SIAM J. MATH. DATA SCI.
Vol. 1, No. 3, pp. 446–475

Multilinear Compressive Sensing and an Application to Convolutional Linear Networks*

François Malgouyres† and Joseph Landsberg‡

Abstract. We study a deep linear network endowed with the following structure: A matrix X is obtained by multiplying K matrices (called factors and corresponding to the action of the layers). The action of each layer (i.e., factor) is obtained by applying a fixed linear operator to a vector of parameters satisfying a constraint. The number of layers is not limited. Assuming that X is given and factors have been estimated, the error between the product of the estimated factors and X (i.e., the reconstruction error) is either the statistical or the empirical risk. We provide necessary and sufficient conditions on the network topology under which a stability property holds. The stability property requires that the error on the parameters defining the near-optimal factors scales linearly with the reconstruction error (i.e., the risk). Therefore, under these conditions on the network topology, any successful learning task leads to stably defined features that can be interpreted. In order to do so, we first evaluate how the Segre embedding and its inverse distort distances. Then we show that any deep structured linear network can be cast as a generic multilinear problem that uses the Segre embedding. This is the *tensorial lifting*. Using the tensorial lifting, we provide a necessary and sufficient condition for the identifiability of the factors up to a scale rearrangement. We finally provide a necessary and sufficient condition called the deep-Null Space Property (because of the analogy with the usual Null Space Property in the compressed sensing framework) which guarantees that the stability property holds. We illustrate the theory with a practical example where the deep structured linear network is a convolutional linear network. We obtain a condition on the scattering of the supports which is strong but not empty. A simple test on the network topology can be implemented to test whether the condition holds.

Key words. interpretable learning, stable recovery, matrix factorization, deep linear networks, convolutional networks

AMS subject classifications. 68T05, 90C99, 15-02

DOI. 10.1137/18M119834X

1. Introduction.

1.1. The aim of the paper.

Deep learning has led to many practical breakthroughs and to significant improvements and state-of-the-art performances in many fields such as computer vision, natural language processing, signal processing, robotics, etc. The range of applications grows at a strong pace. Despite these empirical successes, the theory supporting deep learning

*Received by the editors July 5, 2018; accepted for publication (in revised form) May 16, 2019; published electronically August 21, 2019.
https://doi.org/10.1137/18M119834X
Funding: This work was supported by the DEEL program on Dependable and Explainable Learning (https://www.deel.ai). The second author was supported by NSF grants DMS-1405348 and AF1814254.
†Institut de Mathématiques de Toulouse, UMR5219 Université de Toulouse, CNRS UPS IMT, F-31062 Toulouse Cedex 9, France (Francois.Malgouyres@math.univ-toulouse.fr), and Institut de Recherche Technologique Saint Exupery.
‡Department of Mathematics, Texas A&M University, College Station, TX 77843-3368 USA (jml@math.tamu.edu).

is still far from satisfactory. For instance, sharp and accurate answers to the most natural questions on (i) the efficiency of optimization algorithms when applied to the objective function minimized in deep learning [55, 37, 22, 23, 69, 10, 11, 44, 80]; (ii) the expressiveness of the networks [12, 3, 31, 26, 45, 27, 64, 77]; and (iii) guarantees on the statistical risk [42, 34, 81, 70] for learned neural networks are still missing. This makes it difficult to optimize and configure neural networks. Moreover, the absence of answers to these questions prevents certification that systems built with deep learning algorithms are robust.

The reasons explaining the outcome of a neural network are often difficult to highlight [8, 63, 41]. Even worse, despite the settings described in [6, 4, 14, 53, 72, 84], the instability of the parameters optimizing the deep learning objective does not allow the interpretation of the features defined by these parameters. This last problem is the one we investigate in this work.

Our goal in this paper is to evaluate how far the architectures used in applications are from architectures for which we can guarantee that the parameters returned by the algorithm, and therefore the features defined using these parameters, are stably defined. To do so, we consider two families of networks and establish necessary and sufficient conditions on their topology guaranteeing that the features learned by the algorithm are stably defined.

More precisely, we establish statements of the following form for two families of deep networks. Below, the action of the network parameterized by $\mathbf{h}$ is denoted $f_{\mathbf{h}}$.

Informal Theorem 1.1 (stability guarantee). *We assume a known parameterized family of functions $f_{\mathbf{h}}$ and a metric[1] d between parameter pairs. We establish a necessary and sufficient condition on the family $f_{\mathbf{h}}$ guaranteeing the following:*

There exists a constant $C > 0$ such that for any input/output pairs I, X and any pair of parameters $\mathbf{h}^*$, $\overline{\mathbf{h}}$ for which

$$\delta = \|X - f_{\mathbf{h}^*}(I)\|$$

and

$$\eta = \|X - f_{\overline{\mathbf{h}}}(I)\|$$

are sufficiently small, we have

$$d(\overline{h}, h^*) \leq C(\delta + \eta). \tag{1.1}$$

Considering a regression problem, the values δ and η can be interpreted as the statistical or the empirical risk for the parameters $\overline{h}$ and h^*. Inequality (1.1) therefore guarantees that the set made of the parameters leading to a small risk has a small diameter. The features defined using such parameters are therefore stably defined. This seems to be the minimal condition allowing the interpretation of the features. The condition on the family of functions $f_{\mathbf{h}}$ is typically a condition on the topology of the network.

In Informal Theorem 1.1, $\overline{h}$ and h^* might have different roles. For instance, if we know that the input/output pairs have been generated using a particular $\overline{h}$, possibly up to some error as modeled by δ, then (1.1) guarantees that h^* is close to $\overline{h}$ and provides a way to control the statistical risk.

[1]The metric takes into account interlayer rescaling.

The existing stability guarantees [6, 4, 14, 53, 72, 84, 60] consider this setting and describe both a network topology and an algorithm whose output h^* is guaranteed to be close to $\overline{h}$. In this study, we do not make any assumption on the construction of $\overline{h}$ and h^*, and our objective is more modest. With regard to their objective, giving a necessary and sufficient condition of stability plays the same role as a complexity theory statement saying that a particular configuration is NP-hard. It rules out some network topologies.

Notice that when I and X are such that it is possible to have $\delta = \eta = 0$, the above stability guarantee implies that the minimizer of the network objective function is unique. Theorems 6.4 and 6.5 will show that this uniqueness condition is strongly related to the level of overspecification of the network. The simplified and intuitive statement is that optimal solutions of overspecified networks are not unique and are unstable. This explains the instability observed in applications. The theorems analyze this property in detail. This might be viewed as a negative result since overspecification is currently the main hypothesis of statements guaranteeing the success of the neural network optimization.

1.2. The considered deep networks.

1.2.1. Overview. We consider two kinds of deep networks: a general family of deep structured linear networks[2] in sections 6 and 7, and a family of convolutional linear networks in section 8. The formal statements for the deep structured linear networks are in Theorems 7.2 and 7.3. The statements for convolutional linear networks are in Theorem 8.4. Below, we describe the deep structured linear networks.

1.2.2. Deep structured linear networks. The term deep linear network usually corresponds to fully connected feed-forward networks, without bias, in which the activation function is the identity. In the general results described in this paper we consider deep linear networks and provide two means to enforce some structure to the network. As we describe below, the structures can be used to include feed-forward linear networks; convolutional linear networks (as is done in section 8); the action of a ReLU activation function; sparse networks; nonnegative networks; and combinations of the above. The family also includes most matrix factorization problems.

We model a deep structured linear network as a product of matrices called factors. The factors depend linearly on parameters in $\mathbb{R}^S$ for $S \in \mathbb{N}$.

More precisely, consider an arbitrary depth parameter $K \geq 1$. The number of layers is $K+1$, and the layers are enumerated in such a way that the layer receiving the input is $K+1$ and the layer returning the output is 1. We consider sizes $m_1 \dots m_{K+1} \in \mathbb{N}$, writing $m_1 = m$, $m_{K+1} = n$. We consider, for $k = 1, \dots, K$, the linear map

$$\begin{array}{rcl} M_k : \mathbb{R}^S & \longrightarrow & \mathbb{R}^{m_k \times m_{k+1}} \\ h & \longmapsto & M_k(h). \end{array} \tag{1.2}$$

Given some parameters, $h_1, \dots, h_K \in \mathbb{R}^S$, the action of the deep structured linear network is

[2] We call this family *deep structured linear networks* because the family is endowed with tools to impose structures. We analyze the impact of the structure on the stability property. However, these tools might be used to define the usual deep linear network.

the product

$$M_1(h_1)\cdots M_K(h_K).$$

The factor $M_K(h_K)$ might involve the inputs of the samples by considering $M_K(h_K) = M'_K(h_K)I$ for a linear map M'_K and for a matrix I whose columns contain the inputs. Given outputs $X \in \mathbb{R}^{m\times n}$, the optimization of the parameters $h_1,\dots,h_K$ defining the network aims at getting

$$M_1(h_1)\cdots M_K(h_K) \simeq X.$$

To model feed-forward linear networks, the mappings M_k, $k = 1,\dots,K-1$ (and M'_K) construct the matrix by placing the entry of h_k corresponding to an edge in the network in the corresponding entry in $M_k(h_k)$.

For convolutional layers, M_k and M'_K concatenate convolution matrices[3] defined by a portion of the entries in $\mathbf{h}_k$. Each convolution matrix is at the location corresponding to a prescribed edge.

The main argument for studying deep structured linear networks is due to their strong connection to nonlinear networks that uses the rectified linear unit (ReLU)[4] activation function. We explain it in detail. The action of the ReLU activation function at the layer k treats every entry independently of the other entries and multiplies it by either 1 (the entry is kept) or 0 (the entry is canceled). More precisely, denoting $\mathbf{h} = (h_k)_{k=1,\dots,K}$, the action of the ReLU activation function on the layer k is to apply the map $A_k : \mathbb{R}^{m_k\times n} \longmapsto \mathbb{R}^{m_k\times n}$ (where $m_k \times n$ is the size data in the layer k) such that

$$(A_kM)_{i,j} = a_k(\mathbf{h})_{i,j}M_{i,j} \qquad \text{for } (i,j) \in \{1,\dots,m_k\}\times\{1,\dots,n\},$$

where $a_k(\mathbf{h}) \in \{0,1\}^{m_k\times n}$ is defined by

$$a_k(\mathbf{h})_{i,j} = \begin{cases} 1 & \text{if } \Big(M_{k+1}(h_{k+1})A_{k+1}M_{k+2}(h_{k+2})\cdots A_{K-1}M_K(h_K)\Big)_{i,j} \geq 0,\\ 0 & \text{otherwise.}\end{cases}$$

The function

$$\begin{aligned} a_k : \mathbb{R}^{S\times K} &\longrightarrow \{0,1\}^{m_k\times n}\\ \mathbf{h} &\longmapsto a_k(\mathbf{h})\end{aligned}$$

is piecewise constant because $\{0,1\}^{m_k\times n}$ is finite. (This has already been used in [69].) As a consequence, the parameter space $\mathbb{R}^{S\times K}$ is partitioned into subsets such that on every subset a_k is constant for all $k = 1,\dots,K$. Therefore, on every subset the action of the nonlinear network coincides with the action of a deep structured linear network that groups at every layer A_k and M_{k+1}. Further, the landscape of the objective function of the nonlinear neural network that uses ReLU coincides, on every part of the partition, with the landscape of a deep structured linear network. This is a strong argument in favor of the study of deep structured linear networks.

[3]Depending on the situation: Toeplitz, block-Toeplitz, circulant, or block-circulant matrices. The matrices often involve downsampling.

[4]ReLU is the most common activation function.

Notice that deep structured linear networks have also been obtained in [22, 23, 44] by modeling the action of the activation function as random, independent of the input and when considering the expectation of the network action. However, these assumptions are not satisfied by the deep networks used in applications (see [23]), and it is not clear that this link can be exploited to obtain theoretical guarantees for realistic deep networks.

In addition to the structure induced by the operators M_k, we also consider a structure imposed on the vectors h. We assume that we know a collection of models $\mathcal{M} = (\mathcal{M}^L)_{L\in\mathbb{N}}$ with the property that for every L, $\mathcal{M}^L \subset \mathbb{R}^{S\times K}$ is a given subset. We will assume that the parameters $\mathbf{h} \in \mathbb{R}^{S\times K}$ defining the factors are such that there exists $L \in \mathbb{N}$ such that $\mathbf{h} \in \mathcal{M}^L$. For instance, the constraint $\mathbf{h} \in \mathcal{M}^L$ might be used to impose sparsity, grouped sparsity, or cosparsity. One might also use the constraint $\mathbf{h} \in \mathcal{M}^L$ to impose nonnegativity, orthogonality, equality, compactness, etc. Generally speaking $\mathcal{M}$ is used to impose some prior or some form of regularity or to compress the parameter space and obtain better bounds [7]. The models might also be used to alleviate ambiguities. For instance, if the operators M_k and M_{k+1} allow permutations (i.e., there exist $(h_k, h_{k+1}) \neq (g_k, g_{k+1})$ and a permutation matrix C such that $M_k(h_k) = M_k(g_k)C$ and $M_{k+1}(h_{k+1}) = C^{-1}M_{k+1}(g_{k+1})$), we can use a complete ordering of the parameter space $\mathbb{R}^{K\times S}$ and impose, using $\mathcal{M}$, the largest of all the equivalent versions of parameters to be considered.

1.3. Bibliography.

1.3.1. Other matrix factorization and compressed sensing. The content of this paper is strongly related to and can be considered as an extension of the research field usually named *compressed sensing*. Because of the importance of this field of research and to simplify the reading for readers whose main interest is in deep learning, we have separated this part of the bibliography and placed it in section 2. Notice that the statement of Informal Theorem 1.1 can be interpreted in the context of signal recovery. In particular, the results on deep structured linear networks can probably be specialized to be applicable to matrix factorization problems for which stability properties have not been established [78, 20, 21, 59, 58, 46, 56]. We have not investigated this potential.

1.3.2. Tensors and deep networks. The analysis conducted in this paper is based on a connection, named *tensorial lifting*, between deep structured linear networks and a tensor problem (see section 5). The tensorial lifting has already been described in [60], but other connections between tensor and network problems have been described by other authors. In particular, in [26, 27, 45], the authors define a score function using a tensor. They highlight a network topology that computes the score function defined by a tensor decomposable using a CP-decomposition, a Hierarchical Tucker [26, 27], or a tensor train decomposition[45]. They then deduce the expressive power of the network topology from the connections between the tensor decompositions. These results highlight and analyze why deep networks are more expressive than shallow ones. Tensors and tensor decomposition have also been used to represent the cross-moment and construct a solver [42], encode the convolution layers with a tensor of order 4, and manipulate this tensor to improve the network [49, 66, 83] to represent a tensor layer [74, 82].

1.3.3. Stability property for neural networks. To the best of our knowledge, the articles establishing stability properties are [4, 14, 53, 72, 84, 60].

Among them, [14, 53, 84] consider a family of networks of depth 1 or 2 (depending on the article, the definition of the depth may vary). The article [72] contains a study on deep networks (the depth can be large), but the study only focuses on the recovery of one layer. The articles [4, 60] consider networks without depth limitation.

In [14], the authors consider the minimization of the statistical risk (not the empirical risk). The input is assumed Gaussian and the output is generated by a network involving one linear layer followed by ReLU and a mean. The number of intermediate nodes is smaller than the input size. They provide conditions guaranteeing that, with high probability, a randomly initialized gradient descent algorithm converges to the true parameters. The authors of [53] consider a feed-forward network made of one unknown linear layer, followed by ReLU and a sum. The size of the intermediate layer equals the size of the data, the size of the output is 1. Again, they assume Gaussian input data and consider the minimization of the risk (not the empirical risk). They show that the stochastic gradient descent converges to the true solution. In [84], the authors consider a nonlinear layer followed by a linear layer. The size of the intermediate layer is smaller than the size of the input, and the size of the output is 1. They describe an initialization algorithm based on a tensor decomposition such that with high probability, the gradient algorithm minimizing the empirical risk converges to the true parameters that generated the data.

The authors of [72] consider a feed-forward neural network and show that if the input is Gaussian or its distribution is known, a method based on moments and sparse dictionary learning can retrieve the parameters defining the first layer. Nothing is said about the stability or the estimation of the other layers.

The authors of [4] consider deep feed-forward networks which are very sparse and randomly generated. They show that they can be learned with high probability one layer after another. However, very sparse and randomly generated networks are not used in practice, and one might want to study more versatile structures. The article [60] studies deep structured linear networks (without the models $\mathcal{M}$) and uses the same tensorial lifting we use here. However, in [60] the function d measuring the error between parameters is only defined using the ℓ^∞ norm and is not a metric. The transversality condition of [60] is sufficient to guarantee the stability but is not necessary. All these weaknesses are corrected in this extended version. The general result is also specialized to deep convolutional linear networks.

1.4. Organization of the paper. Because it is strongly related, we give an extensive bibliography on compressive sensing and stable recovery properties for matrix factorization problems in section 2. We describe the framework of the paper and our notations in section 3.

The following are the main contributions of this paper:

- In section 4, we investigate and recall several results on tensors, tensor rank, and the Segre embedding. In particular, we investigate how the Segre embedding distorts distances.
- In section 5, we describe the *tensorial lifting*. It expresses any deep structured linear networks in a generic multilinear format. The latter composes a linear lifting operator and the Segre embedding.

- When $\delta = \eta = 0$ (see section 6):
  - We establish a simple geometric condition on the intersection of two sets which is necessary and sufficient to guarantee the identifiability of the parameters up to scale ambiguity (Proposition 6.3).
  - We provide simpler conditions which involve the rank of the lifting operator (defined in section 5) such that we have the following:
    - Underspecified case: If the lifting operator rank is large (e.g., larger than $2K(S-1)+2$, when $\mathcal{M} = \mathbb{R}^{S\times K}$) and the lifting operator is random, for almost every lifting operator, the solution of
    $$M_1(h_1)\cdots M_K(h_K) = X$$
    is identifiable (Theorem 6.4).
    - Overspecified case: If the lifting operator rank is small (e.g., smaller than $2S-1$, when $\mathcal{M} = \mathbb{R}^{S\times K}$), the solution of
    $$M_1(h_1)\cdots M_K(h_K) = X$$
    is not identifiable (Theorem 6.5).
  - We also provide a simple algorithm to compute the rank of the lifting operator (Proposition 5.3).
- Stability guarantee statements for deep structured linear networks are in section 7:
  - We define the deep-Null Space Property (Definition 7.1): a generalization of the usual Null Space Property [25] that also applies to the deep problems.
  - We establish that the deep-Null Space Property is a necessary and sufficient condition to guarantee stability (see the informal statement above or Theorems 7.2 and 7.3).
- We specialize the results to convolutional linear networks in section 8 and establish a simple condition that can be computed (see Algorithm 8.1). The is such that (see Theorem 8.4)
  - if the condition is satisfied, the convolutional linear networks can be stably recovered;
  - if the condition is not satisfied, the convolutional linear network is not identifiable.

  In simple words, the condition holds when the supports of the convolution kernels are sufficiently scattered. This is not satisfied by the convolutional kernel used in applications and explains their instability.

Because of space constraints, all the proofs are provided as a supplementary material or in the public archive [61].

2. Bibliography on matrix factorization and compressed sensing. Before describing the bibliography on compressed sensing, we interpret this stability statement of Informal Theorem 1.1 in the context of signal processing. In signal processing, we usually know that $\overline{h}$ exists and δ represents the sum of a modeling error and noise. Inequality (1.1) guarantees that when the condition is satisfied, even an approximative minimizer of

$$\text{argmin}_{L\in\mathbb{N},(h_k)_{k=1..K}\in\mathcal{M}^L} \ \|M_1(h_1)\cdots M_K(h_K) - X\|^2 \tag{2.1}$$

leads to a solution h^* close $\overline{h}$. This property is often named the *stable recovery guarantee*.

When $\delta = 0$ (i.e., the data exactly fits the model and is not noisy) and $\eta = 0$ (i.e., (2.1) is perfectly solved) this is an *identifiability guarantee*. This is a necessary condition of stable recovery.

In this section, we distinguish the cases $K = 1$, $K = 2$, and $K \geq 3$.

2.1. $K = 1$: Linear inverse problems. The simplest version consists of a model with one layer (i.e., $K = 1$) and $\mathcal{M} = \mathbb{R}^{S \times K}$. Recovering h_1 from X is a linear inverse problem. The data X can be vectorized to form a column vector, and the operator M_1 simply multiplies the column vector h_1 by a fixed (rectangular) matrix. Typically, when the linear inverse problem is overdetermined, the latter matrix has more rows than columns, the uniqueness of a solution to (2.1) depends on the column rank of the matrix, and the stable recovery constant depends on the smallest singular value of M_1.

When the matrix is not full column rank, the identifiability and stable recovery for this problem has been intensively studied for many constraints $\mathcal{M}$. In particular, for sparsity constraints this is the compressed/compressive sensing problem (see the seminal articles [15, 30]). Some compressed sensing statements (especially the ones guaranteeing that any minimizer of the ℓ^0 problem stably recovers the unknown problem) are special cases ($K = 1$) of the statements provided in this paper. We will not give a complete review on compressed sensing but would like to highlight the Null Space Property described in [25]. The fundamental limits of compressed sensing (for a solution of the ℓ^0 problem) have been analyzed in detail in [13].

Although the main novelty of the paper is to investigate stable recovery properties for any $K \geq 1$, we specialize the statements made for $K \geq 1$ to the case $K = 1$ in order to illustrate the new statements and to provide a way of comparison with well-known results.

2.2. $K = 2$: Bilinear inverse problems and bilinear parameterizations. When $k \geq 2$, the problem becomes nonlinear because of the product in (2.1). This significantly complicates the analysis. What follows are the main instances studied in the literature when $K = 2$.

Nonnegative matrix factorization and low rank prior. In nonnegative matrix factorization (NMF) [50], M_1 and M_2 map the entries in h_1 and h_2 at prescribed locations in the factors (say, one column after another). The constraints $\mathcal{M}$ imposes that all the entries in h_1 and h_2 are nonnegative. The NMF has been widely used for many applications.

Conditions guaranteeing that the factors provided by the NMF identify[5] the correct factors (up to rescaling and permutation) were first established in the pioneering work [29]. To the best of our knowledge, this is the first paper addressing recovery guarantees for a problem of depth $K = 2$. It emphasizes a separability condition that guarantees identifiability. The proof is purely geometric and relies on the analysis of inclusions of simplicial cones. This result is significantly extended in [48]. In this paper, the continuity of the NMF estimator is established. Concerning computational aspects, NMF is NP-complete [79]. However, under the separability hypothesis of [29], the solution of the NMF problem can be computed in polynomial time [5].

[5]Stable recovery is not established.

We can slightly generalize[6] the problem and introduce a linear degradation operator

$$H : \mathbb{R}^{m\times n} \longrightarrow \mathbb{R}^{m\times n}.$$

Use the same mapping M_1 and M_2 as for the NMF, with $\mathcal{M} = \mathbb{R}^{S\times K}$, but with a small number of lines (resp., columns) in $M_2(h_2)$ (resp., $M_1(h_1)$). Any solution of the problem

$$(h_1^*, h_2^*) \in \operatorname{argmin}_{(h_1,h_2)\in\mathbb{R}^{S\times K}} \|H(M_1(h_1)M_2(h_2)) - X\|^2$$

leads to a low rank approximation $M_1(h_1^*)M_2(h_2^*)$ of an inverse of H at X. Again, a large corpus of literature exists on the low rank prior [67, 17, 32, 19].

Phase retrieval. Phase retrieval fits the framework described in the present paper when we take

$$M_1(h_1) = \operatorname{diag}(\mathcal{F}h_1), \qquad M_2(h_2) = (\mathcal{F}h_2)^*,$$

and

$$\mathcal{M} = \{(h, h) \in \mathbb{R}^{S\times K} \mid h \in \mathbb{R}^S\},$$

where S is the size of the signal, $\mathcal{F}$ computes N linear measurements of any element in $\mathbb{R}^S$ (typically Fourier measurements), $\operatorname{diag}(.)$ creates an $N \times N$ diagonal matrix whose diagonal contains the input, and * is the (entrywise) complex conjugate.

The tensorial lifting at the core of the present paper generalizes the lifting used in the inspiring work on PhaseLift [52, 18, 16]. As is often the case when $K = 2$, PhaseLift is a semidefinite program that can be efficiently solved when the unknown is of moderate size. These papers also provide conditions on the measurements guaranteeing that the phases are stably recovered by PhaseLift.

The benefit of the generalization introduced with the tensorial lifting is that it applies to any multilinear inverse problem.

Self-calibration and demixing. Measuring operators often depend linearly on parameters that are not perfectly known. The estimation of these parameters is crucial to restoring the data measured by the device. This is the self-calibration problem. This naturally fits the setting of this article: We let h_1 be the parameters defining the sensing matrix and $M_1(h_1)$ be the sensing matrix. Then h_2 defines the signal (or signals) contained in the column(s) of $M_2(h_2)$.

Many instances of this problem have been studied and much progress has been made to obtain algorithms that can be applied to problems of larger and larger size. This leads to a very interesting line of research.

To the best of our knowledge, the first stable recovery statements concern the blind-deconvolution problem. In [2], the authors use a lifting to transform the blind-deconvolution problem into a semidefinite program with an unknown whose size is the product of the sizes[7] of h_1 and h_2. Such problems can be solved for unknowns of moderate size. The authors of [2] provide explicit conditions guaranteeing the stable recovery with high probability. This

[6] The interested readers can check that this generalization only leads to a small change of the lifting operator introduced in section 5. It is therefore done at no cost.

[7] With our notation this is simply $S \times S$, but this can be much more favorable.

idea has been generalized and applied to other similar problems in [24, 9]. The authors of [54] consider a significantly more general calibration model. In this model, $M_1(h_1)$ is diagonal and its diagonal contains the entries of h_1. $M_2(h_2)$ simply multiplies h_2 by a fixed known matrix (the theorems consider a random matrix). The constraint imposes h_2 to be sparse. For this problem, they prove that with high probability the numerical method called SparseLift is stable with a controlled accuracy. SparseLift returns the left and right singular vectors of the solutions of an ℓ^1 optimization problem whose unknown is the same as in [2]. However, solving an ℓ^1 minimization problem is much simpler than a semidefinite problem. This is a very significant practical improvement.

As emphasized in [51], in order to motivate its nonconvex approach, the only drawback of the numerical methods described in [2, 54] is their complexity. The extra complexity is due to the fact that they optimize a variable in the product space $\mathbb{R}^{S\times S}$ and then deduce an approximate solution of the unlifted problem. This is what motivates the authors of [51] to propose a nonconvex approach. The constructed algorithm provably stably recovers the sensing parameters and the signals with a geometric convergence rate.

Sparse coding and dictionary learning. Sparse coding and dictionary learning is another kind of bilinear problem (see [68] for an overview). In that framework, the columns of X contain the data. Most often, people consider two layers: $K = 2$. The layer $M_1(h_1)$ is an optimized dictionary of atoms defined by the parameters h_1, and each column of $M_2(h_2)$ contains the code (or coordinates) of the corresponding column in X. Most often, h_2 is assumed sparse.

The identifiability and stable recovery of the factors has been studied in many dictionary learning contexts and provides guarantees on the approximate recovery of both an incoherent dictionary and sparse coefficients when the number of samples is sufficiently large (i.e., in our setting when n is large). In [36], the authors developed local optimality conditions in the noiseless case, as well as sample complexity bounds for local recovery when $M_1(h_1)$ is square and $M_2(h_2)$ are i.i.d. Bernoulli-Gaussian. This was extended to overcomplete dictionaries in [33] (see also [71] for tight frames) and to the noisy case in [43]. The authors of [75] provide exact recovery results for dictionary learning when the coefficient matrix has Bernoulli–Gaussian entries and the dictionary matrix has full column rank. This was extended to overcomplete dictionaries in [1] and in [6] but only for approximate recovery. Finally, [35] provides such guarantees under general conditions which cover many practical settings.

Contributions in these frameworks. The present article considers the identifiability and stability of the recovery for any $K \geq 1$ in a general and unifying framework. As was already mentioned, we do not investigate computational issues. As will appear later in the paper, the analogue of the lifting at the core of the algorithms described in the above papers (in particular the papers on phase retrieval and self-calibration) is a *tensorial lifting* (see section 5) and involves tensors that cannot be manipulated in practice. Even when we are able to manipulate the tensors, the computation of the best rank 1 approximation of such tensors is an open nonconvex problem. Therefore, there is no numerically efficient and reliable way to extract the unlifted parameters from an optimized tensor. Because of that, we have not yet pursued the construction of a numerical scheme based on the tensorial lifting when $K \geq 3$. As was already mentioned, as of this writing, the success of algorithms for $K \geq 3$ is mostly supported by empirical evidence. Proving their efficiency is a wide open problem (see

[55, 37, 22, 23, 69, 10, 11, 44, 80]). The purpose of the paper is to provide guarantees on the stability of the solution when such an empirical success occurs.

The specialization of the presented results to problems with $K = 2$ leads to necessary and sufficient conditions for the stable recovery. This is slightly different from the usual approach. Usually, authors provide sufficient conditions and argue their sharpness by comparing the number of samples required by their method and the information theoretic limit (typically, the number of independent variables of the problem).

It would of course be interesting to see how far it is possible to unify the different problems with $K = 2$ using the framework of this paper. We have not, however, pursued this route and instead focus on the situation $K \geq 3$.

2.3. $K \geq 3$. The difficulties, when $K \geq 3$, come from the fact that tools used for problems with $K = 2$ are not applicable. In particular, we cannot use the usual lifting, the singular value decomposition, or the sin-θ theorem in [28]. Often, these tools are replaced by analogous objects involving tensors. This complicates the analysis and prohibits the use of numerical schemes that manipulate lifted variables.

To the best of our knowledge, little is known concerning the identifiability and the stability of matrix factorization when $K \geq 3$. The uniqueness of the factorization corresponding to the fast Fourier transform was proved in [57]. Other results consider the identifiability of the factors which are sparse and random [65]. The authors of the present paper have announced preliminary versions of the results described here in [60]. They are significantly extended here.

3. Notation and summary of the hypotheses.

We continue to use the notation introduced in the introduction. For an integer $k \in \mathbb{N}$, set $[k] = \{1, \dots, k\}$.

We consider $K \geq 1$ and $S \geq 2$ and real-valued tensors of order K whose axes are of size S, denoted $T \in \mathbb{R}^{S \times \cdots \times S}$. The space of tensors is abbreviated $\mathbb{R}^{S^K}$. The entries of T are denoted $T_{i_1,\dots,i_K}$, where $(i_1, \dots, i_K) \in [S]^K$. For $\mathbf{i} \in [S]^K$, the entries of $\mathbf{i}$ are $\mathbf{i} = (i_1, \dots, i_K)$ (for $\mathbf{j} \in [S]^K$ we let $\mathbf{j} = (j_1, \dots, j_K)$, etc). We either write $T_{\mathbf{i}}$ or $T_{i_1,\dots,i_K}$.

To simplify notation, from now on, the parameters defining the factors are gathered in a single matrix and are denoted with bold fonts, $\mathbf{h} \in \mathbb{R}^{S \times K}$. The kth vector containing the parameters for the layer k is denoted $\mathbf{h}_k \in \mathbb{R}^S$. The ith entry of the kth vector is denoted $\mathbf{h}_{k,i} \in \mathbb{R}$. A vector not related to an element in $\mathbb{R}^{S \times K}$ is denoted $h \in \mathbb{R}^S$ (i.e., using a light font). Throughout the paper we assume

$$\mathcal{M} = (\mathcal{M}^L)_{L \in \mathbb{N}}, \text{ with } \mathcal{M}^L \subset \mathbb{R}^{S \times K}.$$

We also assume that, for all $L \in \mathbb{N}$, $\mathcal{M}^L \neq \emptyset$. They can, however, be equal or constant after a given L'.

All the vector spaces $\mathbb{R}^{S^K}$, $\mathbb{R}^{S \times K}$, $\mathbb{R}^S$ etc., are equipped with the usual Euclidean norm. This norm is denoted $\|.\|$ and the scalar product $\langle ., . \rangle$. In the particular case of matrices, $\|.\|$ corresponds to the Frobenius norm. We also use the usual p norm, for $p \in [1, \infty]$, and denote it by $\|.\|_p$. In particular, for $\mathbf{h} \in \mathbb{R}^{S \times K}$ and $T \in \mathbb{R}^{S^K}$, we have for $p < +\infty$

$$\|\mathbf{h}\|_p = \left(\sum_{k=1}^{K} \sum_{i=1}^{S} |\mathbf{h}_{k,i}|^p \right)^{1/p}, \qquad \|T\|_p = \left(\sum_{\mathbf{i} \in [S]^K} |T_{\mathbf{i}}|^p \right)^{1/p}$$

and

$$\|\mathbf{h}\|_\infty = \max_{\substack{k\in[K]\\ i\in[S]}} |\mathbf{h}_{k,i}|, \qquad \|T\|_\infty = \max_{\mathbf{i}\in[S]^K} |T_{\mathbf{i}}|.$$

Set

$$\mathbb{R}_*^{S\times K} = \{\mathbf{h}\in\mathbb{R}^{S\times K} \mid \forall k\in[K], \|\mathbf{h}_k\|\neq 0\}. \tag{3.1}$$

Define an equivalence relation on $\mathbb{R}_*^{S\times K}$: For any $\mathbf{h}$, $\mathbf{g}\in\mathbb{R}^{S\times K}$, $\mathbf{h}\sim\mathbf{g}$ if and only if there exist $(\lambda_k)_{k\in[K]}\in\mathbb{R}^K$ such that

$$\prod_{k=1}^K \lambda_k = 1 \qquad \text{and} \qquad \mathbf{h}_k = \lambda_k \mathbf{g}_k \quad \forall k\in[K]. \tag{3.2}$$

Denote the equivalence class of $\mathbf{h}\in\mathbb{R}_*^{S\times K}$ by $\langle\mathbf{h}\rangle$.

The zero tensor is of rank 0. A nonzero tensor $T\in\mathbb{R}^{S^K}$ is of *rank* 1 (or decomposable) if and only if there exists $\mathbf{h}\in\mathbb{R}_*^{S\times K}$ such that T is the outer product of the vectors $\mathbf{h}_k$ for $k\in[K]$. That is, for any $\mathbf{i}\in[S]^K$,

$$T_{\mathbf{i}} = \mathbf{h}_{1,i_1}\cdots\mathbf{h}_{K,i_K}.$$

Let $\Sigma_1\subset\mathbb{R}^{S^K}$ denote the set of tensors of rank 0 or 1.

The *rank* of a tensor $T\in\mathbb{R}^{S^K}$ is

$$\operatorname{rk}(T) = \min\{r\in\mathbb{N} \mid \text{ there exists } T_1,\dots,T_r\in\Sigma_1 \text{ such that } T = T_1+\cdots+T_r\}.$$

For $r\in\mathbb{N}$, let

$$\Sigma_r = \{T\in\mathbb{R}^{S^K} \mid \operatorname{rk}(T)\le r\}.$$

The $*$ *super*script refers to optimal solutions. A set with a $*$ *sub*script means that 0 is ruled out of the set. In particular, $\Sigma_{1,*}$ denotes the nonzero tensors of rank 1. Attention should be paid to $\mathbb{R}_*^{S\times K}$ (see (3.1)).

4. Facts on the Segre embedding and tensors of rank 1 and 2. Parameterize $\Sigma_1\subset\mathbb{R}^{S^K}$ by the map

$$\begin{array}{rccc} P: & \mathbb{R}^{S\times K} & \longrightarrow & \Sigma_1\subset\mathbb{R}^{S^K} \\ & \mathbf{h} & \longmapsto & (\mathbf{h}_{1,i_1}\mathbf{h}_{2,i_2}\cdots\mathbf{h}_{K,i_K})_{\mathbf{i}\in[S]^K}. \end{array} \tag{4.1}$$

The map P is called the Segre embedding and is often denoted $\widehat{Seg}$ in the algebraic geometry literature.

Standard Facts.

1. *Identifiability of* $\langle\mathbf{h}\rangle$ *from* $P(\mathbf{h})$*:* For $\mathbf{h}$ and $\mathbf{g}\in\mathbb{R}_*^{S\times K}$, $P(\mathbf{h}) = P(\mathbf{g})$ if and only if $\langle\mathbf{h}\rangle = \langle\mathbf{g}\rangle$.
2. *Geometrical description of* $\Sigma_{1,*}$*:* $\Sigma_{1,*}$ is a smooth (i.e., C^∞) manifold of dimension $K(S-1)+1$ (see, e.g., [47, Chapter 4, p. 103]).

3. *Geometrical description of* Σ_2*:* We recall that the singular locus $(\overline{\Sigma}_2)_{sing}$ of the closure $\overline{\Sigma}_2$ of Σ_2 has dimension strictly less than that of $\overline{\Sigma}_2$ and that $\overline{\Sigma}_2 \backslash (\overline{\Sigma}_2)_{sing}$ is a smooth manifold. The dimension of $\overline{\Sigma}_2 \backslash (\overline{\Sigma}_2)_{sing}$ is $2K(S-1)+2$ when $K > 2$, and is $4(S-1)$ when $K = 2$ (see, e.g., [47, Chapter 5]).

We can improve Standard Fact 1 and obtain a stability result guaranteeing that if we know a rank 1 tensor sufficiently close to $P(\mathbf{h})$, we approximately know $\langle \mathbf{h} \rangle$. In order to state this, we need to define a metric on $\mathbb{R}_*^{S\times K}/\sim$ (where $\sim$ is defined by (3.2)). This has to be considered with care since, whatever $\mathbf{h} \in \mathbb{R}_*^{S\times K}$, the subset $\{h \mid h \in \langle \mathbf{h} \rangle\}$ is not compact. In particular, considering

$$\mathbf{h}'_k = \begin{cases} \lambda\, \mathbf{h}_k & \text{if } k = 1, \\ \lambda^{-\frac{1}{K-1}}\, \mathbf{h}_k & \text{otherwise,} \end{cases}$$

when λ goes to infinity, we easily construct examples that make the standard metric on equivalence classes useless.[8]

This leads us to consider

$$\mathbb{R}_{\text{diag}}^{S\times K} = \{\mathbf{h} \in \mathbb{R}_*^{S\times K} \mid \forall k \in [K], \|\mathbf{h}_k\|_\infty = \|\mathbf{h}_1\|_\infty\}.$$

The interest in this set comes from the fact that, whatever $\mathbf{h} \in \mathbb{R}_*^{S\times K}$, the set $\langle \mathbf{h} \rangle \cap \mathbb{R}_{\text{diag}}^{S\times K}$ is finite. Indeed, if $\mathbf{g} \in \langle \mathbf{h} \rangle \cap \mathbb{R}_{\text{diag}}^{S\times K}$, then $(\lambda_k)_{k\in[K]} \in \mathbb{R}^K$ such that, for all $k \in [K]$, $\mathbf{h}_k = \lambda_k \mathbf{g}_k$ must all satisfy $|\lambda_k| = 1$, i.e., $\lambda_k = \pm 1$.

Definition 4.1. *For any* $p \in [1, \infty]$*, we define the mapping* $d_p : (\mathbb{R}_*^{S\times K}/\sim \times \mathbb{R}_*^{S\times K}/\sim) \to \mathbb{R}$ *by*

$$d_p(\langle \mathbf{h} \rangle, \langle \mathbf{g} \rangle) = \inf_{\substack{\mathbf{h}' \in \langle \mathbf{h} \rangle \cap \mathbb{R}_{diag}^{S\times K} \\ \mathbf{g}' \in \langle \mathbf{g} \rangle \cap \mathbb{R}_{diag}^{S\times K}}} \|\mathbf{h}' - \mathbf{g}'\|_p \qquad \forall \mathbf{h}, \mathbf{g} \in \mathbb{R}_*^{S\times K}.$$

Proposition 4.2. *For any* $p \in [1, \infty]$*,* d_p *is a metric on* $\mathbb{R}_*^{S\times K}/\sim$.

The proof is in the supplementary material (subsection SM1.1) and the public archive [61].

Notice that the equivalence relationship and metric defined above are not adapted to operators M_k allowing invariance such as permutations. More precisely, for some operators M_k, there exist $\mathbf{h}$, $\mathbf{g}$, and a permutation matrix C such that $(\mathbf{h}_k, \mathbf{h}_{k+1}) \neq (\mathbf{g}_k, \mathbf{g}_{k+1})$ and $M_k(\mathbf{h}_k) = M_k(\mathbf{g}_k)C$ and $M_{k+1}(\mathbf{h}_{k+1}) = C^{-1}M_{k+1}(\mathbf{g}_{k+1})$. In such a case, we have

[8] For instance, if $\mathbf{h}$ and $\mathbf{g} \in \mathbb{R}_*^{S\times K}$ are such that $\mathbf{h}_1 = \mathbf{g}_1$, we have

$$\inf_{\mathbf{h}' \in \langle \mathbf{h} \rangle, \mathbf{g}' \in \langle \mathbf{g} \rangle} \|\mathbf{h}' - \mathbf{g}'\|_p = 0$$

even though we might have $\mathbf{h}_2 \neq \mathbf{g}_2$ (and therefore $\langle \mathbf{h} \rangle \neq \langle \mathbf{g} \rangle$). This does not define a metric.

Also, when $\mathbf{h}$ and $\mathbf{g}$ are such that $\mathbf{h}_k \neq \mathbf{g}_k$, whatever $k \in [K]$, we have

$$\sup_{\mathbf{h}' \in \langle \mathbf{h} \rangle} \inf_{\mathbf{g}' \in \langle \mathbf{g} \rangle} \|\mathbf{h}' - \mathbf{g}'\|_p = +\infty.$$

Therefore, the Hausdorff distance between $\langle \mathbf{h} \rangle$ and $\langle \mathbf{g} \rangle$ is infinite for almost every pair $(\mathbf{h}, \mathbf{g})$. This metric is therefore not very useful in the present context.

$d_p(\langle \mathbf{h} \rangle, \langle \mathbf{g} \rangle) \neq 0$. However, the features defined at the layer k are just permuted and can still be interpreted. As already said, in such a case, it is possible to use the models $\mathcal{M}$ to select one of the equally interpretable $\mathbf{h}$.

Using the above metric, we can state that not only is $\langle \mathbf{h} \rangle$ uniquely determined by $P(\mathbf{h})$, but this operation is stable.

Theorem 4.3 (stability of $\langle \mathbf{h} \rangle$ from $P(\mathbf{h})$). *Let* $\mathbf{h}$ *and* $\mathbf{g} \in \mathbb{R}_*^{S \times K}$ *be such that* $\|P(\mathbf{g}) - P(\mathbf{h})\|_\infty \leq \frac{1}{2} \max(\|P(\mathbf{h})\|_\infty, \|P(\mathbf{g})\|_\infty)$. *For all* $p, q \in [1, \infty]$,

$$d_p(\langle \mathbf{h} \rangle, \langle \mathbf{g} \rangle) \leq 7(KS)^{\frac{1}{p}} \min\left(\|P(\mathbf{h})\|_\infty^{\frac{1}{K}-1}, \|P(\mathbf{g})\|_\infty^{\frac{1}{K}-1} \right) \|P(\mathbf{h}) - P(\mathbf{g})\|_q. \tag{4.2}$$

The proof of the theorem is in the supplementary material (subsection SM1.2) and in [61].

In the final result, the bound established in Theorem 4.3 plays a role similar to the $\sin-\theta$ Theorem of [28] in [54, 18, 2].

The following proposition shows that the upper bound in (4.2) cannot be improved by a significant factor, in particular when q is large.

Proposition 4.4. *There exist* $\mathbf{h}$ *and* $\mathbf{g} \in \mathbb{R}_*^{S \times K}$ *such that* $\|P(\mathbf{g})\|_\infty \leq \|P(\mathbf{h})\|_\infty$, $\|P(\mathbf{g}) - P(\mathbf{h})\|_\infty \leq \frac{1}{2} \|P(\mathbf{h})\|_\infty$, *and*

$$7(KS)^{\frac{1}{p}} \|P(\mathbf{h})\|_\infty^{\frac{1}{K}-1} \|P(\mathbf{h}) - P(\mathbf{g})\|_q \leq C_q \, d_p(\langle \mathbf{h} \rangle, \langle \mathbf{g} \rangle),$$

where

$$C_q = \begin{cases} 28(KS)^{\frac{1}{q}} & \text{if } q < +\infty, \\ 28 & \text{if } q = +\infty. \end{cases}$$

The proof of the proposition is in the supplementary material (subsection SM1.3) and in [61].

As stated in the following theorem, we have a more valuable upper bound in the general case.

Theorem 4.5 ("Lipschitz continuity" of P). *For any* $q \in [1, \infty]$ *and any* $\mathbf{h}$ *and* $\mathbf{g} \in \mathbb{R}_*^{S \times K}$,

$$\|P(\mathbf{h}) - P(\mathbf{g})\|_q \leq S^{\frac{K-1}{q}} K^{1-\frac{1}{q}} \max\left(\|P(\mathbf{h})\|_\infty^{1-\frac{1}{K}}, \|P(\mathbf{g})\|_\infty^{1-\frac{1}{K}} \right) d_q(\langle \mathbf{h} \rangle, \langle \mathbf{g} \rangle). \tag{4.3}$$

The theorem is proved in the supplementary material (subsection SM1.4) and in [61].

Notice that, considering $\mathbf{h}$ and $\mathbf{g} \in \mathbb{R}^{S \times K}$ such that $\mathbf{h}_{k,i} = 1$ and $\mathbf{g}_{k,i} = \varepsilon$, for all $k \in [K]$ and $i \in [S]$ and for a $0 < \varepsilon \ll 1$, we easily calculate

$$S^{\frac{K-1}{q}} K^{1-\frac{1}{q}} \max\left(\|P(\mathbf{h})\|_\infty^{1-\frac{1}{K}}, \|P(\mathbf{g})\|_\infty^{1-\frac{1}{K}} \right) d_q(\langle \mathbf{h} \rangle, \langle \mathbf{g} \rangle) \leq K \|P(\mathbf{h}) - P(\mathbf{g})\|_q.$$

As a consequence, the upper bound in Theorem 4.5 is tight up to at most a factor K.

5. The tensorial lifting. The following proposition is clear (it can be shown by induction on K).

Proposition 5.1. *Let M_k, $k \in [K]$, be as in (1.2). The entries of the matrix*

$$M_1(\mathbf{h}_1)M_2(\mathbf{h}_2)\cdots M_K(\mathbf{h}_K)$$

are multivariate polynomials whose variables are the entries of $\mathbf{h} \in \mathbb{R}^{S\times K}$. Moreover, every entry is the sum of monomials of degree K. Each monomial is a constant times $\mathbf{h}_{1,i_1}\cdots\mathbf{h}_{K,i_K}$ for some $\mathbf{i} \in [S]^K$.

Notice that any monomial $\mathbf{h}_{1,i_1}\cdots\mathbf{h}_{K,i_K}$ is the entry $P(\mathbf{h})_\mathbf{i}$ in the tensor $P(\mathbf{h})$. Therefore every polynomial in the previous proposition takes the form $\sum_{\mathbf{i}\in[S]^K} c_\mathbf{i} P(\mathbf{h})_\mathbf{i}$ for some constants $(c_\mathbf{i})_{\mathbf{i}\in[S]^K}$ independent of $\mathbf{h}$. In words, every entry of the matrix $M_1(\mathbf{h}_1)M_2(\mathbf{h}_2)\cdots M_K(\mathbf{h}_K)$ is obtained by applying a linear form to $P(\mathbf{h})$. Moreover, the polynomial coefficients defining the linear form are uniquely determined by the linear maps $M_1,\dots,M_K$. This leads to the following statement.

Corollary 5.2 (tensorial lifting). *Let M_k, $k \in [K]$ be as in (1.2). The map*

$$(\mathbf{h}_1,\dots,\mathbf{h}_K) \longmapsto M_1(\mathbf{h}_1)M_2(\mathbf{h}_2)\cdots M_K(\mathbf{h}_K)$$

uniquely determines a linear map

$$\mathcal{A} : \mathbb{R}^{S^K} \longrightarrow \mathbb{R}^{m\times n},$$

such that for all $\mathbf{h} \in \mathbb{R}^{S\times K}$

$$M_1(\mathbf{h}_1)M_2(\mathbf{h}_2)\cdots M_K(\mathbf{h}_K) = \mathcal{A}P(\mathbf{h}). \tag{5.1}$$

We call (5.1) and its use the *tensorial lifting*. When $K = 1$, we simply have $\mathcal{A} = M_1$. When $K = 2$, it corresponds to the usual lifting already exploited to establish stability results for phase recovery, blind-deconvolution, self-calibration, sparse coding, etc. Notice that when $K \geq 2$, it may be difficult to provide a closed form expression for the operator $\mathcal{A}$. We can, however, determine simple properties of $\mathcal{A}$. In most reasonable cases, $\mathcal{A}$ is sparse. If the operators M_k simply embed the values of h in a matrix, the matrix representing $\mathcal{A}$ only contains zeros and ones. Since the operators M_k are known, we can compute $\mathcal{A}P(\mathbf{h})$, for any $\mathbf{h} \in \mathbb{R}^{S\times K}$, using (5.1). Said differently, we can compute $\mathcal{A}$ for any rank 1 tensor. Therefore, since $\mathcal{A}$ is linear, we can compute $\mathcal{A}T$ for any low rank tensor T. If the dimensions of the problem permit, one can manipulate $\mathcal{A}$ in a basis of $\mathbb{R}^{S^K}$.

Since $\mathrm{rk}\,(\mathcal{A})$ is an important quantity, we emphasize that $\mathrm{rk}\,(\mathcal{A}) \leq mn$. It is also possible to compute $\mathrm{rk}\,(\mathcal{A})$, when mn is not too large, using the following proposition.

Proposition 5.3. *For R independent random $\mathbf{h}^r$, with $r = 1,\dots,R$, according to the normal distribution in $\mathbb{R}^{S\times K}$, we have with probability* 1

$$\dim(\mathrm{Span}\,((\mathcal{A}P(\mathbf{h}^r))_{r=1,\dots,R})) = \begin{cases} R & \text{if } R \leq \mathrm{rk}\,(\mathcal{A}), \\ \mathrm{rk}\,(\mathcal{A}) & \text{otherwise.} \end{cases} \tag{5.2}$$

The proof is in the supplementary material (subsection SM1.5) and the public archive [61].

Using Corollary 5.2, when (2.1) has a minimizer, we rewrite it in the form

$$\mathbf{h}^* \in \operatorname{argmin}_{L\in\mathbb{N},\mathbf{h}\in\mathcal{M}^L} \|\mathcal{A}P(\mathbf{h}) - X\|^2. \tag{5.3}$$

We now decompose this problem into two subproblems: a least-squares problem,

$$T^* \in \operatorname{argmin}_{T\in\mathbb{R}^{S^K}} \|\mathcal{A}T - X\|^2, \tag{5.4}$$

and a nonconvex problem,

$$\mathbf{h}'^* \in \operatorname{argmin}_{L\in\mathbb{N},\mathbf{h}\in\mathcal{M}^L} \|\mathcal{A}(P(\mathbf{h}) - T^*)\|^2. \tag{5.5}$$

Proposition 5.4. *Let X and $\mathcal{A}$ be such that (2.1) has a minimizer:*

1. *Let $\mathbf{h}^*$ be a solution of (5.3). Then, for any solution T^* of (5.4), $\mathbf{h}^*$ also minimizes (5.5).*
2. *Let T^* be a solution of (5.4) and $\mathbf{h}'^*$ a solution of (5.5). Then $\mathbf{h}'^*$ also minimizes (5.3).*

The proof is in the supplementary material (subsection SM1.6) and the public archive [61].

From now on, because of the equivalence between solutions of (5.5) and (5.3), we stop using the notation $\mathbf{h}'^*$ and write $\mathbf{h}^* \in \operatorname{argmin}_{L\in\mathbb{N},\mathbf{h}\in\mathcal{M}^L} \|\mathcal{A}(P(\mathbf{h}) - T^*)\|^2$.

6. Identifiability (error-free case). Throughout this section, we assume that X is such that there exist $\overline{L}$ and $\overline{\mathbf{h}} \in \mathcal{M}^{\overline{L}}$ such that

$$X = M_1(\overline{\mathbf{h}}_1)\cdots M_K(\overline{\mathbf{h}}_K). \tag{6.1}$$

Under this assumption, $X = \mathcal{A}P(\overline{\mathbf{h}})$, so

$$P(\overline{\mathbf{h}}) \in \operatorname{argmin}_{T\in\mathbb{R}^{S^K}} \|\mathcal{A}T - X\|^2.$$

Moreover, we trivially have $P(\overline{\mathbf{h}}) \in \Sigma_1$, and therefore $\overline{\mathbf{h}}$ minimizes (5.5), (2.1), and (5.3). As a consequence, (2.1) has a minimizer.

We ask whether there exist guarantees that the resolution of (2.1) allows one to recover $\overline{\mathbf{h}}$ up to the usual uncertainties.

In this regard, for any $\mathbf{h} \in \langle\overline{\mathbf{h}}\rangle$, we have $P(\mathbf{h}) = P(\overline{\mathbf{h}})$ and therefore $\mathcal{A}P(\mathbf{h}) = \mathcal{A}P(\overline{\mathbf{h}}) = X$. Thus unless we make further assumptions on $\overline{\mathbf{h}}$, we cannot expect to distinguish any particular element of $\langle\overline{\mathbf{h}}\rangle$ using only X. In other words, recovering $\langle\overline{\mathbf{h}}\rangle$ is the best we can hope for.

Definition 6.1 (identifiability). *We say that* $\langle\overline{\mathbf{h}}\rangle$ is identifiable *if the elements of* $\langle\overline{\mathbf{h}}\rangle$ *are the only solutions of (2.1).*

We say that $\mathcal{M}$ is identifiable *if for every $L \in \mathbb{N}$ and every $\overline{\mathbf{h}} \in \mathcal{M}^L$, $\langle\overline{\mathbf{h}}\rangle$ is identifiable.*

Proposition 6.2 (characterization of the global minimizers). *For any $L^* \in \mathbb{N}$ and any $\mathbf{h}^* \in \mathcal{M}^{L^*}$, $(L^*, \mathbf{h}^*) \in \operatorname{argmin}_{L\in\mathbb{N},\mathbf{h}\in\mathcal{M}} \|\mathcal{A}P(\mathbf{h}) - X\|^2$ if and only if*

$$P(\mathbf{h}^*) \in P(\overline{\mathbf{h}}) + \operatorname{Ker}(\mathcal{A}).$$

The proposition is proved in the supplementary material (subsection SM1.7) and in [61].

In order to state the following proposition, we define for any L and $L' \in \mathbb{N}$

$$P(\mathcal{M}^L) - P(\mathcal{M}^{L'}) := \left\{ P(\mathbf{h}) - P(\mathbf{g}) \mid \quad \mathbf{h} \in \mathcal{M}^L \text{ and } \mathbf{g} \in \mathcal{M}^{L'} \right\} \subset \mathbb{R}^{S^K}.$$

Proposition 6.3 (necessary and sufficient conditions of identifiability).

1. *For any* $\overline{L}$ *and* $\overline{\mathbf{h}} \in \mathcal{M}^{\overline{L}}$, $\langle \overline{\mathbf{h}} \rangle$ *is identifiable if and only if for any* $L \in \mathbb{N}$

$$\left(P(\overline{\mathbf{h}}) + \operatorname{Ker}(\mathcal{A})\right) \cap P(\mathcal{M}^L) \ \subset \ \{P(\overline{\mathbf{h}})\}.$$

2. $\mathcal{M}$ *is identifiable if and only if for any* L *and* $L' \in \mathbb{N}$

$$\operatorname{Ker}(\mathcal{A}) \cap \left(P(\mathcal{M}^L) - P(\mathcal{M}^{L'})\right) \subset \{0\}. \tag{6.2}$$

The proposition is proved in the supplementary material (subsection SM1.8) and in [61].

In the context of the usual compressed sensing (i.e., when $K = 1$, $\mathcal{M}$ contains L-sparse signals, $\mathcal{A}$ is a rectangular matrix with full row rank, and X is a vector), the proposition is already stated in Lemma 3.1 of [25].

In reasonably small cases and when $P(\mathcal{M})$ is algebraic, one can use tools from numerical algebraic geometry such as those described in [39, 40] to check whether condition (6.2) holds or not. The drawback of Proposition 6.3 is that, given a deep structured linear network as described by $\mathcal{A}$, condition (6.2) might be difficult to verify.

We therefore establish simpler conditions related to the identifiability of $\mathcal{M}$. First we establish a condition such that for almost every $\mathcal{A}$ satisfying it, $\mathcal{M}$ is identifiable. The main benefit of this condition is that its constituents can be computed in many practical situations.

Before that, we recall a few facts of algebraic geometry: For $X, Y \subset \mathbb{R}^N$, the *join* of X and Y (see, e.g., [38, Example 8.1]) is

$$J(X,Y) := \overline{\{sx + ty \mid x \in X, \ y \in Y, \ s,t \in \mathbb{R}\}}.$$

If, for all $L \in \mathbb{N}$, $\mathcal{M}^L$ is Zariski closed and invariant under rescaling (e.g., if they are all linear spaces), then $P(\mathcal{M}^L) - P(\mathcal{M}^{L'})$ is a Zariski open subset of $J(P(\mathcal{M}^L), P(\mathcal{M}^{L'}))$. In general, it is contained in this join.

Recall the following fact (*): for complex algebraic varieties $X, Y \subset \mathbb{C}^N$, any component Z of $X \cap Y$ has $\dim(Z) \geq \dim(X) + \dim(Y) - N$, and equality holds generically (we make "generically" precise in our context below). Moreover, if X, Y are invariant under rescaling, since $0 \in X \cap Y$, we have $X \cap Y \neq \emptyset$. (See, e.g., [73, section I.6.2].)

This intersection result indicates that if there exists L, L' such that

$$\operatorname{rk}(\mathcal{A}) < \dim\left(P(\mathcal{M}^L) - P(\mathcal{M}^{L'})\right),$$

we expect to have nonidentifiability; and if the rank is larger, for every pair L, L', we expect identifiability. The following theorem states this more precisely.

Theorem 6.4 (almost surely sufficient condition for identifiability). *For almost every* $\mathcal{A}$ *such that*

$$\operatorname{rk}(\mathcal{A}) \geq \dim\left(J(P(\mathcal{M}^L), P(\mathcal{M}^{L'}))\right) \qquad \forall L, L',$$

$\mathcal{M}$ *is identifiable.*

The theorem is proved in the supplementary material (subsection SM1.9) and in [61].

Since $\dim\left(J(P(\mathcal{M}^L), P(\mathcal{M}^{L'}))\right) \leq \dim\left(P(\mathcal{M}^L)\right) + \dim\left(P(\mathcal{M}^{L'})\right)$, if D_{max} is the maximum dimension of $P(\mathcal{M}^L)$ over all L, one has the same conclusion if $\mathrm{rk}(\mathcal{A}) \geq 2D_{max}$.

When $K = 1$, we illustrate this result by interpreting it in the context of compressive sensing, where $\mathbf{h}$ is a vector, X is a vector, $\mathcal{A}$ is a rectangular sampling matrix of full row rank, and $\mathrm{Ker}\,(\mathcal{A})$ is large. The statement analogous to Theorem 6.4 in the compressive sensing framework takes the following form: "For almost every sampling matrix, any L sparse signal $\mathbf{h}$ can be recovered from $\mathcal{A}\mathbf{h}$ as soon as $2L \leq \mathrm{rk}\,(\mathcal{A})$." Moreover, the constituents of the ℓ^0 minimization model used to recover the signal are also the constituents of (5.3). Again, the main novelty is to extend this result to the identifiability of the factors of deep matrix products.

In order to establish a necessary condition for identifiability, first note that if we extend $P(\mathcal{M}^L) - P(\mathcal{M}^{L'})$ to be scale invariant, this will not affect whether or not it intersects $\ker(\mathcal{A})$ outside of the origin. We immediately conclude that in the complex setting where $\mathcal{M}^L, \mathcal{M}^{L'}$ are both Zariski closed, $\mathcal{M}$ is nonidentifiable whenever $\mathrm{rk}(\mathcal{A}) < \dim\left(P(\mathcal{M}^L) - P(\mathcal{M}^{L'})\right)$. This indicates that we should always expect nonidentifiability whenever $\mathrm{rk}(\mathcal{A}) < \dim\left(P(\mathcal{M}^L) - P(\mathcal{M}^{L'})\right)$, but is not adequate to prove it because real algebraic varieties need not satisfy (*). However, it is true for real linear spaces, so we immediately conclude the following weak result.

Theorem 6.5 (necessary condition for identifiability). *Let $C(P(\mathcal{M}^L) - P(\mathcal{M}^{L'}))$ be the set of all points on all lines through the origin intersecting $P(\mathcal{M}^L) - P(\mathcal{M}^{L'})$, and let q be the maximal dimension of a linear space on $C(P(\mathcal{M}^L) - P(\mathcal{M}^{L'}))$. Then if $q > \mathrm{rk}(\mathcal{A})$, $\mathcal{M}$ is not identifiable. In particular when the $P(\mathcal{M}^L)$'s contain linear space and if we let S' be the largest dimension of these vector spaces, if $2S' > \mathrm{rk}(\mathcal{A})$, then $\mathcal{M}$ is not identifiable.*

Let us illustrate the theorems by considering a deep feed-forward ReLU network and consider the structured linear network obtained by fixing the action of ReLU (as is done in subsection 1.2.2). The matrix X contains the outputs, and the operator M_K multiplies the matrix containing the inputs by the weights between the first and the second layer. For every input/output pair, the action of ReLU is different and removes paths from the input entries to the output entries. We assume, however, that every entry of every output is reached by at least one path in the network starting at a nonzero entry of the input. In that case, it is not difficult to see that $\mathcal{A}$ is a surjection, and therefore $\mathrm{rk}(\mathcal{A}) = mn$, where m is the size of the output and n is the number of learning samples.

The condition in Theorem 6.4 becomes

$$mn \geq \dim\left(\Sigma_2\right) = 2K(S-1) + 2,$$

and KS is typically the number of parameters of the network. The intuition behind Theorem 6.4 is that if the action of ReLU is sufficiently random and if the above inequality holds, we can expect the network to be identifiable with high probability.[9] This situation corresponds to an underparameterized case (favorable for identifiability).

[9]This statement gives the intuition behind Theorem 6.4 but it should be made precise, as emphasized in the perspectives of this paper.

The condition in Theorem 6.5 is

$$2S > mn.$$

When this inequality holds, the network is not identifiable. It corresponds to an overparameterized configuration. In the intermediate situation, when $2S \leq mn < 2K(S-1)+2$, and when the action of the activation function does not introduce sufficiently randomness, the theorems are inconclusive.

Notice that such networks can also be analyzed using Proposition 6.3. It is indeed not difficult to see that if there exist two paths that (1) start from the same entry of the input layer, (2) end at the same entry of the output layer, and (3) are both present (despite the action of ReLU) for every input/output pair, then (6.2) does not hold[10] and $\mathbb{R}^{S\times K}$ is not identifiable. It is not clear at this point that conditions 1, 2, 3 are met by all nonidentifiable structured linear feed-forward networks. However, removing paths from the network (as is done by ReLU and Dropout) is a way to avoid conditions 1, 2, 3 being met.

7. Stability guarantee. In this section, we consider errors of different natures. We assume that there exist $\overline{L}$ and $L^* \in \mathbb{N}$, $\overline{\mathbf{h}} \in \mathcal{M}^{\overline{L}}$, and $\mathbf{h}^* \in \mathcal{M}^{L^*}$, such that

$$\|M_1(\overline{\mathbf{h}}_1)\cdots M_K(\overline{\mathbf{h}}_K) - X\| \leq \delta \tag{7.1}$$

and

$$\|M_1(\mathbf{h}_1^*)\cdots M_K(\mathbf{h}_K^*) - X\| \leq \eta \tag{7.2}$$

for δ and η typically small.

Again, this corresponds to existing unknown parameters $\overline{\mathbf{h}}$ that we estimate from a noisy observation X, using an inaccurate solution $\mathbf{h}^*$ of (2.1) (as in [13], where the case $K=1$ is studied). Otherwise, $\overline{\mathbf{h}}$ and $\mathbf{h}^*$ shall be interpreted as different learned parameters; δ and η are the corresponding risks.

Notice that the above hypothesis does not even require (2.1) to have a solution. Algorithms which do not come with a guarantee sometimes manage to reach small δ and η values. In those cases, the analysis we conduct in this section permits us to get the stability guarantee, despite the lack of a guarantee of the algorithm. Finally, hypotheses (7.1) and (7.2) enable one to obtain guarantees for algorithms that, instead of minimizing (2.1), minimize an objective function which approximates the one in (2.1). This is particularly relevant for machine learning applications when (2.1) can be an empirical risk that needs to be regularized or is not truly minimized (for instance, when using *Dropout* [76]).

A necessary and sufficient condition for the identifiability of $\mathcal{M}$ is stated in Proposition 6.3. The condition is on the way that $\mathrm{Ker}\,(\mathcal{A})$ and $P(\mathcal{M}^L) - P(\mathcal{M}^{L'})$ intersect. In order to get a stability guarantee, we need a stronger condition on the geometry of this intersection to hold for every L and $L' \in \mathbb{N}$. This condition is provided in the next definition.

Definition 7.1 (deep-Null Space Property). *Let $\gamma > 0$ and $\rho > 0$. We say that* $\mathrm{Ker}\,(\mathcal{A})$ *satisfies the* deep-Null Space Property (deep-NSP) *with respect to the collection of models $\mathcal{M}$*

[10] Simply consider two rank 1 tensors, each tensor being a Dirac at the position corresponding to one of the two paths.

with constants (γ, ρ) *if for any* L *and* $L' \in \mathbb{N}$, *any* $T \in P(\mathcal{M}^L) - P(\mathcal{M}^{L'})$ *satisfying* $\|\mathcal{A}T\| \leq \rho$, *and any* $T' \in \operatorname{Ker}(\mathcal{A})$, *we have*

$$\|T\| \leq \gamma \|T - T'\|. \tag{7.3}$$

The deep-NSP implies that, for $T \in P(\mathcal{M}^L) - P(\mathcal{M}^{L'})$ close to $\operatorname{Ker}(\mathcal{A})$ in the sense that $\|\mathcal{A}T\| \leq \rho$, we must have, by decomposing $T = T' + T''$, with $T' \in \operatorname{Ker}(\mathcal{A})$ and T'' in its orthogonal complement,

$$\|T\| \leq \gamma \|T - T'\| = \gamma \|T''\| \leq \frac{\gamma}{\sigma_{min}} \|\mathcal{A}T''\| \leq \frac{\gamma}{\sigma_{min}} \rho,$$

where σ_{min} is the smallest nonzero singular value of $\mathcal{A}$. In words, $\|T\|$ must be small. We can conclude that under the deep-NSP, $P(\mathcal{M}^L) - P(\mathcal{M}^{L'})$ and $\{T \in \mathbb{R}^{S^K} \mid \|\mathcal{A}T\| \leq \rho\}$ intersect at most in the neighborhood of 0.

Additionally, (7.3) implies that in the neighborhood of 0, $\operatorname{Ker}(\mathcal{A})$ and $P(\mathcal{M}^L) - P(\mathcal{M}^{L'})$ are not tangential; i.e., their intersection is transverse.

If $\operatorname{Ker}(\mathcal{A})$ satisfies the deep-NSP with respect to the collection of models $\mathcal{M}$ with constants (γ, ρ), then for all $T' \in \operatorname{Ker}(\mathcal{A})$ and all $T \in P(\mathcal{M}^L) - P(\mathcal{M}^{L'})$ satisfying $\|\mathcal{A}T\| \leq \rho$,

$$\|T'\| \leq \|T\| + \|T' - T\| \leq (\gamma + 1)\|T' - T\|.$$

Therefore,

$$\forall T' \in \operatorname{Ker}(\mathcal{A}), \qquad \|T'\| \leq (\gamma + 1) d_{loc}(T', P(\mathcal{M}^L) - P(\mathcal{M}^{L'})), \tag{7.4}$$

where we have set for any $C \subset \mathbb{R}^{S^K}$

$$d_{loc}(T', C) = \inf_{T \in C, \|\mathcal{A}T\| \leq \rho} \|T' - T\|.$$

The converse is also true: If $\operatorname{Ker}(\mathcal{A})$ satisfies (7.4), it satisfies the deep-NSP with respect to the collection of models $\mathcal{M}$ with appropriate constants. In the context of the usual compressed sensing (i.e., when $K = 1$, $\mathcal{M}^L$ contains L-sparse signals, $\mathcal{A}$ is a rectangular matrix with full row rank, and X is a vector), the localization appearing in d_{loc} can be discarded since the inequality must hold when T' is small and since in this case this localization has no effect. Therefore, in the compressed sensing context, (7.4) (and therefore deep-NSP) is the usual Null Space Property with respect to L-sparse vectors, as defined in [25]. However, deep-NSP is generalized to take into account deep structured linear network. This motivates the name.

In the general case, the deep-NSP can be understood as a local version of the generalized-NSP for $\mathcal{A}$ relative to $P(\cup_{L \in \mathbb{N}} \mathcal{M}^L) - P(\cup_{L \in \mathbb{N}} \mathcal{M}^L)$, as defined in [13]. Our interest in locality (as imposed by the constraint $\|\mathcal{A}T\| \leq \rho$) is motivated by the fact that we want to use the deep-NSP when the signal-to-noise ratio is controlled (i.e., the hypotheses of Theorem 4.3 are satisfied). The condition for the stability property therefore includes such hypotheses.

We have not adapted the robust-NSP defined in [13]. The benefit in not using this definition is to obtain a simpler definition for deep-NSP. In particular (7.3) does not involve the geometry of $\mathcal{A}$ in the orthogonal complement of $\operatorname{Ker}(\mathcal{A})$. Looking in detail at the benefit of this adaptation is of great interest.

Finally, we trivially have the following facts:

- If $\mathrm{Ker}\,(\mathcal{A}) = \{0\}$, then $\mathrm{Ker}\,(\mathcal{A})$ satisfies the deep-NSP with respect to the model $\mathbb{R}^{S\times K}$ with constants $(1, +\infty)$.
- For any $\gamma' \geq \gamma$: If $\mathrm{Ker}\,(\mathcal{A})$ satisfies the deep-NSP with respect to the collection of models $\mathcal{M}$ with constants (γ, ρ), then $\mathrm{Ker}\,(\mathcal{A})$ satisfies the deep-NSP with respect to the collection of models $\mathcal{M}$ with constants (γ', ρ).
- For any $\mathcal{M}' \subset \mathcal{M}$: If $\mathrm{Ker}\,(\mathcal{A})$ satisfies the deep-NSP with respect to the collection of models $\mathcal{M}$ with constants (γ, ρ), then $\mathrm{Ker}\,(\mathcal{A})$ satisfies the deep-NSP with respect to the collection of models $\mathcal{M}'$ with constants (γ, ρ). In particular, if $\mathrm{Ker}\,(\mathcal{A})$ satisfies the deep-NSP with respect to the model $\mathbb{R}^{S\times K}$ with constants (γ, ρ), it satisfies the deep-NSP with respect to any collection of models, with constants (γ, ρ).

Theorem 7.2 (sufficient condition for the stability property). *Assume* $\mathrm{Ker}\,(\mathcal{A})$ *satisfies the deep-NSP with respect to the collection of models* $\mathcal{M}$ *and with the constants* (γ, ρ). *For any* $\mathbf{h}^*$ *as in* (7.2) *with* η *and* δ *(see* (7.2) *and* (7.1)*) such that* $\delta + \eta \leq \rho$, *we have*

$$\|P(\mathbf{h}^*) - P(\overline{\mathbf{h}})\| \leq \frac{\gamma}{\sigma_{min}}\ (\delta + \eta),$$

where σ_{min} *is the smallest nonzero singular value of* $\mathcal{A}$. *Moreover, if* $\overline{\mathbf{h}} \in \mathbb{R}_*^{S\times K}$ *and* $\frac{\gamma}{\sigma_{min}}\ (\delta + \eta) \leq \frac{1}{2}\ \max\left(\|P(\overline{\mathbf{h}})\|_\infty, \|P(\mathbf{h}^*)\|_\infty\right)$, *then*

$$d_p(\langle \mathbf{h}^* \rangle, \langle \overline{\mathbf{h}} \rangle) \leq \frac{7(KS)^{\frac{1}{p}}\gamma}{\sigma_{min}} \min\left(\|P(\overline{\mathbf{h}})\|_\infty^{\frac{1}{K}-1}, \|P(\mathbf{h}^*)\|_\infty^{\frac{1}{K}-1}\right)(\delta + \eta). \tag{7.5}$$

The first part of the proof is very similar to standard proofs in the compressed sensing and stable recovery literature. The second part simply uses Theorem 4.3. The theorem is proved in the supplementary material (subsection SM1.10) and in [61].

Theorem 7.2 provides a sufficient condition to obtain stability. The only significant hypothesis made on the deep structured linear network is that $\mathrm{Ker}\,(\mathcal{A})$ satisfies the deep-NSP with respect to the collection of models $\mathcal{M}$. One might ask whether this hypothesis is sharp or not. The next theorem shows that the answer to this question is positive.

Theorem 7.3 (necessary condition for the stability property). *Assume the stability property holds: There exist* C *and* $\delta > 0$ *such that for any* $\overline{L} \in \mathbb{N}$, $\overline{\mathbf{h}} \in \mathcal{M}^{\overline{L}}$, *any* $X = \mathcal{A}P(\overline{\mathbf{h}}) + e$, *with* $\|e\| \leq \delta$, *any* $L^* \in \mathbb{N}$, *and any* $\mathbf{h}^* \in \mathcal{M}^{L^*}$ *such that*

$$\|\mathcal{A}P(\mathbf{h}^*) - X\|^2 \leq \|e\|,$$

we have

$$d_2(\langle \mathbf{h}^* \rangle, \langle \overline{\mathbf{h}} \rangle) \leq C\ \min\left(\|P(\overline{\mathbf{h}})\|_\infty^{\frac{1}{K}-1}, \|P(\mathbf{h}^*)\|_\infty^{\frac{1}{K}-1}\right)\|e\|.$$

Then $\mathrm{Ker}\,(\mathcal{A})$ *satisfies the deep-NSP with respect to the collection of models* $\mathcal{M}$ *with constants*

$$(\gamma, \rho) = (CS^{\frac{K-1}{2}}\sqrt{K}\ \sigma_{max}, \delta),$$

where σ_{max} *is the spectral radius of* $\mathcal{A}$.

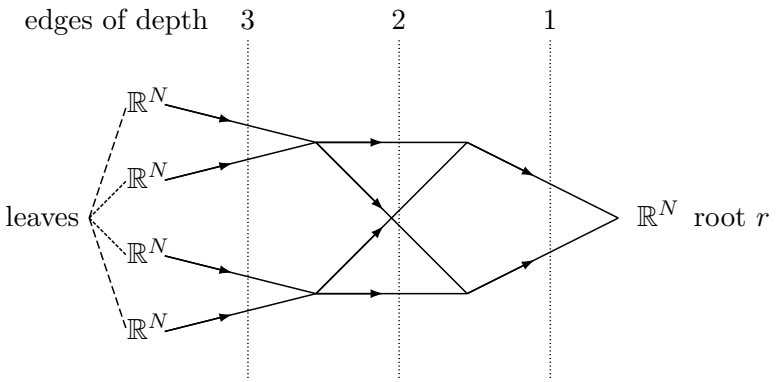

Figure 1. *Example of the considered convolutional linear network. To every edge is attached a convolution kernel. The network does not involve nonlinearities or sampling.*

The first part of the proof was inspired by and is similar to the proof of the analogous converse statement in [25]. The second part simply uses Theorem 4.5. The theorem is proved in the supplementary material (subsection SM1.11) and in [61].

The sharpness of the known results when $K = 2$ is usually argued by comparing the number of samples necessary for the recovery and the information theoretic limit of the problem. As far as the authors know, the above theorem is therefore new even when $K = 2$.

As is usually the case with the Null Space Property or Restricted Isometry Property, it will often be difficult or impossible to establish that a particular operator $\mathcal{A}$ satisfies the deep-NSP with respect to the collection of models $\mathcal{M}$. To find favorable cases, we need to consider random operators $\mathcal{A}$ such that the distribution of $\mathcal{A}$ enables one to establish that the deep-NSP holds with high probability, when in the right configurations (see the bibliography in section 2, whose references contain many examples of such arguments). The most common distribution for the analogue of $\mathcal{A}$ includes operators/matrices whose coefficients are Gaussian or Bernoulli. Collections of models inducing sparsity, nonnegativity, low rank constraints, etc., are the most studied. In this regard, the fact that there exists a low complexity test guaranteeing that the networks considered in section 8 can be stably recovered is an exception.

8. Application to convolutional linear network. We consider a convolutional linear network as depicted in Figure 1. The network typically aims at performing a linear analysis or synthesis of a signal living in $\mathbb{R}^N$. The considered convolutional linear network is defined from a rooted directed acyclic graph $\mathcal{G}(\mathcal{E}, \mathcal{N})$ composed of nodes $\mathcal{N}$ and edges $\mathcal{E}$. Each edge connects two nodes. The root of the graph is denoted by r, and the set containing all its leaves is denoted by $\mathcal{F}$. We denote by $\mathcal{P}$ the set of all paths connecting the leaves and the root. We assume, without loss of generality, that the length of any path between any leaf and the root is independent of the considered leaf and equal to some constant $K \geq 0$. We also assume that, for any edge $e \in \mathcal{E}$, the number of edges separating e and the root is the same for all paths between e and r. It is called the depth of e. We also say that e belongs to the layer k. For any $k \in [K]$, we denote by $\mathcal{E}(k)$ the set containing all the edges of depth k.

Moreover, to any edge e is attached a convolution kernel of support $\mathcal{S}_e \subset [N]$. We assume

(without loss of generality) that $\sum_{e\in\mathcal{E}(k)}|\mathcal{S}_e|$ is independent of k ($|\mathcal{S}_e|$ denotes the cardinality of $\mathcal{S}_e$). We take

$$S=\sum_{e\in\mathcal{E}(1)}|\mathcal{S}_e|.$$

For any edge e, we consider the mapping $\mathcal{T}_e:\mathbb{R}^S\longrightarrow\mathbb{R}^N$ that maps any $h\in\mathbb{R}^S$ into the convolution kernel h_e, attached to the edge e, whose support is $\mathcal{S}_e$. It simply writes at the right location (i.e., those in $\mathcal{S}_e$) the entries of h defining the kernel on the edge e.

At each layer k, the convolutional linear network computes, for all $e\in\mathcal{E}(k)$, the convolution between the signal at the origin of e; then it attaches to any ending node the sum of all the convolutions arriving at that node. Examples of such convolutional linear networks include wavelets, wavelet packets [62], or the fast transforms optimized in [20, 21]. It is clear that the operation performed at any layer depends linearly on the parameters $h\in\mathbb{R}^S$ and that its results serve as inputs for the next layer. The convolutional linear network therefore depends on parameters $\mathbf{h}\in\mathbb{R}^{S\times K}$ and takes the form

$$X=M_1(\mathbf{h}_1)\cdots M_K(\mathbf{h}_K),$$

where the operators M_k satisfy (1.2).

This section aims at identifying conditions such that any unknown parameters $\overline{\mathbf{h}}\in\mathbb{R}^{S\times K}$ can be identified or stably recovered from $X=M_1(\overline{\mathbf{h}}_1)\cdots M_K(\overline{\mathbf{h}}_K)$ (possibly corrupted by an error).

In order to do so, we introduce some notation. We apply the convolutional linear network to an input $x\in\mathbb{R}^{N|\mathcal{F}|}$, where x is the concatenation of the signals $x^f\in\mathbb{R}^N$ for $f\in\mathcal{F}$. Therefore, X is the (horizontal) concatenation of $|\mathcal{F}|$ matrices $X^f\in\mathbb{R}^{N\times N}$ such that

$$Xx=\sum_{f\in\mathcal{F}}X^fx^f\qquad\forall x\in\mathbb{R}^{N|\mathcal{F}|}. \tag{8.1}$$

Consider the convolutional linear network defined by $\mathbf{h}\in\mathbb{R}^{S\times K}$ as well as $f\in\mathcal{F}$ and $n\in[N]$. The column of X corresponding to the entry n in the leaf f is the translation by n of

$$\sum_{p\in\mathcal{P}(f)}\mathcal{T}^p(\mathbf{h})\,, \tag{8.2}$$

where $\mathcal{P}(f)$ contains all the paths of $\mathcal{P}$ starting from the leaf f and

$$\mathcal{T}^p(\mathbf{h})\;=\mathcal{T}_{e^1}(\mathbf{h}_1)*\cdots*\mathcal{T}_{e^K}(\mathbf{h}_K),\qquad\text{with } p=(e^1,\dots,e^K),$$

is the composition of convolutions along the path p.

For any $k\in[K]$, define the mapping $\mathbf{e}_k:[S]\longrightarrow\mathcal{E}(k)$, which provides for any $i\in[S]$ the unique edge of $\mathcal{E}(k)$ such that the ith entry of $h\in\mathbb{R}^S$ contributes to $\mathcal{T}_{\mathbf{e}_k(i)}(h)$. For any $\mathbf{i}\in[S]^K$, let $\mathbf{p}_\mathbf{i}=(\mathbf{e}_1(\mathbf{i}_1),\dots,\mathbf{e}_K(\mathbf{i}_K))$ and

$$\mathbf{I}=\left\{\mathbf{i}\in[S]^K|\mathbf{p}_\mathbf{i}\in\mathcal{P}\right\}.$$

The latter contains all the indices corresponding to a valid path in the network. For any set of parameters $\mathbf{h}\in\mathbb{R}^{S\times K}$ and any path $\mathbf{p}\in\mathcal{P}$, we also let $\mathbf{h}^\mathbf{p}$ denote the restriction of $\mathbf{h}$ to

its indices contributing to the kernels on the path $\mathbf{p}$. We let $\mathbb{1} \in \mathbb{R}^S$ denote a vector of size S with all its entries equal to 1. For any edge e, $\mathbb{1}^e \in \mathbb{R}^S$ consists of zeros except for the entries corresponding to the edge e, which are equal to 1. For any $\mathbf{p} = (e^1, \dots, e^K) \in \mathcal{P}$, the support of $M_1(\mathbb{1}^{e_1}) \cdots M_K(\mathbb{1}^{e_K})$ is denoted by $\mathrm{Supp}\,(\mathbf{p})$.

Finally, by Corollary 5.2 there exists a unique mapping

$$\mathcal{A} : \mathbb{R}^{S^K} \longrightarrow \mathbb{R}^{N \times N|\mathcal{F}|}$$

such that

$$\mathcal{A}P(\mathbf{h}) = M_1(\mathbf{h}_1) \cdots M_K(\mathbf{h}_K) \qquad \forall \mathbf{h} \in \mathbb{R}^{S \times K},$$

where P is the Segre embedding defined in (4.1).

Definition 8.1. *We say the topology of the network is* sufficiently scattered *if and only if all the entries of* $M_1(\mathbb{1}) \cdots M_K(\mathbb{1})$ *belong to* $\{0, 1\}$.

The following statements will show that having a sufficiently scattered topology is a necessary and sufficient condition for the stability of the optimal parameters. Before going into this, we illustrate the scattering property with a simple example.

Consider a simple composition of two convolutions ($K = 2$ and $|\mathcal{P}| = 1$). At first, we make an assumption on the supports $\mathcal{S}_e$, imposing that the supports of both kernels are in $\{1, 2, 3\}$. The topology is obviously not sufficiently scattered. Indeed, some of the entries of the convolution kernel corresponding to the matrix $M_1(\mathbb{1})M_2(\mathbb{1})$ are equal to 2.

Now consider an assumption on the network topology imposing $\{1, 2, 3\}$ for the support of the first kernel and $\{1, 10\}$ for the second. When we observe the convolution of two kernels having such supports, we see two replicas of the first kernel; the amplitudes of the replicas depend on the second kernel, and both kernels are identifiable. In this last example, the network topology is sufficiently scattered.

The scattering condition can easily be computed using Algorithm 8.1. Indeed, when applying the network to a Dirac in the leaf f, using (8.1), we obtain the convolution kernel of X_f. We can then easily test if X_f only contains 0's and 1's. The numerical complexity of Algorithm 8.1 is essentially the cost for applying $|\mathcal{F}|$ times the network. It is usually low. Notice that a network is sufficiently scattered if and only if, for all leaves $f \in \mathcal{F}$, the subnetworks originating at f are sufficiently scattered. The scattering of these subnetworks is independent. The fact that the convolution kernels, for the different leaves, overlap does not affect the scattering property.

Finally, besides the known examples in blind-deconvolution (i.e., when $K = 2$ and $|\mathcal{P}| = 1$) [2, 9], there are (truly deep) convolutional linear networks satisfying the condition of the first statement of Proposition 8.2. For instance, the convolutional linear network corresponding to the undecimated Haar (wavelet)[11] transform is a tree and for any of its leaves $f \in \mathcal{F}$, $|\mathcal{P}(f)| = 1$. Moreover, the support of the kernel living on the edge e, of depth k, on this path is $\{0, 2^k\}$. It is therefore not difficult to check that the first condition of Proposition 8.2 holds.

[11] Undecimated means computed with the "Algorithme à trous" [62, sections 5.5.2 and 6.3.2]. The Haar wavelet is described in [62, section 7.2.2, p. 247, and Example 7.7, p. 235].

Algorithm 8.1. Algorithm testing if the topology of the convolutional network leads to the stability guarantee.

Input: The network topology.
Ouput: Boolean output = "true" if the topology is sufficiently scattered; "false" otherwise.

output = true
For each $f \in \mathcal{F}$ **do**
  Build x: a Dirac positioned at the leaf f
  Apply the network to x in order to compute $y = M_1(\mathbb{1}) \cdots M_K(\mathbb{1})x$
  If some of the entries of y are outside $\{0, 1\}$, then set output = false
end For each

Proposition 8.2 (necessary condition of identifiability of convolutional linear network).

- *Either the topology of the network is sufficiently scattered and then*
  1. *for any distinct* $\mathbf{p}$ *and* $\mathbf{p}' \in \mathcal{P}$, $\mathrm{Supp}\,(\mathbf{p}) \cap \mathrm{Supp}\,(\mathbf{p}') = \emptyset$;
  2. $\mathrm{Ker}\,(\mathcal{A}) = \{T \in \mathbb{R}^{S^K} | \forall \mathbf{i} \in \mathbf{I}, T_{\mathbf{i}} = 0\}$;
- *or the topology of the network is not sufficiently scattered and then* $\mathbb{R}^{S\times K}$ *is not identifiable.*

The proposition is proved in the supplementary material and in [61].

Proposition 8.3. *If* $|\mathcal{P}| = 1$ *and the topology of the network is sufficiently scattered, then* $\mathrm{Ker}\,(\mathcal{A}) = \{0\}$ *and* $\mathrm{Ker}\,(\mathcal{A})$ *satisfies the deep-NSP with respect to any model collection* $\mathcal{M}$ *with constant* $(\gamma, \rho) = (1, +\infty)$. *Moreover, we have* $\sigma_{min} = \sqrt{N}$.

The proposition is proved in the supplementary material and in [61].

In what follows, we establish stability results for a convolutional linear network estimator. In order to do so, we consider a convolutional linear network of known structure $\mathcal{G}(\mathcal{E}, \mathcal{N})$ and $(\mathcal{S}_e)_{e\in\mathcal{E}}$. We consider parameters $\overline{\mathbf{h}} \in \mathbb{R}^{S\times K}$ and $\mathbf{h}^* \in \mathbb{R}^{S\times K}$ such that

$$\|M_1(\overline{\mathbf{h}}_1) \cdots M_K(\overline{\mathbf{h}}_K) - X\| \leq \delta \tag{8.3}$$

and

$$\|M_1(\mathbf{h}_1^*) \cdots M_K(\mathbf{h}_K^*) - X\| \leq \eta. \tag{8.4}$$

We say that two networks sharing the same structure and defined by $\mathbf{h}$ and $\mathbf{g} \in \mathbb{R}^{S\times K}$ are equivalent if and only if

$$\forall \mathbf{p} \in \mathcal{P}, \exists (\lambda_e)_{e\in\mathbf{p}} \in \mathbb{R}^{\mathbf{p}}, \text{ such that } \prod_{e\in\mathbf{p}} \lambda_e = 1 \text{ and } \forall e \in \mathbf{p}, \mathcal{T}_e(\mathbf{g}) = \lambda_e \mathcal{T}_e(\mathbf{h}).$$

The equivalence class of $\mathbf{h} \in \mathbb{R}^{S\times K}$ is denoted by $\{\mathbf{h}\}$. For any $p \in [1, +\infty]$, we define

$$\delta_p(\{\mathbf{h}\}, \{\mathbf{g}\}) = \left(\sum_{\mathbf{p}\in\mathcal{P}} d_p(\langle \mathbf{h}^{\mathbf{p}} \rangle, \langle \mathbf{g}^{\mathbf{p}} \rangle)^p \right)^{\frac{1}{p}},$$

where we recall that $\mathbf{h}^{\mathbf{p}}$ (resp., $\mathbf{g}^{\mathbf{p}}$) denotes the restriction of $\mathbf{h}$ (resp., $\mathbf{g}$) to the path $\mathbf{p}$ and d_p is defined in Definition 4.1. Since d_p is a metric, it follows that δ_p is a metric between network classes.

We summarize the results concerning convolutional networks in the following theorem.

Theorem 8.4 (necessary and sufficient condition of stable recovery of convolutional linear network). *If Algorithm 8.1 returns "false," the network topology is not sufficiently scattered and the network is not identifiable.*

If Algorithm 8.1 returns "true," if $\overline{\mathbf{h}}$ *and* $\mathbf{h}^*$ *satisfy (8.3) and (8.4), and*

- *if all the edges support a significant convolution kernel, there exists* $\varepsilon > 0$ *such that, for all* $e \in \mathcal{E}$, $\|\mathcal{T}_e(\overline{\mathbf{h}})\|_\infty \geq \varepsilon$;
- *if the "signal-to-noise ratio" is sufficient,* $\delta + \eta \leq \frac{\sqrt{N}\varepsilon^K}{2}$,

then the networks defined by $\mathbf{h}^*$ *and* $\overline{\mathbf{h}}$ *are close to each other,*

$$\delta_p(\{\mathbf{h}^*\}, \{\overline{\mathbf{h}}\}) \leq 7(KS')^{\frac{1}{p}}\varepsilon^{1-K}\ \frac{\delta+\eta}{\sqrt{N}},$$

where $S' = \max_{e\in\mathcal{E}} |\mathcal{S}_e|$ *is the size of the largest convolution kernel.*

The theorem is proved in the supplementary material (subsection SM1.14) and in [61].

9. Conclusion and perspectives. In this paper, we have established necessary and sufficient conditions for the identifiability and stable recovery of deep structured linear networks. They rely on the lifting of the problem in a tensor space. The technique is called *tensorial lifting*. The main results are proved using compressed sensing techniques and properties of the Segre embedding (the embedding that maps the parameters in the tensor space). The general results are then specialized to establish necessary and sufficient conditions for the stable recovery of a convolutional linear network of any depth $K \geq 1$.

Among the most salient perspectives, we mention the possibility to study deep feed-forward ReLU networks. For such a network, the action of ReLU is different for every sample; this leads to a different operator $\mathcal{A}$ for every sample; and all the different $\mathcal{A}$'s sense (linearly) the same rank 1 tensor. We can concatenate these operators to form a unique sensing operator. For instance, when modeling the action of ReLU as a Bernouilli variable applied to every path of the network, we expect to obtain sample complexity bounds (for instance) under the favorable hypothesis that an oracle has given us the action of ReLU.

A natural perspective of this work is also to study compressed networks (see [7]) when the compression preserves the expressivity of the network.

Finally, the model considered in this paper approximately solves polynomial equations: $\mathcal{A}P(\mathbf{h}) \sim X$. The structure of the polynomials is induced by the operators M_k (i.e., the network topology) and is very particular and restrictive. For instance, we only consider homogeneous polynomials in $P(\mathbf{h})$. Extending this work to larger families of polynomials as well as limits of polynomials seems natural.

REFERENCES

[1] A. Agarwal, A. Anandkumar, and P. Netrapalli, *A clustering approach to learning sparsely used overcomplete dictionaries*, IEEE Trans. Inform. Theory, 63 (2016), pp. 575–592.

[2] A. Ahmed, B. Recht, and J. Romberg, *Blind deconvolution using convex programming*, IEEE Trans. Inform. Theory, 60 (2014), pp. 1711–1732.
[3] A. Andoni, R. Panigrahy, G. Valiant, and L. Zhang, *Learning polynomials with neural networks*, in International Conference on Machine Learning, 2014, pp. 1908–1916.
[4] S. Arora, A. Bhaskara, R. Ge, and T. Ma, *Provable bounds for learning some deep representations*, in International Conference on Machine Learning, 2014, pp. 584–592.
[5] S. Arora, R. Ge, R. Kannan, and A. Moitra, *Computing a nonnegative matrix factorization–provably*, in Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing, ACM, 2012, pp. 145–162.
[6] S. Arora, R. Ge, and A. Moitra, *New algorithms for learning incoherent and overcomplete dictionaries*, in COLT, 2014, pp. 779–806.
[7] S. Arora, R. Ge, B. Neyshabur, and Y. Zhang, *Stronger generalization bounds for deep nets via a compression approach*, in International Conference on Machine Learning, 2018, pp. 254–263.
[8] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, *On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation*, PloS One, 10 (2015), e0130140.
[9] S. Bahmani and J. Romberg, *Lifting for blind deconvolution in random mask imaging: Identifiability and convex relaxation*, SIAM J. Imaging Sci., 8 (2015), pp. 2203–2238, https://doi.org/10.1137/141002165.
[10] P. Baldi and K. Hornik, *Neural networks and principal component analysis: Learning from examples without local minima*, Neural Networks, 2 (1989), pp. 53–58.
[11] P. F. Baldi and K. Hornik, *Learning in linear neural networks: A survey*, IEEE Trans. Neural Networks, 6 (1995), pp. 837–858.
[12] A. R. Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Trans. Inform. Theory, 39 (1993), pp. 930–945.
[13] A. Bourrier, M. Davies, T. Peleg, P. Pérez, and R. Gribonval, *Fundamental performance limits for ideal decoders in high-dimensional linear inverse problems*, IEEE Trans. Inform. Theory, 60 (2014), pp. 7928–7946.
[14] A. Brutzkus and A. Globerson, *Globally optimal gradient descent for a ConvNet with Gaussian inputs*, in Proceedings of the 34 th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017, pp. 605–614.
[15] E. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, 52 (2006), pp. 489–509.
[16] E. J. Candès, Y. C. Eldar, T. Strohmer, and V. Voroninski, *Phase retrieval via matrix completion*, SIAM Rev., 57 (2015), pp. 225–251, https://doi.org/10.1137/151005099.
[17] E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717–772.
[18] E. J. Candès, T. Strohmer, and V. Voroninski, *Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming*, Comm. Pure Appl. Math., 66 (2013), pp. 1241–1274.
[19] E. J. Candès and T. Tao, *The power of convex relaxation: Near-optimal matrix completion*, IEEE Trans. Inform. Theory, 56 (2010), pp. 2053–2080.
[20] O. Chabiron, F. Malgouyres, J.-Y. Tourneret, and N. Dobigeon, *Toward fast transform learning*, Int. J. Comput. Vis., (2014), pp. 1–22.
[21] O. Chabiron, F. Malgouyres, H. Wendt, and J.-Y. Tourneret, *Optimization of a Fast Transform Structured as a Convolutional Tree*, preprint hal-01258514, 2016.
[22] A. Choromanska, M. Henaff, M. Mathieu, G. Ben Arous, and Y. LeCun, *The loss surfaces of multilayer networks*, in Artificial Intelligence and Statistics, 2015, pp. 192–204.
[23] A. Choromanska, Y. LeCun, and G. Ben Arous, *Open problem: The landscape of the loss surfaces of multilayer networks*, in Conference on Learning Theory, 2015, pp. 1756–1760.
[24] S. Choudhary and U. Mitra, *Identifiability Scaling Laws in Bilinear Inverse Problems*, preprint, https://arxiv.org/abs/1402.2637, 2014.
[25] A. Cohen, W. Dahmen, and R. DeVore, *Compressed sensing and best-term approximation*, J. Amer. Math. Soc., 22 (2009), pp. 211–231.

[26] N. Cohen, O. Sharir, and A. Shashua, *On the expressive power of deep learning: A tensor analysis*, in Conference on Learning Theory, 2016, pp. 698–728.
[27] N. Cohen and A. Shashua, *Convolutional rectifier networks as generalized tensor decompositions*, in International Conference on Machine Learning, 2016, pp. 955–963.
[28] C. Davis and W. M. Kahan, *The rotation of eigenvectors by a perturbation.* III, SIAM J. Numer. Anal., 7 (1970), pp. 1–46, https://doi.org/10.1137/0707001.
[29] D. Donoho and V. Stodden, *When does non-negative matrix factorization give a correct decomposition into parts?*, in Advances in Neural Information Processing Systems, 2004, pp. 1141–1148.
[30] D. L. Donoho, *Compressed sensing*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1289–1306.
[31] R. Eldan and O. Shamir, *The power of depth for feedforward neural networks*, in Conference on Learning Theory, 2016, pp. 907–940.
[32] M. Fazel, E. Candes, B. Recht, and P. Parrilo, *Compressed sensing and robust recovery of low rank matrices*, in 42nd Asilomar Conference on Signals, Systems and Computers, IEEE, 2008, pp. 1043–1047.
[33] Q. Geng, H. Wangy, and J. Wright, *On the local correctness of ℓ_1-minimization for dictionary learning*, in International Symposium on Information Theory (ISIT), 2014.
[34] S. Goel and A. Klivans, *Eigenvalue decay implies polynomial-time learnability for neural networks*, in Advances in Neural Information Processing Systems, 2017, pp. 2192–2202.
[35] R. Gribonval, R. Jenatton, and F. Bach, *Sample complexity of dictionary learning and other matrix factorizations*, IEEE Trans. Inform. Theory, 61 (2015), pp. 3469–3486.
[36] R. Gribonval and K. Schnass, *Dictionary identification—sparse matrix-factorisation via ℓ_1-minimisation*, IEEE Trans. Inform. Theory, 56 (2010), pp. 3523–3539.
[37] B. D. Haeffele and R. Vidal, *Global Optimality in Tensor Factorization, Deep Learning, and Beyond*, preprint, https://arxiv.org/abs/1506.07540, 2015.
[38] J. Harris, *Algebraic Geometry. A First Course*, Grad. Texts in Math. 133, Springer-Verlag, New York, 1995; corrected reprint of the 1992 original.
[39] J. D. Hauenstein and A. J. Sommese, *Witness sets of projections*, Appl. Math. Comput., 217 (2010), pp. 3349–3354.
[40] J. D. Hauenstein and A. J. Sommese, *Membership tests for images of algebraic sets by linear projections*, Appl. Math. Comput., 219 (2013), pp. 6809–6818.
[41] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, *Generating visual explanations*, in European Conference on Computer Vision, Springer, 2016, pp. 3–19.
[42] M. Janzamin, H. Sedghi, and A. Anandkumar, *Beating the Perils of Non-convexity: Guaranteed Training of Neural Networks Using Tensor Methods*, preprint, https://arxiv.org/abs/1506.08473, 2015.
[43] R. Jenatton, R. Gribonval, and F. Bach, *Local Stability and Robustness of Sparse Dictionary Learning in the Presence of Noise*, preprint, https://arxiv.org/abs/1210.0685, 2012.
[44] K. Kawaguchi, *Deep learning without poor local minima*, in Advances in Neural Information Processing Systems, 2016, pp. 586–594.
[45] V. Khrulkov, A. Novikov, and I. Oseledets, *Expressive Power of Recurrent Neural Networks*, preprint, https://arxiv.org/abs/1711.00811, 2017.
[46] R. Kondor, N. Teneva, and V. Garg, *Multiresolution matrix factorization*, in Proceedings of the 31st International Conference on Machine Learning (ICML-14), 2014, pp. 1620–1628.
[47] J. M. Landsberg, *Tensors: Geometry and Applications*, Graduate Stud. Math. 128, AMS, 2012.
[48] H. Laurberg, M. G. Christensen, M. D. Plumbley, L. K. Hansen, and S. H. Jensen, *Theorems on positive data: On the uniqueness of NMF*, Comput. Intell. Neurosci., 2008 (2008), 764206.
[49] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, *Speeding-Up Convolutional Neural Networks Using Fine-Tuned CP-Decomposition*, preprint, https://arxiv.org/abs/1412.6553, 2014.
[50] D. D. Lee and H. S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.
[51] X. Li, S. Ling, T. Strohmer, and K. Wei, *Rapid, robust, and reliable blind deconvolution via nonconvex optimization*, Appl. Comput. Harmon. Anal., 2018.
[52] X. Li and V. Voroninski, *Sparse signal recovery from quadratic measurements via convex programming*,

SIAM J. Math. Anal., 45 (2013), pp. 3019–3033, https://doi.org/10.1137/120893707.
[53] Y. Li and Y. Yuan, *Convergence analysis of two-layer neural networks with ReLU activation*, in Advances in Neural Information Processing Systems, 2017, pp. 597–607.
[54] S. Ling and T. Strohmer, *Self-calibration and biconvex compressive sensing*, Inverse Problems, 31 (2015), 115002.
[55] R. Livni, S. Shalev-Shwartz, and O. Shamir, *On the computational efficiency of training neural networks*, in Advances in Neural Information Processing Systems, 2014, pp. 855–863.
[56] S. Lyu and X. Wang, *On algorithms for sparse multi-factor NMF*, in Proceedings of the 26th International Conference on Neural Information Processing Systems, NIPS'13, Curran Associates, 2013, pp. 602–610.
[57] L. Le Magoarou, *Matrices efficientes pour le traitement du signal et l'apprentissage automatique*, Ph.D. thesis, Université Bretagne Loire, 2016.
[58] L. Le Magoarou and R. Gribonval, *Are there approximate fast Fourier transforms on graphs?*, in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 4811–4815.
[59] L. Le Magoarou and R. Gribonval, *Flexible multi-layer sparse approximations of matrices and applications*, IEEE J. Selected Topics Signal Process., 10 (2016), pp. 688–700.
[60] F. Malgouyres and J. Landsberg, *On the identifiability and stable recovery of deep/multi-layer structured matrix factorization*, in 2016 IEEE Information Theory Workshop, 2016, pp. 315–319.
[61] F. Malgouyres and J. Landsberg, *Multilinear Compressive Sensing and an Application to Convolutional Linear Networks*, preprint hal-01494267, 2017.
[62] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, Boston, 1998.
[63] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, *Explaining nonlinear classification decisions with deep Taylor decomposition*, Pattern Recog., 65 (2017), pp. 211–222.
[64] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, *On the number of linear regions of deep neural networks*, in Advances in Neural Information Processing Systems, 2014, pp. 2924–2932.
[65] B. Neyshabur and R. Panigrahy, *Sparse Matrix Factorization*, preprint, https://arxiv.org/abs/1311.3315, 2013.
[66] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov, *Tensorizing neural networks*, in Advances in Neural Information Processing Systems, 2015, pp. 442–450.
[67] B. Recht, M. Fazel, and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Rev., 52 (2010), pp. 471–501, https://doi.org/10.1137/070697835.
[68] R. Rubinstein, A. Bruckstein, and M. Elad, *Dictionaries for sparse representation modeling*, Proc. IEEE - Special issue on applications of sparse representation and compressive sensing, 98 (2010), pp. 1045–1057.
[69] I. Safran and O. Shamir, *On the quality of the initial basin in overspecified neural networks*, in International Conference on Machine Learning, 2016, pp. 774–782.
[70] J. Schmidt-Hieber, *Nonparametric regression using deep neural networks with ReLU activation function*, Ann. Statist., to appear.
[71] K. Schnass, *On the identifiability of overcomplete dictionaries via the minimisation principle underlying K-SVD*, Appl. Comput. Harmon. Anal., 37 (2014), pp. 464–491.
[72] H. Sedghi and A. Anandkumar, *Provable methods for training neural networks with sparse connectivity*, in Deep Learning and Representation Learning Workshop: NIPS, 2014.
[73] I. R. Shafarevich, *Basic Algebraic Geometry. 1. Varieties in Projective Space*, 3rd ed., Springer, Heidelberg, 2013.
[74] R. Socher, D. Chen, C. D. Manning, and A. Ng, *Reasoning with neural tensor networks for knowledge base completion*, in Advances in Neural Information Processing Systems, 2013, pp. 926–934.
[75] D. Spielmana, H. Wang, and J. Wright, *Exact recovery of sparsely-used dictionaries*, in COLT, 2012, 37–1.
[76] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *Dropout: A simple way to prevent neural networks from overfitting*, J. Mach. Learn. Res., 15 (2014), pp. 1929–1958.

[77] M. Telgarsky, *Benefits of depth in neural networks*, in Conference on Learning Theory, 2016, pp. 1517–1539.

[78] T. Tsiligkaridis, A. O. Hero, and S. Zhou, *On convergence of Kronecker graphical lasso algorithms*, IEEE Trans. Signal Process., 61 (2013), pp. 1743–1755.

[79] S. A. Vavasis, *On the complexity of nonnegative matrix factorization*, SIAM J. Optim., 20 (2009), pp. 1364–1377, https://doi.org/10.1137/070709967.

[80] L. Venturi, A. S. Bandeira, and J. Bruna, *Spurious Valleys in Two-Layer Neural Network Optimization Landscapes*, preprint, https://arxiv.org/abs/1802.06384, 2018.

[81] B. Xie, Y. Liang, and L. Song, *Diverse neural network learns true target functions*, in Artificial Intelligence and Statistics, 2017, pp. 1216–1224.

[82] D. Yu, L. Deng, and F. Seide, *The deep tensor neural network with applications to large vocabulary speech recognition*, IEEE Trans. Audio, Speech, Language Process., 21 (2013), pp. 388–396.

[83] C. Yunpeng, J. Xiaojie, K. Bingyi, F. Jiashi, and Y. Shuicheng, *Sharing Residual Units through Collective Tensor Factorization in Deep Neural Networks*, preprint, https://arxiv.org/abs/1703.02180, 2017.

[84] K. Zhong, Z. Song, P. Jain, P. L. Bartlett, and I. S. Dhillon, *Recovery guarantees for one-hidden-layer neural networks*, in Proceedings of the 34th International Conference on Machine Learning, D. Precup and Y. W. Teh, eds., Proc. Mach. Learn. Res. 70, Sydney, Australia, PMLR, 2017, pp. 4140–4149.