_{PGBP}$.

Finally, we employed 3 feedforward neural networks trained for approximating $\boldsymbol{\Gamma}_{PGBP}$: a Linear Transformer (Katharopoulos et al., 2020), a single dense layer, and a dense deep network having parameter count similar to the Transformer. Network structure details can be found in the supplementary material of Footnote*. For the nonnegative norm values, we used Rectified Linear Unit (ReLU) activation at the top of these networks. To migitate vanishing gradients, we also added a batch normalization layer in some cases. After obtaining the approximate pooled $\hat{\boldsymbol{\Gamma}}_{PGBP}$, we applied the smaller $\boldsymbol{w}_{PGBP}$ as the classifier.

### 3.1.2 Loss Functions

Whenever training was necessary for classification (see Sect. 3.2.2), we pretrained our methods to minimize the unsupervised reconstruction loss $\|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{\Gamma}_{(G)BP}\|_2^2$.

During classification and attack phase, we used a total loss function $J\left(\boldsymbol{D}, \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{X}, \text{class}\left(\boldsymbol{X}\right)\right)$ consisting of a common classification loss term with an optional regularization term.

For the classification loss, we made our choice depending on the number of classes. For the 2 class (binary classification) case we used hinge loss, whereas for the multiclass case we applied the categorical cross-entropy loss.

The regularization loss was specifically employed to test whether it can further improve the adversarial robustness. For this, we introduced a *gap regularization* term to encourage a better separation between active and inactive groups. We intended to increase the smallest difference of preactivations between the smallest active and the largest inactive group norm within a mini-batch of $\boldsymbol{\Gamma}_{(G)BP}$ samples:

$$J_{\text{gap}} = -\min_{i=1,\ldots,N}\Bigg(\min_{j:\,\phi_\gamma\left(||\boldsymbol{\Gamma}^{(i)}_{(G)BP,G_j}||_2\right)\neq 0} ||\boldsymbol{\Gamma}^{(i)}_{(G)BP,G_j}||_2 - \max_{j:\,\phi_\gamma\left(||\boldsymbol{\Gamma}^{(i)}_{(G)BP,G_j}||_2\right)=0} ||\boldsymbol{\Gamma}^{(i)}_{(G)BP,G_j}||_2\Bigg), \tag{10}$$

where $i$ is the sample index, $||\boldsymbol{\Gamma}^{(i)}_{(G)BP,G_j}||_2$ is the $\ell_2$ norm of group $j$ within $\boldsymbol{\Gamma}^{(i)}_{(G)BP}$ (i.e., an element of $\boldsymbol{\Gamma}^{(i)}_{PGBP}$) and $\phi_\gamma$ is an appropriate proximal operator. For the BP case we applied group size 1.

For the training of the feedfoward networks, we applied mean squared error against $\boldsymbol{\Gamma}_{PGBP}$.

### 3.1.3 Adversarial Attacks

To generate the perturbed input $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}$, we used the Iterative Fast Gradient Sign Method (IFGSM) (Kurakin et al., 2016). Specifically, this starts from $\boldsymbol{X}$ and takes $T$ bounded steps wrt. $\ell_\infty$ and $\ell_2$ norms according to the sign of gradient of the total loss $J$ to get $\boldsymbol{Y} = \boldsymbol{Y}_T$:

$$\begin{aligned}
\boldsymbol{Y}_0 &= \boldsymbol{X}, \\
\boldsymbol{G}_{t-1} &= \nabla_{\boldsymbol{Y}_{t-1}} J\left(\boldsymbol{D}, \boldsymbol{w}, \boldsymbol{b}, \boldsymbol{Y}_{t-1}, \text{class}\left(\boldsymbol{X}\right)\right) \\
\boldsymbol{Y}_t &= \text{clamp}\left(\boldsymbol{Y}_{t-1} + a \cdot \text{sgn}\left(\boldsymbol{G}_{t-1}\right)\right).
\end{aligned} \tag{11}$$

where for the learning rate we set $a = \frac{\varepsilon}{T}$ and clamp is a clipping function. Throughout our experiments, we used $T = 20$; for our values of ε, see Sect. 3.3. For most cases, the attack was white box and if applicable, the total loss $J$ included the optional gap regularization term. However, for the 3 Feedforward networks we computed $\boldsymbol{Y}$ using PGBP, resulting in a black box attack.

## 3.2 Datasets

We used three datasets; two synthetic ones and MNIST.

### 3.2.1 Synthetic Data

We generated two synthetic datasets, one without and another with group pooling, according to the following procedure. First, we built a *dictionary* $\boldsymbol{D} \in \mathbb{R}^{100\times 300}$ using normalized Grassmannian packing with 75 groups of size 4 (Dhillon et al., 2008). We generated two normalized random classifiers $\boldsymbol{w} \in \mathbb{R}^{300}$ and $\boldsymbol{w}_{PGBP} \in \mathbb{R}^{75}$ with components drawn from the normal distribution $\mathcal{N}(0,1)$ and set the bias term to zero ($\boldsymbol{b} = 0$). Next, we created the respective input sets. We kept randomly generating $\boldsymbol{\Gamma} \in \mathbb{R}^{300}$ vectors having 8 nonzero groups of size 4 with activations drawn uniformly from $[1,2]$ and computed $\boldsymbol{X} = \boldsymbol{D}\boldsymbol{\Gamma}$. We collected two sets of 10,000 $\boldsymbol{X}$ vectors that satisfied classification margin $O(\boldsymbol{X}) \geq \eta \in \{0.03, 0.1, 0.3\}$ in terms of the classifiers $\boldsymbol{w}$ and $\boldsymbol{w}_{PGBP}$ acting on top of $\boldsymbol{\Gamma}$ (no pooling) and the $\ell_2$ norms of the groups of $\boldsymbol{\Gamma}$ (pooled), respectively. While running our methods, we used a single dense layer and a linear classifier layer with the true parameters ($\boldsymbol{D}$, $\boldsymbol{w}_{(PGBP)}$).

### 3.2.2 MNIST Data

We employed image classification on the real MNIST dataset. The images were vectorized and we preprocessed to zero mean and unit variance. We used a fully connected (dense) dictionary $\boldsymbol{D} \in \mathbb{R}^{784\times 256}$, hidden representation $\boldsymbol{\Gamma}_{(G)BP} \in \mathbb{R}^{256}$ with optionally 32 groups of size 8 for our grouped methods, and a fully connected softmax classifier $\boldsymbol{w}$ mapping to the 10 class probabilities acting either on top of the full $\boldsymbol{\Gamma}_{(G)BP}$ (i.e., $\boldsymbol{w}_i \in \mathbb{R}^{256}$, $i = 1,\ldots,10$) or the compressed $\boldsymbol{\Gamma}_{PGBP}$ (i.e., $\boldsymbol{w}_{PGBP,i} \in \mathbb{R}^{32}$, $i = 1,\ldots,10$). Since in this case the true parameters ($\boldsymbol{D}$, $\boldsymbol{w}$, $\boldsymbol{b}$) were not available for our single layer methods, we tried to learn these via backpropagation over the training set. For this, we applied Stochastic Gradient Descent (SGD) (Bottou et al., 2018) over 500 epochs with early stopping patience 10. To prevent dead units in $\boldsymbol{D}$, we increased γ linearly between 0 and its final value over the initial 4 epochs.

In agreement with the sparse case (Sulam et al., 2020), we found that pretraining the dictionary using reconstruction loss (see Sect. 3.1.2) is beneficial in the group case, too.

## 3.3 Experimental Results

We note that our numerical studies are outside of the scope of the theory as shown in the supplementary material of Footnote* since (i) only about 50% of the perfect group combinations could be found in the synthetic case and (ii) the group assumption is not warranted for the MNIST dataset.

### 3.3.1 Synthetic Experiments

We used three margins, 0.03, 0.1, and 0.3 on the synthetic data. Results for margin 0.1 of the no group pooling and group pooled synthetic experiments are shown in Fig. 2 a) and b), respectively. See the supplementary material of Footnote* for the rest.

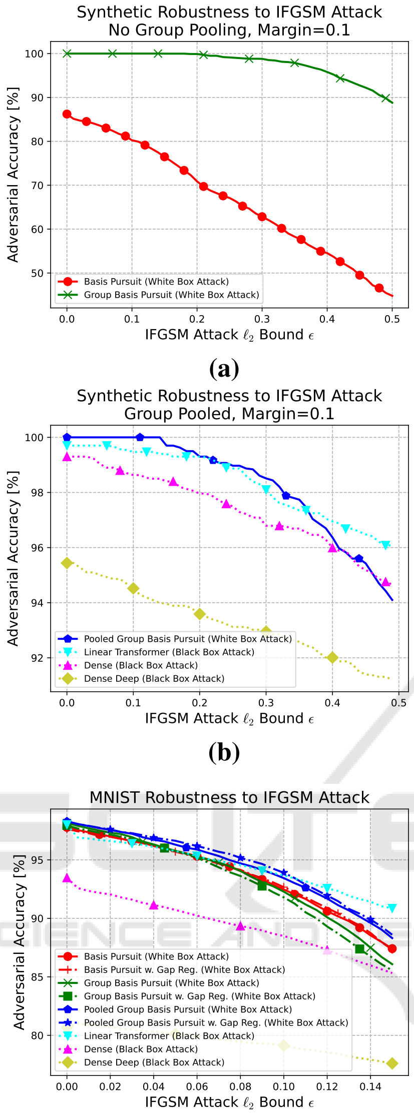
For the no group pooling experiment, we found that BP achieves low accuracy even without attacks, and it breaks down rapidly for increasing ε. In contrast, our GBP achieves perfect scores for low ε, since it has access to the ground truth group structure of the data, and it is able to leverage it. For large ε values, it still breaks down and is slower than BP in the studies domain. Note, however, that the search space is much larger for BP than for GBP.

For the group pooled experiment, the dense, deep dense and transformer networks were trained to approximate PGBP instead of the ground truth, hence they score worse for zero attack. Up to $\varepsilon \approx 0.14$ values, PGBP reaches perfect accuracy. Beyond that and due to the different nature of the attack (white box for PGBP and black box for the others), the breakdown is faster for PGBP than for the other methods. The effect is more pronounced for smaller margins (see the supplementary material of Footnote*). Out of the three feedforward estimations, the transformer performed the best.

### 3.3.2 MNIST Experiment

On MNIST, we compared BP, GBP, PGBP, their respective gap regularized variants and the 3 feedfoward networks. Our results are depicted in Fig. 2 c).

Among the white box attacked pursuit methods, PGBP gave the best results for both the non-attacked and for the attacked case, indicating the benefits of the pooled representation, i.e., it is more difficult to attack group norms than the elements within groups. We think that this result deserves further investigation.



(a)

(b)

(c)

Figure 2: Results for adversarial robustness against Iterative Fast Gradient Sign Method (IFGSM) attack. Datasets differ for all subfigures. Best viewed zoomed in. **(a):** Synthetic dataset, no group pooling: our Group Basis Pursuit (GBP, green) obtains 100% accuracy for small ε and considerably outperforms Basis Pursuit (BP, red) as it can exploit the given group structure. **(b):** Synthetic dataset, group pooling: Pooled Group Basis Pursuit (PGBP, blue) achieves perfect scores for small ε. Break down is faster than for the Linear Transformer (LT, cyan) and the Dense (magenta) networks due to the difference between white box and black box attacks. Deep network (yellow) having parameter count similar to LT is overfitting. **(c)**: MNIST dataset: PGBP is the best for small ε, and it also consistently outperforms all BP and GBP variants for large ε. For some methods, gap regularization (dash-dotted) increases performance. For large ε, black box attacked LT scores the highest. Deep network overfits.

BP and GBP were worse and their curves crossed each other.

Gap regularization (Eq. (3.1.2)) slightly increased performance for BP and PGBP, but it impairs GBP. We believe that this technique may be improved by making it less restrictive, similarly to the modifications for mutual coherence in (Murdock and Lucey, 2020), e.g., by averaging the terms.

Feedforward nets were attacked by the black box method. The Linear Transformer obtained the best results. Deep Network was difficult to teach; it was overfitting.

# 4 DISCUSSION

We have dealt with the structural extensions of basis pursuit methods. We have extended the stability theory of sparse networks and their cascaded versions as follows:

1. The non-cascaded extension (Cazenavette et al., 2021) that includes skip connections beyond the off-diagonal identity blocks of the matrix depicted in Fig. 2 that is the lower triangular part of the matrix can be filled by general blocks has stability proof.
2. Stability proof holds if non-zero general block matrices occur in the upper triangular matrix representing unrolled feedback connections.
3. Stability proof holds if representation elements within any layers are grouped.
4. Different layers and groups can have different biases, diverse norms, such as $\ell_1$, $\ell_{1,2}$, and the elastic norm.
5. The theorem is valid for Convolutional Neural Networks.
6. Proofs are valid for positive coding for the sparse case and under certain conditions, for the group case, too.

Feedforward estimations are fast and our experiments indicate that they are relatively accurate especially for the Linear Transformer for the group structures when there is no attack. In case of attacks, the transformer shows reasonable robustness against black box attacks. However, it seems that transformers are also fragile for white box attacks (Bai et al., 2021). Attacks can be detected as shown by the vast literature on this subject. For recent reviews, see (Akhtar et al., 2021; Salehi et al., 2021) and the references therein. Detection of the attacks can optimize the speed if all (P)GBP and feedforward estimating networks are run in parallel and the detection is fast so it can make the choice in time.

Performances could be improved by introducing additional regularization loss terms (Murdock and Lucey, 2021). We could improve our results by adding a loss term aiming to increase the gap between the groups that will become active and the groups that will be inactive after soft thresholding. Our results are promising and the present loss term (Eq. (3.1.2)) may be too strict. Another interesting loss term could be the minimization of the mutual coherence of $\boldsymbol{D}$ (Murdock and Lucey, 2020) and we leave this examination for future works.

Our experimental studies can be generalized in several ways. Firstly, a single layer can not be perfect for all problems. The hierarchy of layers is most promising for searching for groups of different sizes. As an example, edge detectors can be built hierarchically using CNNs, see, e.g., (Poma et al., 2020).

Further, we restricted the investigations to groups of the same size and the same bias, even though that inputs may be best fit by groups of different sizes, or even by including a subset of single elements, and the bias may also differ. This is an architecture optimization problem, where the solution is unknown. Learning of the sparse representation is however, promising since under rather strict conditions, high-quality sparse dictionaries can be found (Arora et al., 2015). The step to search for groups is still desired since (a) the search space may become smaller by the groups and (b) the presence of the active groups may be estimated quickly and accurately using feedforward methods, especially transformers (in the absence of attacks). In turn, feedforward estimation of the groups followed by (P)GBP with different group sizes including single atoms seems worth studying.

## 5 CONCLUSIONS

We studied the adversarial robustness of sparse coding. We proved theorems for a large variety of structural generalizations, including: groups within layers, diverse connectivities between the layers and versions of optimization costs related to the $\ell_1$ norm. We also studied group sparse networks experimentally. We demonstrated that our GBP can outperform BP, and that our PGBP works better than both using 8 times smaller representation. We found that PGBP offers fast feedforward estimations and the transformer version shows considerable robustness for the datasets we studied. Finally, we showed that gap regularization can improve robustness even further, as suggested by condition *4)* of Theorem 3.

Yet, the scope of our studies are limited from multiple perspectives. First, the suprisingly great performance of our PGBP despite its small representation calls for further investigations using more complex datasets and attacks, as MNIST and IFGSM are too simple and specialized compared to real world scenarios. Second, we believe that theoretical extensions to PGBP are possible, and that varying group sizes and other loss functions may provide performance improvements.

Defenses against noise, novelties, anomalies and, in particular, against adversarial attacks may be solved by combining our robust, structured sparse networks with out-of-distribution detection methods.

## ACKNOWLEDGEMENTS

The research was supported by (a) the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program, (b) Application Domain Specific Highly Reliable IT Solutions project of the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme no. 2020-4.1.1.-TKP2020 (National Challenges Subprogramme) funding scheme and (c) D. Szeghy was partially supported by the NKFIH Grant K128862.

## REFERENCES

Akhtar, N., Mian, A., Kardan, N., and Shah, M. (2021). Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access*, 9:155161–155196.

Arora, S., Ge, R., Ma, T., and Moitra, A. (2015). Simple, efficient, and neural algorithms for sparse coding. In *Conf. on Learn. Theo.*, pages 113–149. PMLR.

Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. (2011). Optimization with sparsity-inducing penalties. *arXiv:1108.0775*.

Bai, Y., Mei, J., Yuille, A. L., and Xie, C. (2021). Are transformers more robust than cnns? *Adv. in Neural Inf. Proc. Syst.*, 34.

Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311.

Cazenavette, G., Murdock, C., and Lucey, S. (2021). Architectural adversarial robustness: The case for deep pursuit. In *IEEE/CVF Conf. on Comp. Vis. and Patt. Recogn.*, pages 7150–7158.

Chen, S. S., Donoho, D. L., and Saunders, M. A. (2001). Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159.

Dhillon, I. S., Heath, J. R., Strohmer, T., and Tropp, J. A. (2008). Constructing packings in Grassmannian manifolds via alternating projection. *Exp. Math.*, 17(1):9–35.

Donoho, D. L. and Elad, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell_1$ minimization. *Proc. Natl. Acad. Sci.*, 100(5):2197–2202.

Elad, M. (2010). *Sparse & Redundant Representations and Their Applications in Signal and Image Processing*. Springer Science & Business Media.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv:1412.6572*.

Gregor, K. and LeCun, Y. (2010). Learning fast approximations of sparse coding. In *27th Int. Conf. on Mach. Learn.*, pages 399–406.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. (2020). Transformers are rnns: Fast autoregressive transformers with linear attention. In *Int. Conf. on Mach. Learn.*, pages 5156–5165. PMLR.

Kurakin, A., Goodfellow, I. J., and Bengio, S. (2016). Adversarial examples in the physical world. *arXiv:1607.02533*.

Liu, Y., Chen, X., Liu, C., and Song, D. (2016). Delving into transferable adversarial examples and black-box attacks. *arXiv:1611.02770*.

Lőrincz, A., Milacski, Z. A., Pintér, B., and Verő, A. L. (2016). Columnar machine: Fast estimation of structured sparse codes. *Biol. Insp. Cogn. Arch.*, 15:19–33.

Murdock, C. and Lucey, S. (2020). Dataless model selection with the deep frame potential. In *IEEE/CVF Conf. on Comp. Vis. and Patt. Recogn.*, pages 11257–11265.

Murdock, C. and Lucey, S. (2021). Reframing neural networks: Deep structure in overcomplete representations. *arXiv:2103.05804*.

Papyan, V., Romano, Y., and Elad, M. (2017a). Convolutional neural networks analyzed via convolutional sparse coding. *J. Mach. Learn. Res.*, 18(1):2887–2938.

Papyan, V., Sulam, J., and Elad, M. (2016). Working locally thinking globally-Part II: Stability and algorithms for convolutional sparse coding. *arXiv:1607.02009*.

Papyan, V., Sulam, J., and Elad, M. (2017b). Working locally thinking globally: Theoretical guarantees for convolutional sparse coding. *IEEE Trans. Signal Process.*, 65(21):5687–5701.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. *arXiv:1912.01703*.

Poma, X. S., Riba, E., and Sappa, A. (2020). Dense extreme inception network: Towards a robust CNN model for edge detection. In *IEEE/CVF Winter Conf. on Apps. of Comp. Vis.*, pages 1923–1932.

Romano, Y., Aberdam, A., Sulam, J., and Elad, M. (2020). Adversarial noise attacks of deep learning architectures: Stability analysis via sparse-modeled signals. *J Math. Imag. and Vis.*, 62(3):313–327.

Salehi, M., Mirzaei, H., Hendrycks, D., Li, Y., Rohban, M. H., and Sabokrou, M. (2021). A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *arXiv:2110.14051*.

Sulam, J., Muthukumar, R., and Arora, R. (2020). Adversarial robustness of supervised sparse coding. In *Adv. in Neural Inf. Proc. Syst.*, volume 33, pages 2110–2121.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the LASSO. *J. R. Stat. Soc. Series B (Methodol.)*, 58(1):267–288.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Series B (Methodol.)*, 68(1):49–67.

# APPENDIX

Due to space constraints, we were only able to state our main result of Theorem 3 here. The rest of our theorems and all proofs can be found in the supplementary material of Footnote*, the url is located right below the abstract.