The difficulty of computing stable and accurate neural networks: On the barriers of deep learning and Smale’s 18th problem

Matthew J. Colbrook[a,1,2], **Vegard Antun**[b,1,2], **and Anders C. Hansen**[a,b,2]

[a]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge CB3 0WA, United Kingdom; and [b]Department of Mathematics, University of Oslo, 0316 Oslo, Norway

Edited by Ronald DeVore, Texas A&M University, College Station, TX; received April 16, 2021; accepted October 26, 2021

Deep learning (DL) has had unprecedented success and is now entering scientific computing with full force. However, current DL methods typically suffer from instability, even when universal approximation properties guarantee the existence of stable neural networks (NNs). We address this paradox by demonstrating basic well-conditioned problems in scientific computing where one can prove the existence of NNs with great approximation qualities; however, there does not exist any algorithm, even randomized, that can train (or compute) such a NN. For any positive integers $K > 2$ and L, there are cases where simultaneously 1) no randomized training algorithm can compute a NN correct to K digits with probability greater than 1/2; 2) there exists a deterministic training algorithm that computes a NN with $K - 1$ correct digits, but any such (even randomized) algorithm needs arbitrarily many training data; and 3) there exists a deterministic training algorithm that computes a NN with $K - 2$ correct digits using no more than L training samples. These results imply a classification theory describing conditions under which (stable) NNs with a given accuracy can be computed by an algorithm. We begin this theory by establishing sufficient conditions for the existence of algorithms that compute stable NNs in inverse problems. We introduce fast iterative restarted networks (FIRENETs), which we both prove and numerically verify are stable. Moreover, we prove that only $\mathcal{O}(|\log(\epsilon)|)$ layers are needed for an ϵ-accurate solution to the inverse problem.

stability and accuracy | AI and deep learning | inverse problems | Smale’s 18th problem | solvability complexity index hierarchy

Deep learning (DL) has demonstrated unparalleled accomplishments in fields ranging from image classification and computer vision (1–3), to voice recognition and automated diagnosis in medicine (4–6), to inverse problems and image reconstruction (7–12). However, there is now overwhelming empirical evidence that current DL techniques typically lead to unstable methods, a phenomenon that seems universal and present in all of the applications listed above (13–21) and in most of the new artificial intelligence (AI) technologies. These instabilities are often detected by what has become commonly known in the literature as “adversarial attacks.” Moreover, the instabilities can be present even in random cases and not just worst-case scenarios (22)—see Fig. 1 for an example of AI-generated hallucinations. There is a growing awareness of this problem in high-stakes applications and society as a whole (20, 23, 24), and instability seems to be the Achilles’ heel of modern AI and DL (Fig. 2, *Top* row). For example, this is a problem in real-world clinical practice. Facebook and New York University’s 2019 FastMRI challenge reported that networks that performed well in terms of standard image quality metrics were prone to false negatives, failing to reconstruct small, but physically relevant image abnormalities (25). Subsequently, the 2020 FastMRI challenge (26) focused on pathologies, noting, “Such hallucinatory features are not acceptable and especially problematic if they mimic normal structures that are either not present or actually abnormal. Neural network models can be unstable as demonstrated via adversarial perturbation studies (19).” For similar examples in microscopy, see refs. 27 and 28. The tolerance level for false positives/negatives varies within different applications. However, for scenarios with a high cost of misanalysis, it is imperative that false negatives/positives be avoided. AI-generated hallucinations therefore pose a serious danger in applications such as medical diagnosis.

Nevertheless, classical approximation theorems show that a continuous function can be approximated arbitrarily well by a neural network (NN) (29, 30). Thus, stable problems described by stable functions can always be solved stably with a NN. This leads to the following fundamental question:

Question. *Why does DL lead to unstable methods and AI-generated hallucinations, even in scenarios where one can prove that stable and accurate neural networks exist?*

Foundations of AI for Inverse Problems. To answer the above question we initiate a program on the foundations of AI, determining the limits of what DL can achieve in inverse problems. It is crucial to realize that an existence proof of suitable NNs does not always imply that they can be constructed by a training algorithm.

Significance

Instability is the Achilles’ heel of modern artificial intelligence (AI) and a paradox, with training algorithms finding unstable neural networks (NNs) despite the existence of stable ones. This foundational issue relates to Smale’s 18th mathematical problem for the 21st century on the limits of AI. By expanding methodologies initiated by Gödel and Turing, we demonstrate limitations on the existence of (even randomized) algorithms for computing NNs. Despite numerous existence results of NNs with great approximation properties, only in specific cases do there also exist algorithms that can compute them. We initiate a classification theory on which NNs can be trained and introduce NNs that—under suitable conditions—are robust to perturbations and exponentially accurate in the number of hidden layers.

Author contributions: M.J.C., V.A., and A.C.H. designed research, performed research, and wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1] M.J.C. and V.A. contributed equally to this work.

[2] To whom correspondence may be addressed. Email: m.colbrook@damtp.cam.ac.uk, vegarant@math.uio.no, or a.hansen@damtp.cam.ac.uk.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2107151119/-/DCSupplemental.

Published March 16, 2022.

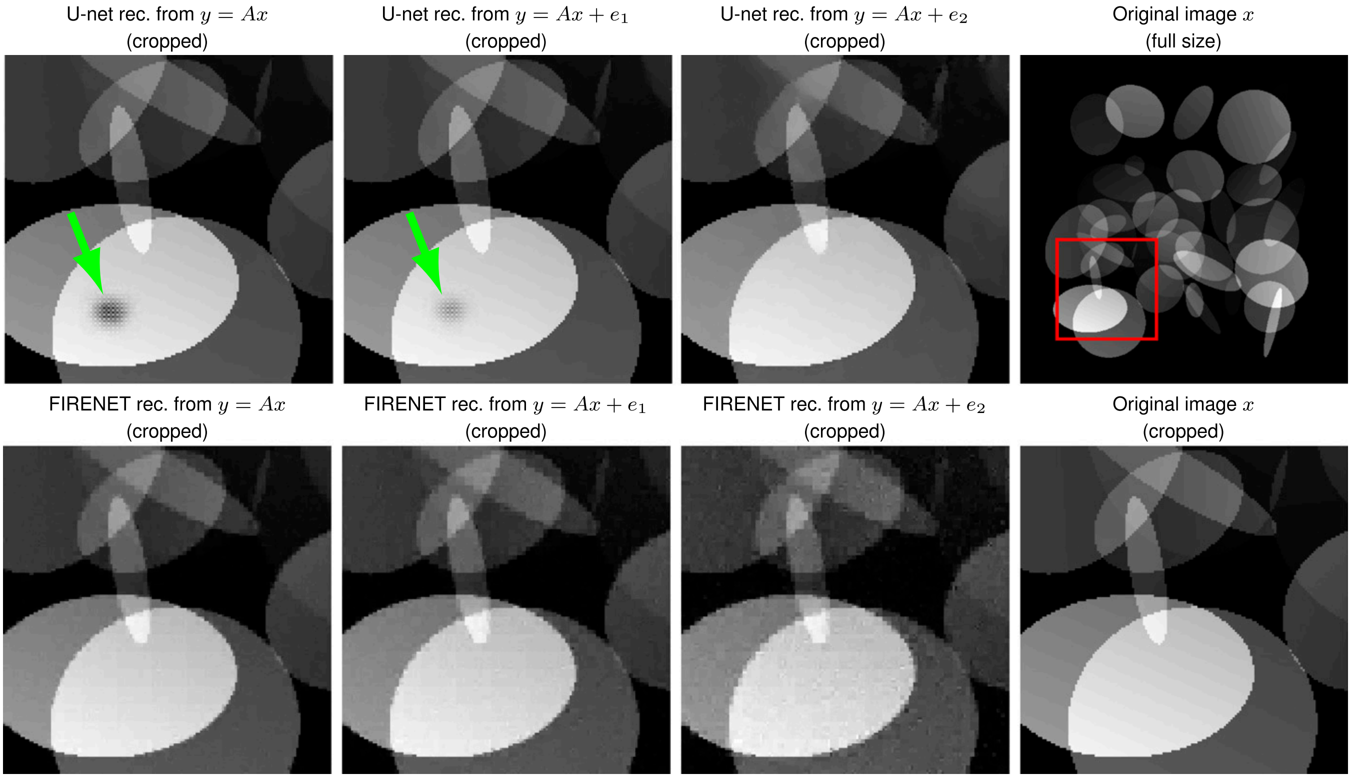

Fig. 1. AI-generated hallucinations. A trained NN, based on a U-net architecture and trained on a set of ellipses images, generates a black area in a white ellipse (*Top Left* image, shown as green arrow) when reconstructing the original image *x* from noiseless measurements. By adding random Gaussian noise e_1 and e_2 (where $\|e_1\|_{l^2}/\|e_2\|_{l^2} \approx 2/5$) to the measurements, we see that the trained NN removes the aspiring black ellipse (*Top* row, *Center Left* to *Center Right*). FIRENET on the other hand is completely stable with and without random Gaussian noise (*Bottom* row, *Left* to *Center Right*). In *Right* column, we show the original image *x*, with a red square (*Top Right*) indicating the cropped area. In this example, $A \in \mathbb{C}^{m\times N}$ is a subsampled discrete Fourier transform with $m/N \approx 0.12$.

Furthermore, it is not difficult to compute stable NNs. For example, the zero network is stable, but not particularly useful. The big problem is to compute NNs that are both stable and accurate (30, 31). Scientific computing itself is based on the pillars of stability and accuracy. However, there is often a trade-off between the two. There may be barriers preventing the existence of stable and accurate algorithms, and sometimes accuracy must be sacrificed to secure stability.

Main Results. We consider the canonical inverse problem of an underdetermined system of linear equations:

$$\text{Given measurements } y = Ax + e \in \mathbb{C}^m, \text{recover } x \in \mathbb{C}^N. \quad [1]$$

Here, $A \in \mathbb{C}^{m\times N}$ represents a sampling model ($m < N$), such as a subsampled discrete Fourier transform in MRI, and *x* the unknown quantity. The problem in Eq. **1** forms the basis for much of inverse problems and image analysis. The vector *e* models noise or perturbations. Our results demonstrate fundamental barriers preventing NNs (despite their existence) from being computed by algorithms. This helps shed light on the intricate question of why current algorithms in DL produce unstable networks, despite the fact that stable NNs often exist in the particular application. We show the following:

1) *Theorems 1* and *2*: There are well-conditioned problems (suitable condition numbers bounded by 1) where, paradoxically, mappings from training data to suitable NNs exist, but no training algorithm (even randomized) can compute approximations of the NNs from the training data.
2) *Theorem 2*: The existence of algorithms computing NNs depends on the desired accuracy. For any $K \in \mathbb{Z}_{\geq 3}$, there are well-conditioned classes of problems where simultaneously 1) algorithms may compute NNs to $K-1$ digits of accuracy, but not K; 2) achieving $K-1$ digits of accuracy requires arbitrarily many training data; and 3) achieving $K-2$ correct digits requires only one training datum.
3) *Theorems 3* and *4*: Under specific conditions that are typically present in, for example, MRI, there are algorithms that compute stable NNs for the problem in Eq. **1**. These NNs, which we call fast iterative restarted networks (FIRENETs), converge exponentially in the number of hidden layers. Crucially, we prove that FIRENETs are robust to perturbations (Fig. 2, *Bottom* row), and they can even be used to stabilize unstable NNs (Fig. 3).
4) There is a trade-off between stability and accuracy in DL, with limits on how well a stable NN can perform in inverse problems. Fig. 4 demonstrates this with a U-net trained on images consisting of ellipses that is quite stable. However, when a detail not in the training set is added, it washes it out almost entirely. FIRENETs offer a blend of both stability and accuracy. However, they are by no means the end of the story. Tracing out the optimal stability vs. accuracy trade-off is crucial for applications and will no doubt require a myriad of different techniques to tackle different problems and stability tolerances.

Fundamental Barriers

We first consider basic mappings used in modern mathematics of information, inverse problems, and optimization. Given a matrix $A \in \mathbb{C}^{m\times N}$ and a vector $y \in \mathbb{C}^m$, we consider the following three popular minimization problems:

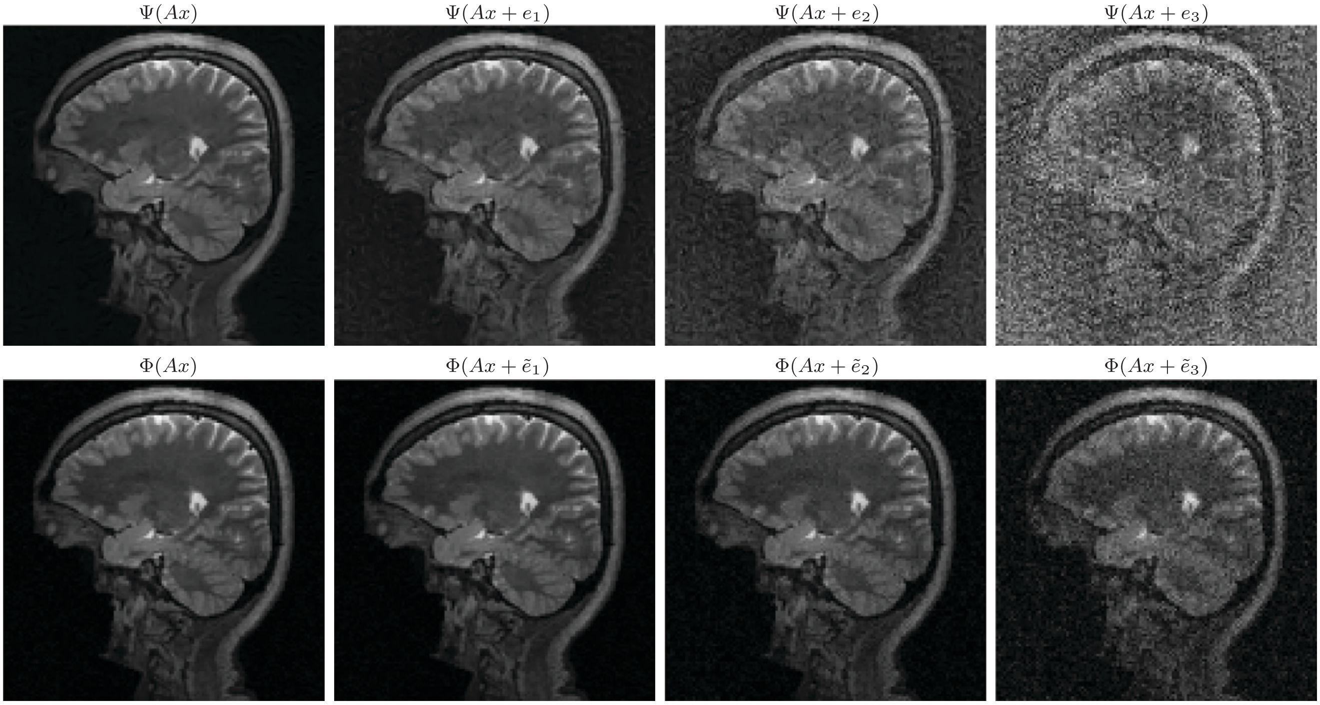

Fig. 2. *Top* row (unstable neural network in image reconstruction): The neural network AUTOMAP (60) represents the tip of the iceberg of DL in inverse problems. Ref. 60, pp. 1 and 487, promises that one can "…observe superior immunity to noise…." Moreover, the follow-up announcement (ref. 83, pp. 1 and 309) proclaims "A deep-learning-based approach improves speed, accuracy and robustness of biomedical image reconstruction." However, as we see in *Top* row, the AUTOMAP reconstruction $\Psi(Ax + e_j)$ from the subsampled noisy Fourier MRI data $Ax + e_j$ is completely unstable. Here, $A \in \mathbb{C}^{m\times N}$ is a subsampled Fourier transform, x is the original image, and the e_j s are perturbations meant to simulate the worst-case effect. Note that the condition number $\mathrm{cond}(AA^*) = 1$, so the instabilities are not caused by poor condition. The network weights were provided by the authors of ref. 60, which trained and tested it on brain images from the Massachusetts General Hospital Human Connectome Project (MGH-USC HCP) dataset (84). The image x is taken from this dataset. *Bottom* row (the FIRENET is stable to worst-case perturbations): Using the same method, we compute perturbations $\tilde{e}_j$ to simulate the worst-case effect for the FIRENET $\Phi\colon \mathbb{C}^m \to \mathbb{C}^N$. As can be seen, FIRENET is stable to these worst-case perturbations. Here x and $A \in \mathbb{C}^{m\times N}$ are the same image and sampling matrix as for AUTOMAP. Moreover, for each $j = 1, 2, 3$ we have ensured that $\|\tilde{e}_j\|_{l^2} \geq \|e_j\|_{l^2}$, where the e_j s are the perturbations for AUTOMAP (we have denoted the perturbations for FIRENET by $\tilde{e}_j$ to emphasize that these adversarial perturbations are sought for FIRENET and have nothing to do with the perturbations for AUTOMAP).

$$(P_1) \quad \mathrm{argmin}_{x\in\mathbb{C}^N}\, F_1^A(x) := \|x\|_{l^1_w}, \text{s.t.} \|Ax - y\|_{l^2} \leq \epsilon,$$

$$(P_2) \quad \mathrm{argmin}_{x\in\mathbb{C}^N}\, F_2^A(x, y, \lambda) := \lambda\|x\|_{l^1_w} + \|Ax - y\|^2_{l^2},$$

$$(P_3) \quad \mathrm{argmin}_{x\in\mathbb{C}^N}\, F_3^A(x, y, \lambda) := \lambda\|x\|_{l^1_w} + \|Ax - y\|_{l^2},$$

known respectively as quadratically constrained basis pursuit [we always assume existence of a feasible x for (P_1)], unconstrained least absolute shrinkage and selection operator (LASSO), and unconstrained square-root LASSO. Such sparse regularization problems are often used as benchmarks for Eq. **1**, and we prove impossibility results for computing the NNs that can approximate these mappings. Our results initiate a classification theory on which NNs can be computed to a certain accuracy.

The parameters λ and ϵ are positive rational numbers, and the weighted l^1_w norm is given by $\|x\|_{l^1_w} := \sum_{l=1}^N w_l|x_l|$, where each weight w_j is a positive rational. Throughout, we let

$$\Xi_j(A, y) \text{ be the set of minimizers for } (P_j). \qquad \textbf{[2]}$$

Let $A \in \mathbb{C}^{m\times N}$ and let $\mathcal{S} = \{y_k\}_{k=1}^R \subset \mathbb{C}^m$ be a collection of samples ($R \in \mathbb{N}$). We consider the following key question:

Question. *Given a collection Ω of pairs $(A, \mathcal{S})$, does there exist a neural network approximating Ξ_j, and if so, can such an approximation be trained or determined by an algorithm?*

To make this question precise, note that A and samples in $\mathcal{S}$ will typically never be exact, but can be approximated/stored to arbitrary precision. For example, this would occur if A was a subsampled discrete cosine transform. Thus, we assume access to rational approximations $\{y_{k,n}\}_{k=1}^R$ and A_n with

$$\|y_{k,n} - y_k\|_{l^2} \leq 2^{-n}, \quad \|A_n - A\| \leq 2^{-n}, \quad \forall n \in \mathbb{N}, \qquad \textbf{[3]}$$

where $\|\cdot\|$ refers to the usual Euclidean operator norm. The bounds 2^{-n} are simply for convenience and can be replaced by any other sequence converging to zero. We also assume access to rational $\{x_{k,n}\}_{k=1}^R$ with

$$\inf_{x^*\in\Xi_j(A_n, y_{k,n})} \|x_{k,n} - x^*\|_{l^2} \leq 2^{-n}, \quad \forall n \in \mathbb{N}. \qquad \textbf{[4]}$$

Hence, the training data associated with $(A, \mathcal{S}) \in \Omega$ must be

$$\iota_{A,\mathcal{S}} := \{(y_{k,n}, A_n, x_{k,n}) \mid k = 1, \ldots, R, \text{and } n \in \mathbb{N}\}. \qquad \textbf{[5]}$$

This set is formed of arbitrary precision rational approximations of finite collections of data associated with $(A, \mathcal{S})$. Given a collection Ω of pairs $(A, \mathcal{S})$, the class of all such admissible training data is denoted by

$$\Omega_{\mathcal{T}} := \{\iota_{A,\mathcal{S}} \text{ as in Eq. 5} \mid (A, \mathcal{S}) \in \Omega, \text{Eqs. 3 to 4 hold}\}.$$

Statements addressing the above question are summarized in *Theorems 1* and *2*. We use $\mathcal{N}_{m,N}$ to denote the class of NNs from $\mathbb{C}^m$ to $\mathbb{C}^N$. We use standard definitions of feedforward NNs (32), precisely given in *SI Appendix*.

Theorem 1. *For any collection Ω of such $(A, \mathcal{S})$ described above, there exists a mapping*

$$\mathcal{K}\colon \Omega_{\mathcal{T}} \to \mathcal{N}_{m,N}, \quad \mathcal{K}(\iota_{A,\mathcal{S}}) = \varphi_{A,\mathcal{S}},$$
$$s.t. \quad \varphi_{A,\mathcal{S}}(y) \in \Xi_j(A, y), \quad \forall y \in \mathcal{S}.$$

In words, $\mathcal{K}$ maps the training data $\Omega_{\mathcal{T}}$ to NNs that solve the optimization problem (P_j) for each $(A, \mathcal{S}) \in \Omega$.

Despite the existence of NNs guaranteed by *Theorem 1*, computing or training such a NN from training data is most delicate. The following is stated precisely and proved in *SI Appendix*. We also include results for randomized algorithms, which are common in DL (e.g., stochastic gradient descent).

Theorem 2. *Consider the optimization problem (P_j) for fixed parameters $\lambda \in (0, 1]$ or $\epsilon \in (0, 1/2]$ and $w_l = 1$, where $N \geq 2$ and $m < N$. Let $K > 2$ be a positive integer and let $L \in \mathbb{N}$. Then there exists an infinite class $\Omega = \Omega(K, L)$ of elements $(A, \mathcal{S})$ as above, with the following properties. The class Ω is well-conditioned with relevant condition numbers bounded by 1 independent of all parameters. However, the following hold simultaneously* (*where accuracy is measured in the l^2 norm*):

1) (*K digits of accuracy impossible*) *There does not exist any algorithm that, given a training set $\iota_{A,\mathcal{S}} \in \Omega_{\mathcal{T}}$, produces a NN with K digits of accuracy for any element of $\mathcal{S}$. Furthermore, for any $p > 1/2$, no probabilistic algorithm* (Blum–Shub–Smale [*BSS*], *Turing, or any model of computation*) *can produce a NN with K digits of accuracy with probability at least p.*
2) (*$K - 1$ digits of accuracy possible but requires arbitrarily many training data*) *There does exist a deterministic Turing machine that, given a training set $\iota_{A,\mathcal{S}} \in \Omega_{\mathcal{T}}$, produces a NN accurate to $K - 1$ digits over $\mathcal{S}$. However, for any probabilistic Turing machine, $M \in \mathbb{N}$ and $p \in \left[0, \frac{N-m}{N+1-m}\right)$ that produces a NN, there exists a training set $\iota_{A,\mathcal{S}} \in \Omega_{\mathcal{T}}$ such that for all $y \in \mathcal{S}$, the probability of failing to achieve $K - 1$ digits or requiring more than M training data is greater than p.*
3) (*$K - 2$ digits of accuracy possible with L training data*) *There does exist a deterministic Turing machine that, given a training set $\iota_{A,\mathcal{S}} \in \Omega_{\mathcal{T}}$ and using only L training data from each $\iota_{A,\mathcal{S}}$, produces a NN accurate to $K - 2$ digits over $\mathcal{S}$.*

Remark 1 (condition and class size). The statement in Theorem 2 refers to the standard condition numbers used in optimization and scientific computing. For precise definitions, see *SI Appendix*. The class Ω we construct is infinite. Similarly, one can design a finite class Ω with the same conclusion by allowing the sample size *R* to be infinite.

Remark 2 (distributions on training data). In DL it is often the case that one assumes some probability distribution on the training data. This is not needed for Theorem 2. However, having a probability distribution on the training data $\iota_{A,\mathcal{S}}$ would not invalidate statement 1 in Theorem 2. In particular, there is no (computable) probability distribution that would make statement 1 in Theorem 2 cease to be true. This follows from the probabilistic part of statement 1 in Theorem 2, as the existence of such a (computable) distribution and an algorithm would yield a randomized algorithm violating statement 1 in Theorem 2.

Remark 3 (on the role of K in Theorem 2). The result should be understood as fixing an integer *K* (and *L*) and then $\Omega = \Omega(K, L)$ depends on *K* and *L*. However, given a particular Ω one can ask, what is the largest *K* such that one can compute *K* correct digits? Note that we typically have $K = \lfloor \log(\epsilon^{-1}) \rfloor$, where $\epsilon > 0$ is the so-called breakdown epsilon of the problem (33), i.e., the largest $\epsilon > 0$ for which all algorithms will fail to provide ϵ accuracy. When the breakdown epsilon $\epsilon > 0$, it is typically impossible to check whether an algorithm fails (33). Thus, even if an algorithm would succeed with probability $1/2$, one could never trust the output.

Table 1. Impossibility of computing approximations of the existing neural network to arbitrary accuracy

Ψ_{A_n}	Φ_{A_n}	$\|A_n - A\| \leq 2^{-n}$	10^{-K}	$\Omega(K)$
		$\|y_n - y\|_{l^2} \leq 2^{-n}$		
0.2999690	0.2597827	$n = 10$	10^{-1}	$K = 1$
0.3000000	0.2598050	$n = 20$	10^{-1}	$K = 1$
0.3000000	0.2598052	$n = 30$	10^{-1}	$K = 1$
0.0030000	0.0025980	$n = 10$	10^{-3}	$K = 3$
0.0030000	0.0025980	$n = 20$	10^{-3}	$K = 3$
0.0030000	0.0025980	$n = 30$	10^{-3}	$K = 3$
0.0000030	0.0000015	$n = 10$	10^{-6}	$K = 6$
0.0000030	0.0000015	$n = 20$	10^{-6}	$K = 6$
0.0000030	0.0000015	$n = 30$	10^{-6}	$K = 6$

We demonstrate statement 1 from Theorem 2 on FIRENETs Φ_{A_n} and LISTA networks Ψ_{A_n}. Shown is the shortest l^2 distance between the output from the networks and the true solution of the problem (P_3), with $w_l = 1$ and $\lambda = 1$, for different values of n and K. Note that none of the networks can compute the existing correct NN (that exists by Theorem 1 and coincides with Ξ_3) to 10^{-K} digits accuracy, while all of them are able to compute approximations that are accurate to 10^{-K+1} digits [for the input class $\Omega(K)$]. This agrees exactly with Theorem 2.

Remark 4 (Gödel, Turing, Smale, and Theorem 2). Theorems 1 and 2 demonstrate basic limitations on the existence of algorithms that can compute NNs despite their existence. This relates to Smale's 18th problem, "What are the limits of intelligence, both artificial and human?", from the list of mathematical problems for the 21st century (34), which echoes the Turing test from 1950 (35). Smale's discussion is motivated by the results of Gödel (36) and Turing (37) establishing impossibility results on what mathematics and digital computers can achieve (38). Our results are actually stronger, however, than what can be obtained with Turing's techniques. Theorem 2 holds even for any randomized Turing or BSS machine that can solve the halting problem. It immediately opens up for a classification theory on which NNs can be computed by randomized algorithms. Theorem 3 is a first step in this direction. See also the work by Niyogi, Smale, and Weinberger (39) on existence results of algorithms for learning.

Numerical Example. To highlight the impossibility of computing NNs (Theorem 2)—despite their existence by Theorem 1—we consider the following numerical example: Consider the problem (P_3), with $w_l = 1$ and $\lambda = 1$. Theorem 2 is stated for a specific input class $\Omega = \Omega(K)$ depending on the accuracy parameter *K*, and in this example we consider three different such classes. In Theorem 2, we required that $K > 2$ so that $K - 2 > 0$, but this is not necessary to show the impossibility statement 1, so we consider $K = 1, 3, 6$. Full details of the following experiment are given in *SI Appendix*.

To show that it is impossible to compute NNs that can solve (P_3) to arbitrary accuracy we consider FIRENETs Φ_{A_n} (the NNs in Theorem 3) and learned ISTA (LISTA) networks Ψ_{A_n} based on the architecture choice from ref. 40. The networks are trained to high accuracy on training data on the form of Eq. **5** with $R = 8{,}000$ training samples and n given as in Table 1. In all cases $N = 20$, $m = N - 1$, and the $x_{k,n}$ s minimizing (P_3) with input data $(y_{k,n}, A_n)$ are all 6-sparse. The choice of *N*, *m*, and sparsity is to allow for fast training; other choices are certainly possible.

Table 1 shows the errors for both LISTA and FIRENETs. Both network types are given input data (y_n, A_n), approximating the true data (y, A). As is clear from Table 1, none of the networks are able to compute an approximation to the true minimizer in $\Xi_3(A, y)$ to *K* digits accuracy. However, both networks compute an approximation with $K - 1$ digits accuracy. These observations agree precisely with Theorem 2.

The Subtlety and Difficulty of Removing Instabilities and the Need for Additional Assumptions. Theorem 2 shows that the problems (P_j) cannot, in general, be solved by any training algorithm. Hence, any attempt at using the problems (P_j) as approximate solution maps of the general inverse problem in Eq. **1**, without additional assumptions, is doomed to fail. This is not just the case for reconstruction using sparse regularization, but also applies to other methods. In fact, any stable and accurate reconstruction procedure must be "kernel aware" (22), a property that most DL methods do not enforce. A reconstruction method $\Psi: \mathbb{C}^m \to \mathbb{C}^N$ lacks kernel awareness if it approximately recovers two vectors

$$\|\Psi(Ax) - x\| \leq \epsilon \quad \text{and} \quad \|\Psi(Ax') - x'\| \leq \epsilon \qquad \textbf{[6]}$$

whose difference $\|x - x'\| \gg 2\epsilon$ is large, but where the difference lies close to the null space of A (which is nontrivial due to $m < N$) so that $\|A(x - x')\| < \epsilon$. In particular, by applying Eq. **6** and the triangle inequality twice, we have that

$$\|\Psi(Ax) - \Psi(Ax')\| \geq \|x - x'\| - 2\epsilon \qquad \textbf{[7]}$$

implying instability, as it requires only a perturbation $e = A(x' - x)$ of size $\|e\| < \epsilon$ for $\Psi(Ax + e) = \Psi(Ax')$ to reconstruct the wrong image. The issue here is that if we want to accurately recover x and x', i.e., we want Eq. **6** to hold, then we cannot simultaneously have that $x - x'$ lies close to the kernel. Later we shall see conditions that circumvent this issue for our model class, thereby allowing us to compute stable and accurate NNs.

While training can encourage the conditions in Eq. **6** to hold, it is not clear how many of the defense techniques in DL, simultaneously, will protect against the condition $\|A(x - x')\| < \epsilon$. One standard attempt to remedy instabilities is adversarial training (41). However, while this strategy can potentially avoid Eq. **6**, it may yield poor performance. For example, consider the following optimization problem, which generates a reconstruction in the form of a NN given samples $\Theta = \{(y_s, x_s) : s = 1, \ldots, R, Ax_s = y_s\}$ and $\epsilon, \lambda > 0$:

$$\min_{\phi \in \mathcal{N}_{m,N}} \sum_{s=1}^{R} \max_{\|e\|_{l^2} \leq \epsilon} \{\|x_s - \phi(y_s)\|_{l^2}^2 + \lambda \|x_s - \phi(y_s + e)\|_{l^2}^2\}. \qquad \textbf{[8]}$$

In other words, for each training point $(y, x) \in \Theta$ we find the worst-case perturbation e in the ϵ -ball around y. This is a simplified model of what one might do using generative adversarial networks (GANs) to approximate adversarial perturbations (42, 43). For simplicity, assume that A has full row rank m and that we have access to exact measurements $y_s = Ax_s$. Suppose that our sample is such that $\min_{i \neq j} \|y_i - y_j\|_{l^2} > 2\epsilon$. In this case, ϕ minimizes Eq. **8** if and only if $\phi(y_s + e) = x_s$ for all e with $\|e\|_{l^2} \leq \epsilon$. A piecewise affine network achieving this can easily be constructed using ReLU (rectified linear unit) activation functions. Now suppose that x_2 is altered so that $x_1 - x_2$ lies in the kernel of A. Then for any minimizer ϕ, we must have $\phi(y_1 + e) = \phi(y_2 + e) = (x_1 + x_2)/2$ for any e with $\|e\|_{l^2} \leq \epsilon$, and hence we can never be more than $\|x_1 - \phi(y_1)\| = \|x_1 - x_2\|_{l^2}/2$ accurate over the whole test sample. Similar arguments apply to other methods aimed at improving robustness such as adding noise to training samples (known as "jittering") (Fig. 4). Given such examples and Theorem 2, we arrive at the following question:

Question. *Are there sufficient conditions on A that imply the existence of an algorithm that can compute a neural network that is both stable and accurate for the problem in Eq.* **1***?*

Sufficient Conditions for Algorithms to Compute Stable and Accurate NNs

Sparse regularization, such as the problems (P_j), forms the core of many start-of-the-art reconstruction algorithms for inverse problems. We now demonstrate a sufficient condition (from compressed sensing) guaranteeing the existence of algorithms for stable and accurate NNs. Sparsity in levels is a standard sparsity model for natural images (44–47) as images are sparse in levels in X-lets (wavelets, curvelets, shearlets, etc.).

Definition 1 (Sparsity in Levels). *Let* $\mathbf{M} = (M_1, \ldots, M_r) \in \mathbb{N}^r$, $1 \leq M_1 < \ldots < M_r = N$, *and* $\mathbf{s} = (s_1, \ldots, s_r) \in \mathbb{N}_0^r$, *where* $s_l \leq M_l - M_{l-1}$ *for* $l = 1, \ldots, r$ $(M_0 = 0)$. $x \in \mathbb{C}^N$ *is* $(\mathbf{s}, \mathbf{M})$ *-sparse in levels if* $|\text{supp}(x) \cap \{M_{l-1} + 1, \ldots, M_l\}| \leq s_l$ *for* $l = 1, \ldots, r$. *The total sparsity is* $s = s_1 + \ldots + s_r$. *We denote the set of* $(\mathbf{s}, \mathbf{M})$ *-sparse vectors by* $\Sigma_{\mathbf{s},\mathbf{M}}$. *We also define the following measure of distance of a vector x to* $\Sigma_{\mathbf{s},\mathbf{M}}$ *by*

$$\sigma_{\mathbf{s},\mathbf{M}}(x)_{l_w^1} = \inf\{\|x - z\|_{l_w^1} : z \in \Sigma_{\mathbf{s},\mathbf{M}}\}.$$

This model has been used to explain the effectiveness of compressed sensing (46, 48–52) in real-life applications (53). For simplicity, we assume that each $s_l > 0$ and that $w_i = w_{(l)}$ if $M_{l-1} + 1 \leq i \leq M_l$ (the weights in the l_w^1 norm are constant in each level). For a vector c that is compressible in the wavelet basis, $\sigma_{\mathbf{s},\mathbf{M}}(x)_{l_w^1}$ is expected to be small if x is the vector of wavelet coefficients of c and the levels correspond to wavelet levels (54). In general, the weights are a prior on anticipated support of the vector (55), and we discuss some specific optimal choices in *SI Appendix*.

For $\mathcal{I} \subset \{1, \ldots, N\}$, let $P_{\mathcal{I}} \in \mathbb{C}^{N \times N}$ denote the projection $(P_{\mathcal{I}}x)_i = x_i$ if $i \in \mathcal{I}$ and $(P_{\mathcal{I}}x)_i = 0$ otherwise. The key kernel-aware property that allows for stable and accurate recovery of $(\mathbf{s}, \mathbf{M})$ -spare vectors for the inverse problem Eq. **1** is the weighted robust null space property in levels (wrNSPL):

Definition 2 (wrNSPL). *Let* $(\mathbf{s}, \mathbf{M})$ *be local sparsities and sparsity levels, respectively. For weights* $\{w_i\}_{i=1}^N$, $A \in \mathbb{C}^{m \times N}$ *satisfies the wrNSPL of order* $(\mathbf{s}, \mathbf{M})$ *with constants* $0 < \rho < 1$ *and* $\gamma > 0$ *if for any* $(\mathbf{s}, \mathbf{M})$ *support set* $\mathcal{I} \subset \{1, \ldots, N\}$ *(with complement* $\mathcal{I}^c = \{1, \ldots, N\} \backslash \mathcal{I}$*),*

$$\|P_{\mathcal{I}}x\|_{l^2} \leq \frac{\rho \|P_{\mathcal{I}^c}x\|_{l_w^1}}{\sqrt{\sum_{l=1}^r w_{(l)}^2 s_l}} + \gamma \|Ax\|_{l^2}, \qquad \textit{for all } x \in \mathbb{C}^N.$$

We highlight that if A satisfies the wrNSPL, then

$$\|x - x'\|_{l^2} \leq C \|A(x - x')\|_{l^2}, \quad \forall x, x' \in \Sigma_{\mathbf{s},\mathbf{M}},$$

where $C = C(\rho, \gamma) > 0$ is a constant depending only on ρ and γ (*SI Appendix*). This ensures that if $\|x - x'\|_{\ell^2} \gg 2\epsilon$, then we cannot, simultaneously, have that $\|A(x - x')\| < \epsilon$, causing the instability in Eq. **7**. Below, we give natural examples of sampling in compressed imaging where such a property holds, for known ρ and γ, with large probability. We can now state a simplified version of our result (the full version with explicit constants is given and proved in *SI Appendix*):

Theorem 3. *There exists an algorithm such that for any input sparsity parameters* $(\mathbf{s}, \mathbf{M})$, *weights* $\{w_i\}_{i=1}^N$, $A \in \mathbb{C}^{m \times N}$ *(with the input A given by* $\{A_l\}$*) satisfying the wrNSPL with constants* $0 < \rho < 1$ *and* $\gamma > 0$ *(also input), and input parameters* $n \in \mathbb{N}$ *and* $\{\delta, b_1, b_2\} \subset \mathbb{Q}_{>0}$, *the algorithm outputs a neural network* ϕ_n *with* $\mathcal{O}(n)$ *hidden layers and* $\mathcal{O}(N)$ *width with the following property: For any* $x \in \mathbb{C}^N$, $y \in \mathbb{C}^m$ *with*

$$\sigma_{\mathbf{s},\mathbf{M}}(x)_{l_w^1} + \|Ax - y\|_{l^2} \lesssim \delta, \quad \|x\|_{l^2} \lesssim b_1, \quad \|y\|_{l^2} \lesssim b_2,$$

we have $\|\phi_n(y) - x\|_{l^2} \lesssim \delta + e^{-n}$.

Hence, up to the small error term $\sigma_{\mathbf{s},\mathbf{M}}(x)_{l_w^1}$, as $n \to \infty$ (with exponential convergence), we recover x stably with an error proportional to the measurement error $\|Ax - y\|_{l^2}$. The explicit constant in front of the $\|Ax - y\|_{l^2}$ term can be thought of as an asymptotic local Lipschitz constant for the NNs as $n \to \infty$ and thus measures stability of inexact input y. The error of order $\sigma_{\mathbf{s},\mathbf{M}}(x)_{l_w^1}$ measures how close the vector x is from the model class of sparse in levels vectors. In the full version of Theorem 3, we also bound the error when we only approximately apply the nonlinear maps of the NNs and show that these errors can only accumulate slowly as n increases. In other words, we also gain a form of numerical stability of the forward pass of the NN. We call our NNs FIRENETs.

Remark 5 (unrolling does not in general yield an algorithm producing an accurate network). Unrolling iterative methods has a rich history in DL (9, 56). Note, however, that Theorem 2 demonstrates that despite the existence of an accurate neural network, there are scenarios where no algorithm exists that can compute it. Thus, unrolling optimization methods can work only under certain assumptions. Our results are related to the work of Ben-Tal and Nemirovski (57) (see also ref. 58), which shows how key assumptions such as the robust nullspace property help bound the error of the approximation to a minimizer in terms of error bounds on the approximation to the objective function. This is related to robust optimization (59).

In the case that we do not know ρ or γ (the constants in the definition of wrNSPL), we can perform a log-scale grid search for suitable parameters. By increasing the width of the NNs to $\mathcal{O}(N\log(n))$, we can still gain exponential convergence in n by choosing the parameters in the grid search that lead to the vector with minimal F_3^A [the objective function of (P_3)]. In other cases, such as Theorem 4 below, it is possible to prove probabilistic results where ρ and γ are known.

Examples in Image Recovery. As an application, we consider Fourier and Walsh (binary) sampling, using Haar wavelets as a sparsifying transform. Our results can also be generalized to infinite-dimensional settings via higher-order Daubechies wavelets. Theorem 3 is quite general and there are numerous other applications where problem-dependent results similar to Theorem 4 can be shown.

Let $K = 2^r$ for $r \in \mathbb{N}$, and set $N = K^d$ so that the objective is to recover a vectorized d-dimensional tensor $c \in \mathbb{C}^N$. Let $V \in \mathbb{C}^{N\times N}$ correspond to the d-dimensional discrete Fourier or Walsh transform (*SI Appendix*). Let $\mathcal{I} \subset \{1, \ldots, N\}$ be a sampling pattern with cardinality $m = |\mathcal{I}|$ and let $D = \mathrm{diag}(d_1, \ldots, d_m) \in \mathbb{C}^{m\times m}$ be a suitable diagonal scaling matrix, whose entries along the diagonal depend only on $\mathcal{I}$. We assume we can observe the subsampled, scaled and noisy measurements $y = DP_{\mathcal{I}}Vc + e \in \mathbb{C}^m$, where projection $P_{\mathcal{I}}$ is treated as an $m \times N$ matrix by ignoring the zero entries.

To recover a sparse representation of c, we consider Haar wavelet coefficients. Denote the discrete d-dimensional Haar wavelet transform by $\Psi \in \mathbb{C}^{N\times N}$ and note that $\Psi^* = \Psi^{-1}$ since Ψ is unitary. To recover the wavelet coefficients $x = \Psi c$ of c, we consider the matrix $A = DP_{\mathcal{I}}V\Psi^*$ and observe that $y = Ax + e = DP_{\mathcal{I}}Vc + e$. A key result in this work is that we can design a probabilistic sampling strategy (*SI Appendix*), for both Fourier and Walsh sampling in d dimensions, requiring no more than $m \gtrsim (s_1 + \ldots + s_r) \cdot \mathcal{L}$ samples, that can ensure with high probability that A satisfies the wrNSPL with certain constants. The sparsity in levels structure (*Definition 1*) is chosen to correspond to the r wavelet levels. Here $\mathcal{L}$ is a logarithmic term in N, m, s, and $\epsilon_{\mathbb{P}}^{-1}$ [where $\epsilon_{\mathbb{P}} \in (0, 1)$ is a probability]. This result is crucial, as it makes A kernel aware for vectors that are approximately $(\mathbf{s},\mathbf{M})$ -sparse and allows us (using Theorem 3) to design NNs that can stably and accurately recover approximately $(\mathbf{s},\mathbf{M})$ -sparse vectors. Moreover, due to the exponential convergence in Theorem 3, the depth of these NNs depends only logarithmically on the error δ. Below follows a simplified version of our result (the full precise version is given and proved in *SI Appendix*).

Theorem 4. *Consider the above setup of recovering wavelet coefficients $x = \Psi c$ of a tensor $c \in \mathbb{C}^{K^d}$ from subsampled, scaled and noisy Fourier or Walsh measurements $y = DP_{\mathcal{I}}Vc + e$. Let $A = DP_{\mathcal{I}}V\Psi^*$, $m = |\mathcal{I}|$, and $\epsilon_{\mathbb{P}} \in (0, 1)$. We then have the following*:

1) *If $\mathcal{I} \subset \{1, \ldots, N\}$ is a random sampling pattern drawn according to the strategy specified in SI Appendix, and*

$$m \gtrsim (s_1 + \cdots + s_r) \cdot \mathcal{L},$$

then with probability $1 - \epsilon_{\mathbb{P}}$, A satisfies the wrNSPL of order $(\mathbf{s},\mathbf{M})$ with constants $(\rho, \gamma) = (1/2, \sqrt{2})$, $w_{(l)} = \sqrt{s/s_l}$, $s = s_1 + \cdots + s_r$. Here $\mathcal{L}$ denotes a term logarithmic in $\epsilon_{\mathbb{P}}^{-1}, N, m$ and s.

2) *Suppose $\mathcal{I}$ is chosen as above. For any $\delta \in (0, 1)$, let $\mathcal{J}(\delta, \mathbf{s}, \mathbf{M}, w)$ be the set of all $y = Ax + e \in \mathbb{C}^m$ where*

$$\|x\|_{l^2} \le 1, \quad \max\left\{\sigma_{\mathbf{s},\mathbf{M}}(x)_{l_w^1}, \|e\|_{l^2}\right\} \le \delta. \qquad \textbf{[9]}$$

We provide an algorithm that constructs a neural network ϕ with $\mathcal{O}(\log(\delta^{-1}))$ hidden layers [and width bounded by $2(N + m)$] such that with probability at least $1 - \epsilon_{\mathbb{P}}$,

$$\|\phi(y) - c\|_{l^2} \lesssim \delta, \quad \forall y = Ax + e \in \mathcal{J}(\delta, \mathbf{s}, \mathbf{M}, w).$$

Balancing the Stability and Accuracy Trade-Off

Current DL methods for image reconstruction can be unstable in the sense that 1) a tiny perturbation, in either the image or the sampling domain, can cause severe artifacts in the reconstructed image (Fig. 2, *Top* row) and/or 2) a tiny detail in the image domain might be washed out in the reconstructed image (lack of accuracy), resulting in potential false negatives. Inevitably, there is a stability–accuracy trade-off for this type of linear inverse problem, making it impossible for any reconstruction method to become arbitrarily stable without sacrificing accuracy or vice versa. Here, we show that the NNs computed by our algorithm (FIRENETs) are stable with respect to adversarial perturbations and accurate for images that are sparse in wavelets (cf. Theorem 4). As most images are sparse in wavelets, these networks also show great generalization properties to unseen images.

Adversarial Perturbations for AUTOMAP and FIRENETs. Fig. 2 (*Top* row) shows the stability test, developed in ref. 19, applied to the automated transform by manifold approximation (AUTOMAP) (60) network used for MRI reconstruction with 60% subsampling. The stability test is run on the AUTOMAP network to find a sequence of perturbations $\|e_1\|_{l^2} < \|e_2\|_{l^2} < \|e_3\|_{l^2}$. As can be seen from Fig. 2, *Top* row, the network reconstruction completely deforms the image and the reconstruction is severely unstable (similar results for other networks are demonstrated in ref. 19).

In contrast, we have applied the stability test, but now for the FIRENETs reported in this paper. Fig. 2 (*Bottom* row) shows the results for the constructed FIRENETs, where we rename the perturbations $\tilde{e}_j$ to emphasize the fact that these perturbations are sought for the FIRENETs and have nothing to do with the adversarial perturbations for AUTOMAP. We now see that despite the search for adversarial perturbations, the reconstruction remains stable. The error in the reconstruction was also found to be at most of the same order of the perturbation (as expected from the stability in Theorem 3). In applying the test to FIRENETs, we tested/tuned the parameters in the gradient ascent algorithm considerably (much more so than was needed for applying the test to AUTOMAP, where finding instabilities was straightforward) to find the worst reconstruction results, yet the reconstruction remained stable. Note also that this is just one form of stability test and it is likely that there are many other tests for creating instabilities for NNs for inverse problems. This highlights the importance of results such as Theorem 3, which guarantees stability regardless of the perturbation.

To demonstrate the generalization properties of our NNs, we show the stability test applied to FIRENETs for a range of images in *SI Appendix*. This shows stability across different types of images and highlights that conditions such as Definition 2 allow great generalization properties.

Stabilizing Unstable NNs with FIRENETs. Our NNs also act as a stabilizer. For example, Fig. 3 shows the adversarial example for AUTOMAP (taken from Fig. 2), but now shows what happens when we take the reconstruction from AUTOMAP as an input to our FIRENETs. Here we use the fact that we can view our networks as approximations of unrolled and restarted iterative methods, allowing us to use the output of AUTOMAP as an additional input for the reconstruction. We see that FIRENETs fix the output of AUTOMAP and stabilize the reconstruction. Moreover, the full concatenation itself of the networks remains stable to adversarial attacks.

The Stability vs. Accuracy Trade-Off and False Negatives. It is easy to produce a perfectly stable network: The zero network is the obvious candidate! However, this network would obviously have poor performance and produce many false negatives. The challenge is to simultaneously ensure performance and stability. Fig. 4 highlights this issue. Here we have trained two NNs to recover a set of ellipses images from noise-free and noisy Fourier measurements. The noise-free measurements are generated as $y = Ax$, where $A \in \mathbb{C}^{m \times N}$ is a subsampled discrete Fourier transform, with $m/N = 0.15$ and $N = 1,024^2$. The noisy measurements are generated as $y = Ax + ce$, where A is as before, and the real and imaginary components of $e \in \mathbb{C}^m$ are drawn from a zero mean and unit variance normal distribution $\mathcal{N}(0,1)$, and $c \in \mathbb{R}$ is drawn from the uniform distribution $\mathrm{Unif}([0,100])$. The noise $ce \in \mathbb{C}^m$ is generated on the fly during the training process.

The trained networks use a standard benchmarking architecture for image reconstruction and map $y \mapsto \phi(A^* y)$, where $\phi\colon \mathbb{C}^N \to \mathbb{R}^N$ is a trainable U-net NN (8, 61). Training networks with noisy measurements, using for example this architecture, have previously been used as an example of how to create NNs that are robust toward adversarial attacks (62). As we can see from Fig. 4 (*Bottom* row) this is the case, as it does indeed create a NN that is stable with respect to worst-case perturbations. However, a key issue is that it is also producing false negatives due to its inability to reconstruct details. Similarly, as reported in the 2019 FastMRI challenge, trained NNs that performed well in terms of standard image quality metrics were prone to false negatives: They failed to reconstruct small, but physically relevant image abnormalities (25). Pathologies, generalization, and AI-generated hallucinations were subsequently a focus of the 2020 challenge (26). FIRENET, on the other hand, has a guaranteed performance (on images approximately sparse in wavelet bases) and stability, given specific conditions on the sampling procedure. The challenge is to determine the optimal balance between accuracy and stability, a well-known problem in numerical analysis.

Concluding Remarks

1) (Algorithms may not exist—Smale's 18th problem) There are well-conditioned problems where accurate NNs exist, but no algorithm can compute them. Understanding this phenomenon is essential to addressing Smale's 18th problem on the limits of AI. Moreover, limitations established in this paper suggest a classification theory describing the conditions needed for the existence of algorithms that can compute stable and accurate NNs (remark 5).
2) (Classifications and Hilbert's program) The strong optimism regarding the abilities of AI is comparable to the optimism surrounding mathematics in the early 20th century, led by D. Hilbert. Hilbert believed that mathematics could prove or disprove any statement and, moreover, that there were no restrictions on which problems could be solved by algorithms. Gödel (36) and Turing (37) turned Hilbert's optimism upside down by their foundational contributions establishing impossibility results on what mathematics and digital computers can achieve.

   Hilbert's program on the foundations of mathematics led to a rich mathematical theory and modern logic and computer science, where substantial efforts were made to classify which problems can be computed. We have sketched a similar program for modern AI, where we provide certain sufficient conditions for the existence of algorithms to produce stable and accurate NNs. We believe that such a program on the foundations of AI is necessary and will act as an invaluable catalyst for the advancement of AI.
3) (Trade-off between stability and accuracy) For inverse problems there is an intrinsic trade-off between stability and accuracy. We demonstrated NNs that offer a blend of both stability and accuracy, for the sparsity in levels class. Balancing these two interests is crucial for applications and will no doubt require a myriad of future techniques to be developed.

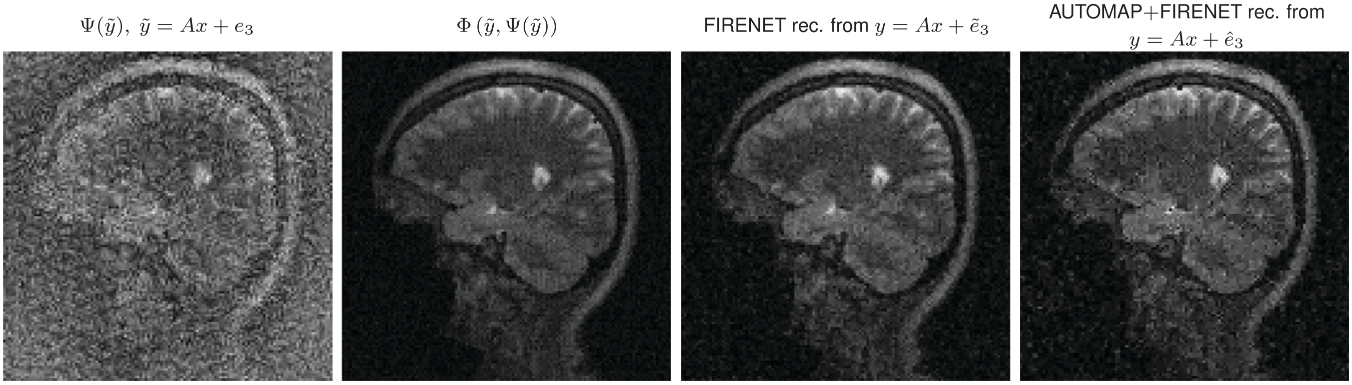
$\Psi(\tilde{y}),\ \tilde{y} = Ax + e_3$
$\Phi\left(\tilde{y}, \Psi(\tilde{y})\right)$
FIRENET rec. from $y = Ax + \tilde{e}_3$
AUTOMAP+FIRENET rec. from $y = Ax + \hat{e}_3$

Fig. 3. Adding a few FIRENET layers at the end of AUTOMAP makes it stable. The FIRENET $\Phi\colon \mathbb{C}^m \times \mathbb{C}^N \to \mathbb{C}^N$ takes as input measurements $y \in \mathbb{C}^m$ and an initial guess for x, which we call $x_0 \in \mathbb{C}^N$. We now concatenate a 25-layer ($p = 5$, $n = 5$) FIRENET Φ and the AUTOMAP network $\Psi\colon \mathbb{C}^m \to \mathbb{C}^N$, by using the output from AUTOMAP as initial guess x_0; i.e., we consider the neural network mapping $y \mapsto \Phi(y, \Psi(y))$. In this experiment, we consider the image x from Fig. 2 and the perturbed measurements $\tilde{y} = Ax + e_3$ (here A is as in Fig. 2). *Left* shows the reconstruction of AUTOMAP from Fig. 2. *Center Left* shows the reconstruction of FIRENET with $x_0 = \Psi(\tilde{y})$. *Center Right* shows the reconstruction of FIRENET from Fig. 2. *Right* shows the reconstruction of the concatenated network with a worst-case perturbation $\hat{e}_3$ such that $\|\hat{e}_3\|_{l^2} \geq \|e_3\|_{l^2}$. In all other experiments we set $x_0 = 0$ and consider Φ as a mapping $\Phi\colon \mathbb{C}^m \to \mathbb{C}^N$.

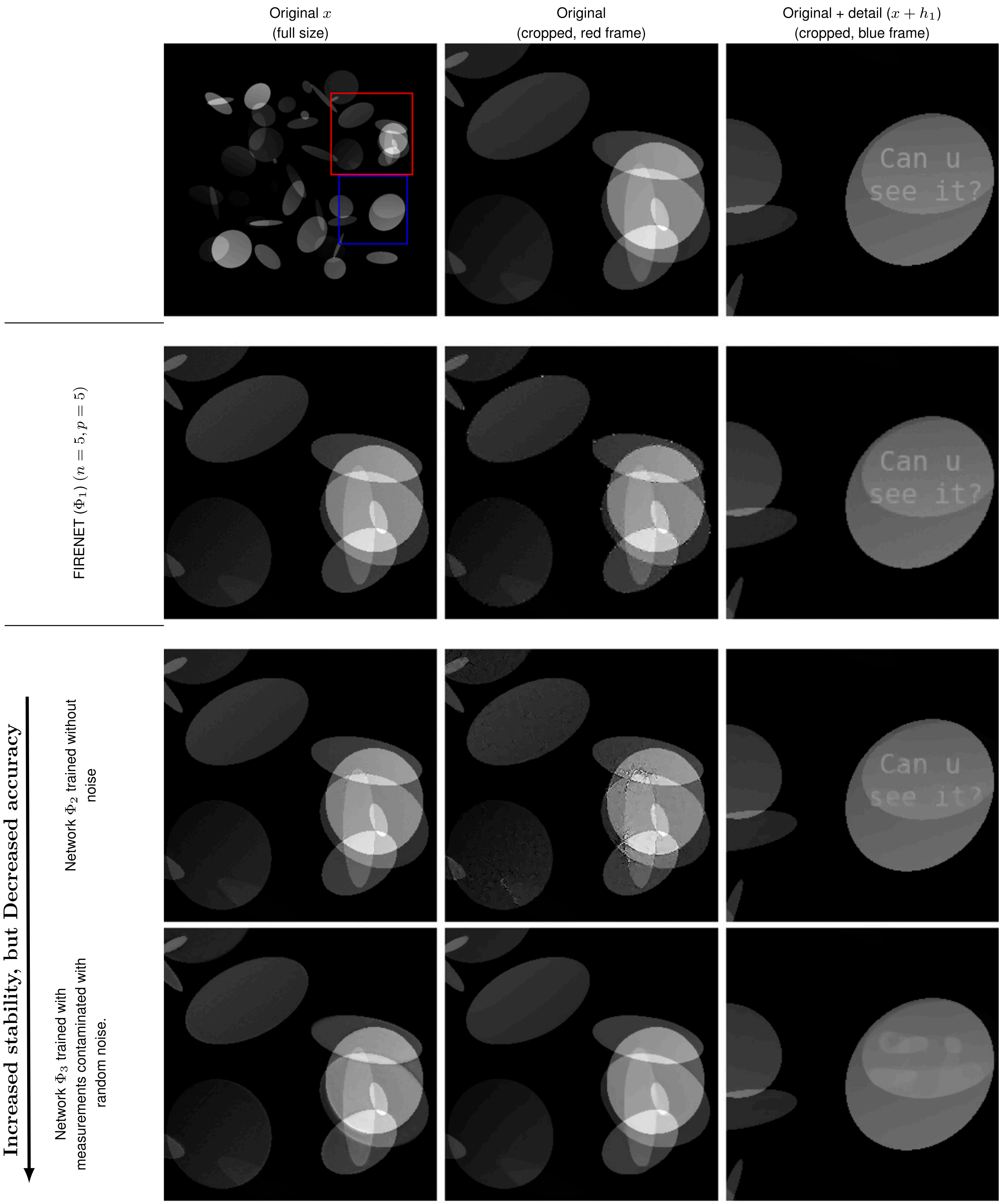
Original x
(full size)
Original
(cropped, red frame)
Original + detail $(x + h_1)$
(cropped, blue frame)
FIRENET (Φ_1) $(n = 5, p = 5)$
Network Φ_2 trained without noise
Network Φ_3 trained with measurements contaminated with random noise.
Increased stability, but Decreased accuracy
Can u see it?

Fig. 4. Trained neural networks with limited performance can be stable. We examine the accuracy/stability trade-off for linear inverse problems by considering three reconstruction networks $\Phi_j : \mathbb{C}^m \to \mathbb{C}^N$, $j = 1, 2, 3$. Here Φ_1 is a FIRENET, whereas Φ_2 and Φ_3 are the U-nets mentioned in the main text, trained without and with noisy measurements, respectively. For each network, we compute a perturbation $w_j \in \mathbb{C}^N$ meant to simulate the worst-case effect, and we show a cropped version of the perturbed images $x + w_j$ in *Left* column (rows 2 to 4). In *Center* column (rows 2 to 4), we show the reconstructed images $\Phi_j(A(x + w_j))$ from each of the networks. In *Right* column (rows 2 to 4) we test the networks' ability to reconstruct a tiny detail h_1, in the form of the text "Can u see it?". As we see, the network trained on noisy measurements is stable to worst-case perturbations, but it is not accurate. Conversely, the network trained without noise is accurate but not stable. The FIRENET is balancing this trade-off and is accurate for images that are sparse in wavelets and stable to worst-case perturbations.

Tracing out the optimal stability vs. accuracy trade-off remains largely an open problem and depends on several factors such as the model class one wishes to recover, the error tolerance of the application, and the error metric used. We have shown stability and accuracy results in the l^2 norm, since it is common in the literature to measure noise via this norm. We expect a program quantifying the stability and accuracy trade-off to be of particular relevance in the increasing number of real-world implementations of machine learning in inverse problems.

4) (Inverse problems vs. classification problems) The mathematical techniques used in this paper are applied to inverse problems. However, the mathematical framework of ref. 33 can be used to produce similar impossibility results for computing NNs in classification problems (63).
5) (Future work—Which NNs can be computed?) There is an enormous literature (29, 30, 64–66) on the existence of NNs with great approximation qualities. However, Theorem 2 shows that only certain accuracy may be computationally achievable. Our results are just the beginning of a mathematical theory studying which NNs can be computed by algorithms. This opens up for a theory covering other sufficient (and potentially necessary) conditions guaranteeing stability and accuracy and extensions to other inverse problems such as phase retrieval (67, 68). One can also prove similar computational barriers in other settings via the tools developed in this paper.

Methods: The Solvability Complexity Index Hierarchy

Our proof techniques for fundamental barriers in Theorem 2 stem from the mathematics behind the solvability complexity index (SCI) hierarchy (33, 69–78). The SCI hierarchy generalizes the fundamental problems of Smale (79, 80) on existence of algorithms and work by McMullen (81) and Doyle and McMullen (82). We extend and refine these techniques, in particular those of ref. 33, and generalize the mathematics behind the extended Smale's ninth problem (33, 34)—which also builds on the SCI hierarchy. More precisely, to prove our results we develop the concept of sequential general algorithms. General algorithms are a key tool in the mathematics of the SCI hierarchy. Sequential general algorithms extend this concept and capture the notion of adaptive and/or probabilistic choice of training data. The architectures of the NNs in Theorem 3 are based on unrolled primal–dual iterations for (P_3). In addition to providing stability, the wrNSPL allows us to prove exponential convergence through a careful restarting and reweighting scheme. Full theoretical derivations are given in *SI Appendix*.

Data Availability. All the code and data used to produce the figures in this paper are available from GitHub, https://www.github.com/Comp-Foundations-and-Barriers-of-AI/firenet.

ACKNOWLEDGMENTS. We are grateful to Kristian Haug for allowing us to use parts of code for unrolled primal–dual algorithms in TensorFlow. This work was supported by a Research Fellowship at Trinity College, Cambridge University (to M.J.C.), as well as by a Leverhulme Prize and a Royal Society University Research Fellowship (to A.C.H.).

1. A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks " in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger, Eds. (Curran Associates, Inc., 2012), pp. 1097–1105.
2. K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2016), pp. 770–778.
3. R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2014), pp. 580–587.
4. G. Hinton *et al.*, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **29**, 82–97 (2012).
5. J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, V. Svetnik, Deep neural nets as a method for quantitative structure-activity relationships. *J. Chem. Inf. Model.* **55**, 263–274 (2015).
6. G. E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio Speech Lang. Process.* **20**, 30–42 (2011).
7. U. S. Kamilov *et al.*, Learning approach to optical tomography. *Optica* **2**, 517–522 (2015).
8. K. H. Jin, M. T. McCann, E. Froustey, M. Unser, Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Process.* **26**, 4509–4522 (2017).
9. M. T. McCann, K. H. Jin, M. Unser, Convolutional neural networks for inverse problems in imaging: A review. *IEEE Signal Process. Mag.* **34**, 85–95 (2017).
10. K. Hammernik *et al.*, Learning a variational network for reconstruction of accelerated MRI data. *Magn. Reson. Med.* **79**, 3055–3071 (2018).
11. S. Arridge, P. Maass, O. Öktem, C. B. Schönlieb, Solving inverse problems using data-driven models. *Acta Numer.* **28**, 1–174 (2019).
12. G. Ongie *et al.*, Deep learning techniques for inverse problems in imaging. *IEEE J. Sel. Areas Inf. Theory* **1**, 39–56 (2020).
13. C. Szegedy *et al.*, "Intriguing properties of neural networks" in *International Conference on Learning Representations* (2014). https://openreview.net/forum?id=kklr_MTHMRQjG. (Accessed 3 March 2022).
14. S. M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2016), pp. 2574–2582.
15. S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, "Universal adversarial perturbations" in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2017), pp. 86–94.
16. N. Akhtar, A. Mian, Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* **6**, 14410–14430 (2018).
17. N. Carlini, D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text" in *2018 IEEE Security and Privacy Workshops (SPW)* (IEEE, 2018), pp. 1–7.
18. Y. Huang *et al.*, "Some investigations on robustness of deep learning in limited angle tomography" in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, G. Fichtinger, Eds. (Springer, 2018), pp. 145–153.
19. V. Antun, F. Renna, C. Poon, B. Adcock, A. C. Hansen, On instabilities of deep learning in image reconstruction and the potential costs of AI. *Proc. Natl. Acad. Sci. U.S.A.* **117**, 30088–30095 (2020).
20. S. G. Finlayson *et al.*, Adversarial attacks on medical machine learning. *Science* **363**, 1287–1289 (2019).
21. I. Y. Tyukin, D. J. Higham, A. N. Gorban, "On adversarial examples and stealth attacks in artificial intelligence systems" in *2020 International Joint Conference on Neural Networks* (IEEE, 2020), pp. 1–6.
22. N. M. Gottschling, V. Antun, B. Adcock, A. C. Hansen, The troublesome kernel: Why deep learning for inverse problems is typically unstable. arXiv [Preprint] (2020). https://arxiv.org/abs/2001.01258 (Accessed 5 January 2020).
23. N. Baker *et al.*, "Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence" (Tech. Rep. 1478744, USDOE Office of Science, 2019).
24. R. Hamon, H. Junklewitz, I. Sanchez, *Robustness and Explainability of Artificial Intelligence - From Technical to Policy Solutions* (Publications Office of the European Union, 2020).
25. F. Knoll *et al.*, Advancing machine learning for MR image reconstruction with an open competition: Overview of the 2019 fastMRI challenge. *Magn. Reson. Med.* **84**, 3054–3070 (2020).
26. M. J. Muckley *et al.*, Results of the 2020 fastMRI Challenge for Machine Learning MR Image Reconstruction. *IEEE Trans. Med. Imaging* **40**, 2306–2317 (2021).
27. C. Belthangady, L. A. Royer, Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction. *Nat. Methods* **16**, 1215–1225 (2019).
28. D. P. Hoffman, I. Slavitt, C. A. Fitzpatrick, The promise and peril of deep learning in microscopy. *Nat. Methods* **18**, 131–132 (2021).
29. A. Pinkus, Approximation theory of the MLP model in neural networks. *Acta Numer.* **8**, 143–195 (1999).
30. R. DeVore, B. Hanin, G. Petrova, Neural network approximation. *Acta Numer.* **30**, 327–444 (2021).
31. B. Adcock, N. Dexter, The gap between theory and practice in function approximation with deep neural networks. *SIAM J. Math. Data Sci.* **3**, 624–655 (2021).
32. C. F. Higham, D. J. Higham, Deep learning: An introduction for applied mathematicians. *SIAM Rev.* **61**, 860–891 (2019).
33. A. Bastounis, A. C. Hansen, V. Vlačić, The extended Smale's 9th problem – On computational barriers and paradoxes in estimation, regularisation, computer-assisted proofs and learning. arXiv [Preprint] (2021). https://arxiv.org/abs/2110.15734 (Accessed 29 October 2021).
34. S. Smale, Mathematical problems for the next century. *Math. Intell.* **20**, 7–15 (1998).
35. A. Turing, I.-Computing machinery and intelligence. *Mind* **LIX**, 433–460 (1950).
36. K Gödel, Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatsh. Math. Phys.* **38**, 173–198 (1931).
37. A. Turing, On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **42**, 230–265 (1936).
38. S. Weinberger, *Computers, Rigidity, and Moduli: The Large-Scale Fractal Geometry of Riemannian Moduli Space* (Princeton University Press, Princeton, NJ, 2004).
39. P. Niyogi, S. Smale, S. Weinberger, A topological view of unsupervised learning from noisy data. *SIAM J. Comput.* **40**, 646–663 (2011).
40. X. Chen, J. Liu, Z. Wang, W. Yin, "Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds" in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Q. Weinberger, Eds. (Curran Associates, Inc., 2018), pp. 9061–9071.

41. A. Raj, Y. Bresler, B. Li, "Improving robustness of deep-learning-based image reconstruction" in *International Conference on Machine Learning*, H. Daumé III, A. Singh, Eds. (PMLR, 2020), pp. 7932–7942.
42. I. Goodfellow *et al.*, "Generative adversarial nets" in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Q. Weinberger, Eds. (Curran Associates, Inc., 2014), pp. 2672–2680.
43. M. Arjovsky, S. Chintala, L. Bottou, "Wasserstein generative adversarial networks" in *International Conference on Machine Learning*, D. Precup, Y. W. Teh, Eds. (PMLR, 2017), vol. 70, pp. 214–223.
44. B. Adcock, A. C. Hansen, C. Poon, B. Roman, "Breaking the coherence barrier: A new theory for compressed sensing" in *Forum of Mathematics, Sigma* (Cambridge University Press, 2017), vol. 5.
45. A. Bastounis, A. C. Hansen, On the absence of uniform recovery in many real-world applications of compressed sensing and the restricted isometry property and nullspace property in levels. *SIAM J. Imaging Sci.* **10**, 335–371 (2017).
46. B. Adcock, A. C. Hansen, *Compressive Imaging: Structure, Sampling, Learning* (Cambridge University Press, 2021).
47. C. Boyer, J. Bigot, P. Weiss, Compressed sensing with structured sparsity and structured acquisition. *Appl. Comput. Harmon. Anal.* **46**, 312–350 (2019).
48. E. J. Candès, J. Romberg, T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory* **52**, 489–509 (2006).
49. E. J. Candes, J. K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements. *Commun. Pure Appl. Math.* **59**, 1207–1223 (2006).
50. D. L. Donoho, Compressed sensing. *IEEE Trans. Inf. Theory* **52**, 1289–1306 (2006).
51. D. L. Donoho, J. Tanner, Counting faces of randomly-projected polytopes when the projection radically lowers dimension. *J. Am. Math. Soc.* **22**, 1–53 (2009).
52. A. Cohen, W. Dahmen, R. DeVore, Compressed sensing and best k-term approximation. *J. Am. Math. Soc.* **22**, 211–231 (2009).
53. A. Jones, A. Tamtögl, I. Calvo-Almazán, A. Hansen, Continuous compressed sensing for surface dynamical processes with helium atom scattering. *Sci. Rep.* **6**, 27776 (2016).
54. R. A. DeVore, Nonlinear approximation. *Acta Numer.* **7**, 51–150 (1998).
55. M. P. Friedlander, H. Mansour, R. Saab, Ö. Yilmaz, Recovering compressively sampled signals using partial support information. *IEEE Trans. Inf. Theory* **58**, 1122–1134 (2012).
56. V. Monga, Y. Li, Y. C. Eldar, Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Process. Mag.* **38**, 18–44 (2021).
57. A Ben-Tal, A Nemirovski, Lectures on modern convex optimization (2020–2021). https://www2.isye.gatech.edu/ñemirovs/LMCO_LN.pdf. (Accessed 5 February 2022).
58. Y. E. Nesterov, A. Nemirovski, On first-order algorithms for l1/nuclear norm minimization. *Acta Numer.* **22**, 509–575 (2013).
59. A. Ben-Tal, L. El Ghaoui, A. Nemirovski, *Robust Optimization* (Princeton Series in Applied Mathematics, Princeton University Press, 2009).
60. B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, M. S. Rosen, Image reconstruction by domain-transform manifold learning. *Nature* **555**, 487–492 (2018).
61. J. Long, E. Shelhamer, T. Darrell, "Fully convolutional networks for semantic segmentation" in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE, 2015), pp. 3431–3440.
62. M. Genzel, J. Macdonald, M. März, "Solving inverse problems with deep neural networks–Robustness included?" in *Transactions on Pattern Analysis and Machine Intelligence*, 10.1109/TPAMI.2022.3148324 (2022).
63. A. Bastounis, A. C. Hansen, V. Vlacic, The mathematics of adversarial attacks in AI – Why deep learning is unstable despite the existence of stable neural networks. arXiv [Preprint] (2021). https://arxiv.org/abs/2109.06098 (Accessed 13 September 2021).
64. H. Bölcskei, P. Grohs, G. Kutyniok, P. Petersen, Optimal approximation with sparsely connected deep neural networks. *SIAM J. Math. Data Sci.* **1**, 8–45 (2019).
65. I. Daubechies, R. DeVore, S. Foucart, B. Hanin, G. Petrova, Nonlinear approximation and (deep) ReLU networks. *Constr. Approx.* **55**, 127–172 (2021).
66. W. E. S. Wojtowytsch, On the Banach spaces associated with multi-layer ReLU networks: Function representation, approximation theory and gradient descent dynamics. *CSIAM Transact. Appl. Math* **1**, 387–440 (2020).
67. E. J. Candes, T. Strohmer, V. Voroninski, Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Commun. Pure Appl. Math.* **66**, 1241–1274 (2013).
68. A. Fannjiang, T. Strohmer, The numerics of phase retrieval. *Acta Numer.* **29**, 125–228 (2020).
69. A. C. Hansen, On the solvability complexity index, the n-pseudospectrum and approximations of spectra of operators. *J. Am. Math. Soc.* **24**, 81–124 (2011).
70. A. C. Hansen, O. Nevanlinna, Complexity issues in computing spectra, pseudospectra and resolvents. *Banach Cent. Publ.* **112**, 171–194 (2016).
71. J. Ben-Artzi, M. J. Colbrook, A. C. Hansen, O. Nevanlinna, M. Seidel, Computing spectra – On the solvability complexity index hierarchy and towers of algorithms. arXiv [Preprint] (2020). https://arxiv.org/abs/1508.03280 (Accessed 15 June 2020).
72. J. Ben-Artzi, A. C. Hansen, O. Nevanlinna, M. Seidel, New barriers in complexity theory: On the solvability complexity index and the towers of algorithms. *C. R. Math.* **353**, 931–936 (2015).
73. M. Colbrook, "The foundations of infinite-dimensional spectral computations," PhD thesis, University of Cambridge, Cambridge, UK (2020).
74. M. J. Colbrook, Computing spectral measures and spectral types. *Commun. Math. Phys.* **384**, 433–501 (2021).
75. M. Colbrook, A. Horning, A. Townsend, Computing spectral measures of self-adjoint operators. *SIAM Rev.* **63**, 489–524 (2021).
76. J. Ben-Artzi, M. Marletta, F. Rösler, Computing the sound of the sea in a seashell. *Found. Comput. Math.*, 10.1007/s10208-021-09509-9 (2021).
77. M. J. Colbrook, A. C. Hansen, The foundations of spectral computations via the solvability complexity index hierarchy. arXiv [Preprint] (2021). https://arxiv.org/abs/1908.09592 (Accessed 6 August 2020).
78. H. Boche, V. Pohl, "The solvability complexity index of sampling-based Hilbert transform approximations" in 2019 13th International Conference on Sampling Theory and Applications (SampTA) (IEEE, 2019), pp. 1–4.
79. S. Smale, The fundamental theorem of algebra and complexity theory. *Am. Math. Soc. Bull.* **4**, 1–36 (1981).
80. S. Smale, Complexity theory and numerical analysis. *Acta Numer.* **6**, 523–551 (1997).
81. C. McMullen, Families of rational maps and iterative root-finding algorithms. *Ann. Math.* **125**, 467–493 (1987).
82. P. Doyle, C. McMullen, Solving the quintic by iteration. *Acta Math.* **163**, 151–180 (1989).
83. R. Strack, Imaging: AI transforms image reconstruction. *Nat. Methods* **15**, 309 (2018).
84. Q. Fan *et al.*, MGH-USC Human Connectome Project datasets with ultra-high b-value diffusion MRI. *Neuroimage* **124** (Pt. B), 1108–1114 (2016).