Article

Patterns

Stabilizing deep tomographic reconstruction: Part B. Convergence analysis and adversarial attacks

Highlights

- Heuristically designed ACID framework is analyzed to support its convergence
- Some idealizations and approximations are involved in the convergence analysis
- Adversarial attack algorithm is developed to test stability of the entire ACID workflow
- Convergence of ACID is empirically shown in terms of the Lipschitz constant

Authors

Weiwen Wu, Dianlin Hu, Wenxiang Cong, ..., Hengyong Yu, Varut Vardhanabhuti, Ge Wang

Correspondence

hengyong-yu@ieee.org (H.Y.), varv@hku.hk (V.V.), wangg6@rpi.edu (G.W.)

In brief

We provide an initial theoretical analysis on the convergence of the analytic compressed iterative deep (ACID) scheme and design a dedicated adversarial attacking algorithm to perturb the ACID as a whole and test its stability systematically. In our experiments, we also demonstrate the convergence of the ACID iteration in terms of the Lipschitz constant and the local stability of the ACID scheme against noise. These results help understand the mechanism and performance of ACID and serve as a basis for further research.

Wu et al., 2022, Patterns *3*, 100475
May 13, 2022
https://doi.org/10.1016/j.patter.2022.100475

CellPress

Patterns

CellPress
OPEN ACCESS

Article

Stabilizing deep tomographic reconstruction: Part B. Convergence analysis and adversarial attacks

Weiwen Wu,[1,2,6] Dianlin Hu,[3] Wenxiang Cong,[1] Hongming Shan,[1,4] Shaoyu Wang,[5] Chuang Niu,[1] Pingkun Yan,[1] Hengyong Yu,[5,7,*] Varut Vardhanabhuti,[6,*] and Ge Wang[1,*]

[1]Biomedical Imaging Center, Center for Biotechnology and Interdisciplinary Studies, Department of Biomedical Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA
[2]School of Biomedical Engineering, Sun Yat-sen University, Shenzhen, Guangdong, China
[3]The Laboratory of Image Science and Technology, Southeast University, Nanjing, China
[4]Institute of Science and Technology for Brain-inspired Intelligence, Fudan University, Shanghai, China
[5]Department of Electrical & Computer Engineering, University of Massachusetts Lowell, Lowell, MA, USA
[6]Department of Diagnostic Radiology, Li Ka Shing Faculty of Medicine, University of Hong Kong, Hong Kong SAR, China
[7]Lead contact
*Correspondence: hengyong-yu@ieee.org (H.Y.), varv@hku.hk (V.V.), wangg6@rpi.edu (G.W.)
https://doi.org/10.1016/j.patter.2022.100475

THE BIGGER PICTURE For deep tomographic reconstruction to realize its full potential in practice, it is critically important to address the instabilities of deep reconstruction networks, which were identified in a recent PNAS paper. Our analytic compressed iterative deep (ACID) framework has provided an effective solution to address this challenge by synergizing deep learning and compressed sensing through iterative refinement. Here, we provide an initial convergence analysis, describe an algorithm to attack the entire ACID workflow, and establish not only its capability of stabilizing an unstable deep reconstruction network but also its stability against adversarial attacks dedicated to ACID as a whole. Although our theoretical results are under approximations, they shed light on the converging mechanism of ACID, serving as a basis for further investigation.

1 2 3 4 5 **Proof-of-Concept:** Data science output has been formulated, implemented, and tested for one domain/problem

SUMMARY

Due to lack of the kernel awareness, some popular deep image reconstruction networks are unstable. To address this problem, here we introduce the bounded relative error norm (BREN) property, which is a special case of the Lipschitz continuity. Then, we perform a convergence study consisting of two parts: (1) a heuristic analysis on the convergence of the analytic compressed iterative deep (ACID) scheme (with the simplification that the CS module achieves a perfect sparsification), and (2) a mathematically denser analysis (with the two approximations: [1] A^T is viewed as an inverse A^{-1} in the perspective of an iterative reconstruction procedure and [2] a pseudo-inverse is used for a total variation operator H). Also, we present adversarial attack algorithms to perturb the selected reconstruction networks respectively and, more importantly, to attack the ACID workflow as a whole. Finally, we show the numerical convergence of the ACID iteration in terms of the Lipschitz constant and the local stability against noise.

INTRODUCTION

The vulnerability of neural networks has been demonstrated with adversarial attacks in all major deep learning tasks, from misclassification examples to deep reconstruction instabilities.[1] In the landmark paper, Antun et al. showed that deep reconstruction is unstable due to lack of kernel awareness, but sparsity-promoting reconstruction does not have such a problem.[2] To address these instabilities, we design an analytic compressed iterative deep (ACID) network.[3] The key idea behind

ACID is to combine data-driven priors and sparsity constraints to outperform either simple-minded deep reconstruction networks or established compressed sensing-based reconstruction methods. In our study, we have not only experimentally shown the merits of ACID[3] but also theoretically analyzed the rationale of ACID in terms of its converging behavior and solution characteristics. In the following, we put our analysis on ACID in the perspective of others' analyses on general computational optimization in general and existing representative image reconstruction networks in particular.

There are profound results in non-computability in the field of computer science. Computational optimization is important not only in the field of computer science but also to our real-world applications. The theoretical research on this theme can be traced back to Turing's ground-breaking paper on machine intelligence and Smale's list of problems for the twenty-first century.[4] Recently, Bastounis et al.[5] made remarkable progress in settling this theoretical issue. Their theory bears a major implication for Smale's 18th problem about the boundary of artificial intelligence (AI), especially deep learning as the current mainstream of AI. They show that it is in general non-computable to construct a neural network via loss minimization and apply it to testing data, and such a neural network is generally unstable. For example, there are in principle many classification problems for which "one may have 100% success rate on arbitrarily large training and validation datasets, and yet there are uncountably many points arbitrarily close to the training data for which the trained network will fail."[5]

Tomographic reconstruction is an important type of computational optimization problem, and, interestingly enough, deep networks for image reconstruction can and cannot be computed under different conditions. In the context of these inverse problems, the article by Antun et al. reported instabilities of deep reconstruction networks[2] due to the lack of kernel awareness.[6] Then, a comprehensive follow-up analysis by Antun et al.[7] established the boundary of deep-learning-inspired tomographic reconstruction, which helps address Smale's 18th problem. Among their contributions, the following three points are clearly made on (1) existence, (2) non-existence, and (3) the conditional existence of desirable networks. That is, while the existence of neural networks is proved in the literature for an excellent functional representation, the non-existence is proved of any algorithm that trains or computes such a neural network in a general setting. However, the conditional existence is also proved of such an algorithm to compute an accurate and stable network that solves meaningful inverse problems such as Fourier imaging from sparse data. Specifically, the existence of a network for a universal representation is well known (Theorem 2.1 in Antun et al.[7]), but how to train a network to achieve an accurate and stable approximation is a difficult issue. It has been shown that a counterexample can always be found in a general setting so that the accuracy and robustness of a network cannot be simultaneously obtained (Theorem 2.2 in Antun et al.[7]). On the other hand, under certain conditions, such as sparsity in levels, an accurate and stable network can be indeed obtained (Theorems 5.5 and 5.10 in Antun et al.[7]), with the FIRENET network as a good example.[7] At the core of the construction of FIRENET is kernel awareness. Clearly, training the network defined in subsection 5.1 in Antun et al.[7] cannot obtain kernel awareness and is subject to the phase transition of solutions to the inverse problems. In other words, if the difference between the two images lies close to the null space of the measurement matrix and is bounded from below, the Lipschitz constant of the inverse mapping can be very large, yielding a poor imaging performance. Fortunately, an algorithm can be used to utilize sparsity in levels and find a stable and accurate neural network (Theorems 5.5 and 5.10 in Antun et al.,[7] with uniform recovery guarantees, geometric convergence, and bounds on the number of samples and the number of layers of a network for a pre-specified accuracy).

In addition to the excellent work by Antun et al., active research efforts have been going on to develop deep networks for accurate and stable deep tomographic reconstruction. Representative results include the Learned Experts' Assessment-based Reconstruction Network(LEARN),[8] ItNet network,[9] Momentum-Net,[10] null-space network,[11] as well as deep equilibrium networks.[12]

In Chen et al.,[8] an iterative reconstruction algorithm in the CS framework was unrolled and trained in an end-to-end fashion. The experimental results from the resultant LEARN network on the Mayo Clinic low-dose computed tomography (CT) dataset are competitive with representative methods in terms of artifact reduction, feature preservation, and computational speed. In Genzel et al.,[9] an iterative deep-learning-based reconstruction network was designed to solve underdetermined inverse problems accurately and stably (ItNet shown in Figure 1 in Genzel et al.[9]). In comparison with total-variation minimization, their results reveal that standard end-to-end network architectures are resilient against not only statistical noise but also adversarial perturbations. In Chun et al.,[10] another iterative neural network, referred to as Momentum-Net, was prototyped by combining data-driven regression and model-based image reconstruction (MBIR). Momentum-Net is convergent under reasonable conditions (quadratic majorization via M-Lipschitz continuous gradients). Their results show that Momentum-Net outperformed MBIR and several other networks, but the effect of adversarial attacks on Momentum-Net was not evaluated. In Schwab et al.,[11] a null-space network was studied to offer a theoretical justification to deep learning-based tomographic reconstruction via so-called Φ-regularization. The convergence of the overall reconstruction workflow is proved, assuming a Lipschitz continuity and preserving the data consistency (illustrated in Figure 1 in Schwab et al.[11]). In Gilton et al.,[12] the deep equilibrium models were adapted to find the fixed point with guaranteed convergence under the ε-Lipschitz continuity. Subsequently, the trade-off can be made between reconstruction quality and computational cost.

In connection with the above results, our ACID network has significant merits and unique features. First, ACID is dedicated to overcoming the instabilities of neural networks on extensive datasets in Antun et al.[2] As a result, we have made a solid step forward along the direction of stabilizing deep reconstruction networks, showing that accurate and stable deep reconstruction is feasible, and remains an exciting research opportunity. Second, the ACID network is the first prototype that combines an established sparsity-oriented algorithm, a data-driven direct-reconstruction network, and an iterative data fidelity enforcement (for example, LEARN[8] and multi-domain integrative Swin transformer[13] ignore data consistency, ItNet network[9] lacks kernel awareness, Momentum-Net[10] and DRONE networks[14] miss a

learned mapping from data to images, null-space network[11] uses no sparsity, and deep equilibrium networks[12] focus only on the fixed point that does not imply image sparsity or data fidelity). Third, the converging behavior and solution characteristics of ACID have been analyzed under a reasonable assumption. The assumption is called the bounded relative error norm (BREN), which is a special case of a Lipschitz continuity. The Lipschitz continuity we used in our convergence analysis, which is practically interpreted as the BREN property and experimentally verified in our study, is consistent with the previous studies on non-convex optimization such as in the aforementioned network convergence analyses.[10–12] Furthermore, note that we do not request that the measurement matrix must satisfy a compressed sensing condition such as the restricted isometry property (RIP). This means that a standard sparsity-promotion algorithm may not give a unique solution. In this case, ACID promises to outperform the sparsity-minimization reconstruction alone, because data prior plays a significant role to fill in the gap in deep reconstruction. Last but not least, in addition to an accurate reconstruction performance, ACID has stability in two related aspects: (1) ACID can stabilize an unstable deep reconstruction network (by putting it in the ACID framework), and (2) ACID as a whole iterative procedure is resilient against adversarial attacks. Both aspects of the ACID stability are studied systematically in this work.

RESULTS

Our ACID architecture is heuristically obtained by minimizing an overall objective function. It is necessary to perform a convergent analysis for the iterative scheme to interpret the ACID algorithm. Although the following theoretical analysis is under several approximations, our findings do improve our understanding of the initially heuristically derived ACID scheme. It is underlined that there is no closed-form solution for the non-linear optimization problem, and a computationally efficient iterative formula is preferred for a stable solution. In the process, the errors will be suppressed via ACID iterations so that the ACID algorithm will converge to a desirable solution in the intersection of the space constrained by measured data, the space of sparse solutions, and the space of data-driven deep priors. This mechanism is similar to the conventional algebraic reconstruction technique (ART)/simultaneous algebraic reconstruction technique (SART) algorithm whose convergence was rigorously proved for convex optimization.[15–18]

Interpretation of ACID convergence

In the medical imaging field, a tomographic imaging task can be simplified to a system of linear equations: $\boldsymbol{p}^{(0)} = \boldsymbol{A}\boldsymbol{f}^{*} + \boldsymbol{e}$, where $\boldsymbol{A}\in\mathbb{R}^{m\times N}$ is a system matrix (for example, $\boldsymbol{A}$ is the Radon transform for CT[19] and the Fourier transform for MRI[20]), $\boldsymbol{p}^{(0)}\in\mathbb{R}^{m}$ is an original measurement data, $\boldsymbol{f}^{*}\in\mathbb{R}^{N}$ is the ground truth image of the object to be reconstructed, and $\boldsymbol{e}\in\mathbb{R}^{m}$ is data noise, $||\mathbf{e}||_2\leq\eta$ with a noise level $\eta\geq 0$. We focus on the few-view imaging for CT and sparse sampling for MRI. In this case, the column number of the system matrix is less than its row number, that is $m<N$, meaning that the inverse problem is underdetermined. For the under-deterministic problem, additional prior knowledge must be introduced to uniquely and stably recover the original image. Without loss of generality, we assume that $\boldsymbol{H}\in\mathbb{R}^{N\times N}$ is unitary, and $\boldsymbol{H}^{*}$ is the adjoint of $\boldsymbol{H}$. $\boldsymbol{A}\boldsymbol{H}^{*}$ satisfies the RIP of order s, and $\boldsymbol{H}\boldsymbol{f}^{*}$ is s-sparse. We further assume that the function $\Phi(\cdot)$ models a well-trained neural network, and it continuously maps measurement data to an image. Although $\Phi(\cdot)$ outputs an image $\boldsymbol{f}$ from the measurement, which is an inverse process of the linear system $\boldsymbol{p} = \mathbf{A}\boldsymbol{f}$, we have an approximate form: $\boldsymbol{A}\Phi(\boldsymbol{p})\cong\boldsymbol{p}$ in some sense such as satisfying the aforementioned BREN. Because the system matrix $\boldsymbol{A}$ is underdetermined and the neural network is unstable and may generate an artifact image, $\Phi(\mathbf{A}\boldsymbol{f}) = \boldsymbol{f}+\boldsymbol{f}^{ob}+\boldsymbol{f}^{nl}$, where $\boldsymbol{f}^{ob}$ is observable and $\boldsymbol{f}^{nl}$ is in the null space of $\boldsymbol{A}$. $\boldsymbol{f}^{nl}$ satisfies $\boldsymbol{A}\boldsymbol{f}^{nl} = 0$, and $\|\boldsymbol{A}\boldsymbol{f}^{ob}\|\neq 0$ if $\|\boldsymbol{f}^{ob}\|\neq 0$.

In this work, our goal is to design an iterative framework to stabilize an unstable neural network aided by a CS-based sparsity-promoting module. As an idealized setting to show the essential idea, we assume that the input to the neural network is dataset $\boldsymbol{p}^{(0)}$, and the output of the CS module is $\boldsymbol{f}$. Let us introduce a residual error $\boldsymbol{p}$ in the projection domain, that is $\boldsymbol{p}=\boldsymbol{p}^{(0)}-\mathbf{A}\boldsymbol{f}$ as a target of a correction mechanism. Then, we want to minimize the following objective function:

$$\underset{\boldsymbol{p},\boldsymbol{f}}{\operatorname{argmin}}\frac{1}{2}\|\Phi(\mathbf{A}\boldsymbol{f}+\boldsymbol{p})-\boldsymbol{f}\|_2^2+\frac{\lambda}{2}\|\boldsymbol{p}^{(0)}-\mathbf{A}\boldsymbol{f}-\boldsymbol{p}\|_2^2+\frac{\mu}{2}\|\boldsymbol{p}\|_2^2+\xi\|\boldsymbol{H}\boldsymbol{f}\|_1, \quad \text{(Equation 1)}$$

where $\lambda>0$, $\mu\geq 0$, and $\xi>0$ are hyperparameters, the first term is the difference between the outputs of the neural network and the CS-based sparsifying module, the second term is the measured noise energy, the third term is the residual error energy also in the projection domain, and the last term is to enforce the sparsity of the output image of the CS module, which is subject to the data-fidelity constraint in the projection domain. Let us define

$$L(\boldsymbol{p},\boldsymbol{f}) := \frac{1}{2}\|\Phi(\mathbf{A}\boldsymbol{f}+\boldsymbol{p})-\boldsymbol{f}\|_2^2+\frac{\lambda}{2}\|\boldsymbol{p}^{(0)}-\mathbf{A}\boldsymbol{f}-\boldsymbol{p}\|_2^2+\frac{\mu}{2}\|\boldsymbol{p}\|_2^2+\xi\|\boldsymbol{H}\boldsymbol{f}\|_1. \quad \text{(Equation 2)}$$

Then, we can use the block coordinate descent method[21] to optimize Equation 2 as follows:

$$\begin{cases}\boldsymbol{p}^{(\boldsymbol{k}+1)}=\underset{\boldsymbol{p}}{\operatorname{argmin}}\,L\left(\boldsymbol{p},\boldsymbol{f}^{(\boldsymbol{k})}\right)\\ \boldsymbol{f}^{(\boldsymbol{k}+1)}=\underset{\boldsymbol{f}}{\operatorname{argmin}}\,L\left(\boldsymbol{p}^{(\boldsymbol{k}+1)},\boldsymbol{f}\right)\end{cases}. \quad \text{(Equation 3)}$$

To update $\boldsymbol{p}$, we need to solve the following problem:

$$\boldsymbol{p}^{(\boldsymbol{k}+1)}=\underset{\boldsymbol{p}}{\operatorname{argmin}}\left(\frac{1}{2}\|\Phi\left(\mathbf{A}\boldsymbol{f}^{(\boldsymbol{k})}+\boldsymbol{p}\right)-\boldsymbol{f}^{(\boldsymbol{k})}\|_2^2+\frac{\lambda}{2}\|\boldsymbol{p}^{(0)}-\mathbf{A}\boldsymbol{f}^{(\boldsymbol{k})}-\boldsymbol{p}\|_2^2\right)+\frac{\mu}{2}\|\boldsymbol{p}\|_2^2. \quad \text{(Equation 4)}$$

Computing the partial derivative of the right side of (Equation 4), we have

$$\left(\frac{\partial\Phi\left(\mathbf{A}\boldsymbol{f}^{(\boldsymbol{k})}+\boldsymbol{p}\right)}{\partial\left(\mathbf{A}\boldsymbol{f}^{(\boldsymbol{k})}+\boldsymbol{p}\right)}\right)^{T}\left(\Phi\left(\mathbf{A}\boldsymbol{f}^{(\boldsymbol{k})}+\boldsymbol{p}\right)-\boldsymbol{f}^{(\boldsymbol{k})}\right)+\lambda\left(\mathbf{A}\boldsymbol{f}^{(\boldsymbol{k})}+\boldsymbol{p}-\boldsymbol{p}^{(0)}\right)+\mu\boldsymbol{p}=0. \quad \text{(Equation 5)}$$

Because the neural network is well-trained to solve the problem $\boldsymbol{Af} = \boldsymbol{p}$, we assume $\boldsymbol{A}\Phi(\boldsymbol{p}) \cong \boldsymbol{p}$ (at least on a training dataset). By performing derivative on both sides of $\boldsymbol{A}\Phi(\boldsymbol{p}) \cong \boldsymbol{p}$, we have $\boldsymbol{A}\left(\frac{\partial\Phi(\boldsymbol{p})}{\partial \boldsymbol{p}}\right) \cong \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix. This means $\frac{\partial\Phi(\boldsymbol{p})}{\partial \boldsymbol{p}} \cong \boldsymbol{A}^{-1}$ (in the sense of a pseudo-inverse for an underdetermined matrix $\boldsymbol{A}$ which can be obtained by classical methods such as truncated singular value decomposition [SVD]). In the classic and modern iterative CT reconstruction methods (e.g., SART), while $\boldsymbol{A}^T\boldsymbol{A} \neq \boldsymbol{I}$, the residual error correction mechanism and the resultant cumulative effect of the whole iterative process will make the final solution converge to an optimal solution for projections that are sufficiently sampled.[15,16] In this sense, treating a backprojection operator $\boldsymbol{A}^T$ as an approximate inverse to the projection operator $\boldsymbol{A}$ in each iteration is reasonable. Furthermore, in the ACID iterative framework, we also make the approximation $\boldsymbol{A}^T \cong \boldsymbol{A}^{-1}$. Hence, we have the approximation that $\frac{\partial\Phi(\boldsymbol{p})}{\partial \boldsymbol{p}} \cong \boldsymbol{A}^{-1} \cong \boldsymbol{A}^T$ and $\left(\frac{\partial\Phi(\boldsymbol{p})}{\partial(\boldsymbol{p})}\right)^T \cong \boldsymbol{A}$.

In Equation 5, $\left(\frac{\partial\Phi(\mathbf{Af}^{(\boldsymbol{k})}+\boldsymbol{p})}{\partial(\mathbf{Af}^{(\boldsymbol{k})}+\boldsymbol{p})}\right)^T \cong \boldsymbol{A}$ is the operator transforming a reconstructed image into a measurement dataset, and it is approximated as $\boldsymbol{A}$. By ignoring the observable artifact image from the neural network (since in the iterative correction, the artifact image will be gradually reduced; see the section "method details" for justification), we have $\boldsymbol{A}\Phi(\mathbf{Af}^{(\boldsymbol{k})}+\boldsymbol{p}) \cong \mathbf{Af}^{(\boldsymbol{k})}+\boldsymbol{p}$. Therefore, Equation 5 can be simplified as

$$\boldsymbol{p}^{(k+1)} \cong \frac{\lambda\left(\boldsymbol{p}^{(0)} - \mathbf{Af}^{(\boldsymbol{k})}\right)}{1+\lambda+\mu}. \qquad \text{(Equation 6)}$$

To update $\boldsymbol{f}$, we solve the following problem:

$$\boldsymbol{f}^{(\boldsymbol{k}+1)} = \underset{\boldsymbol{f}}{\operatorname{argmin}}\left(\frac{1}{2}\Phi\left\|\left(\mathbf{Af}+\boldsymbol{p}^{(\boldsymbol{k}+1)}\right) - \boldsymbol{f}\right\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{p}^{(0)} - \mathbf{Af} - \boldsymbol{p}^{(\boldsymbol{k}+1)}\right\|_2^2 + \xi\|\boldsymbol{Hf}\|_1\right). \qquad \text{(Equation 7)}$$

With $\bar{\boldsymbol{f}} = \boldsymbol{Hf}$ and $\boldsymbol{f} = \boldsymbol{H}^*\bar{\boldsymbol{f}}$, Equation 7 is rewritten as follows:

$$\bar{\boldsymbol{f}}^{(\boldsymbol{k}+1)} = \underset{\bar{\boldsymbol{f}}}{\operatorname{argmin}}\left(\frac{1}{2}\Phi\left\|\left(\mathbf{AH}^*\bar{\boldsymbol{f}}+\boldsymbol{p}^{(\boldsymbol{k}+1)}\right) - \boldsymbol{H}^*\bar{\boldsymbol{f}}\right\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{p}^{(0)} - \boldsymbol{AH}^*\bar{\boldsymbol{f}} - \boldsymbol{p}^{(\boldsymbol{k}+1)}\right\|_2^2 + \xi\|\bar{\boldsymbol{f}}\|_1\right). \qquad \text{(Equation 8)}$$

Computing the partial derivative of the right side of Equation 8, we have

$$\left(\boldsymbol{HA}^T\left(\frac{\partial\Phi\left(\mathbf{AH}^*\bar{\boldsymbol{f}}+\boldsymbol{p}^{(\boldsymbol{k}+1)}\right)}{\partial\left(\mathbf{AH}^*\bar{\boldsymbol{f}}+\boldsymbol{p}^{(\boldsymbol{k}+1)}\right)}\right)^T - \boldsymbol{H}\right)\left(\Phi\left(\mathbf{AH}^*\bar{\boldsymbol{f}}+\boldsymbol{p}^{(\boldsymbol{k}+1)}\right) - \boldsymbol{H}^*\bar{\boldsymbol{f}}\right) + \lambda\boldsymbol{HA}^T\left(\boldsymbol{AH}^*\bar{\boldsymbol{f}} - \boldsymbol{p}^{(0)} + \boldsymbol{p}^{(\boldsymbol{k}+1)}\right) + \xi sgn(\bar{\boldsymbol{f}}) = 0. \qquad \text{(Equation 9)}$$

Similarly treating $\left(\frac{\partial\Phi\left(\mathbf{AH}^*\bar{\boldsymbol{f}}+\boldsymbol{p}^{(\boldsymbol{k}+1)}\right)}{\partial\left(\mathbf{AH}^*\bar{\boldsymbol{f}}+\boldsymbol{p}^{(\boldsymbol{k}+1)}\right)}\right)^T$ as $\boldsymbol{A}$ and $\boldsymbol{A}^T \cong \boldsymbol{A}^{-1}$, Equation 9 can be simplified as

$$\lambda\bar{\boldsymbol{f}} + \lambda\boldsymbol{HA}^T\left(\boldsymbol{p}^{(\boldsymbol{k}+1)} - \mathbf{p}^{(0)}\right) + \xi sgn(\bar{\boldsymbol{f}}) \cong 0. \qquad \text{(Equation 10)}$$

From Equation 6, we have

$$\boldsymbol{p}^{(0)} \cong \frac{(1+\lambda+\mu)\boldsymbol{p}^{(k+1)}}{\lambda} + \boldsymbol{Af}^{(\boldsymbol{k})}. \qquad \text{(Equation 11)}$$

By substituting Equation 11 into Equation 10, we have

$$\lambda\bar{\boldsymbol{f}} - (1+\mu)\boldsymbol{HA}^T\boldsymbol{p}^{(k+1)} - \lambda\boldsymbol{Hf}^{(\boldsymbol{k})} + \xi sgn(\bar{\boldsymbol{f}}) \cong 0. \qquad \text{(Equation 12)}$$

Noting that

$$\boldsymbol{A}^T\boldsymbol{p}^{(k+1)} \cong \boldsymbol{A}^{-1}\boldsymbol{p}^{(k+1)} \cong \Phi\left(\boldsymbol{p}^{(\boldsymbol{k}+1)}\right), \qquad \text{(Equation 13)}$$

Equation 12 can be simplified as

$$\lambda\bar{\boldsymbol{f}} - (1+\mu)\boldsymbol{H}\Phi\left(\boldsymbol{p}^{(\boldsymbol{k}+1)}\right) - \lambda\boldsymbol{Hf}^{(\boldsymbol{k})} + \xi sgn(\bar{\boldsymbol{f}}) \cong 0. \qquad \text{(Equation 14)}$$

By rewriting Equation 14 as

$$\bar{\boldsymbol{f}} \cong \boldsymbol{Hf}^{(\boldsymbol{k})} + \frac{(1+\mu)}{\lambda}\boldsymbol{H}\Phi\left(\boldsymbol{p}^{(\boldsymbol{k}+1)}\right) - \frac{\xi}{\lambda}sgn(\bar{\boldsymbol{f}}) \qquad \text{(Equation 15)}$$

We have $\boldsymbol{f}^{(\boldsymbol{k}+1)}$ via soft-threshold filtering:

$$\boldsymbol{f}^{(\boldsymbol{k}+1)} \cong \boldsymbol{H}^*\boldsymbol{S}_{\frac{\xi}{\lambda}}\left(\boldsymbol{H}\left(\boldsymbol{f}^{(\boldsymbol{k})} + \frac{1+\mu}{\lambda}\Phi\left(\boldsymbol{p}^{(\boldsymbol{k}+1)}\right)\right)\right), \qquad \text{(Equation 16)}$$

where the soft-thresholding kernel is defined as

$$S_\varepsilon(x) = \begin{cases} 0, & |x|<\varepsilon \\ x - sgn(x)\varepsilon & otherwise \end{cases}. \qquad \text{(Equation 17)}$$

Combining Equations 6 and 16, we obtain a set of formulas:

$$\begin{cases} \boldsymbol{p}^{(k+1)} \cong \dfrac{\lambda\left(\mathbf{p}^{(0)} - \mathbf{Af}^{(\boldsymbol{k})}\right)}{1+\lambda+\mu} \\ \boldsymbol{f}^{(\boldsymbol{k}+1)} \cong \boldsymbol{H}^*\boldsymbol{S}_{\frac{\xi}{\lambda}}\left(\boldsymbol{H}\left(\boldsymbol{f}^{(\boldsymbol{k})} + \dfrac{1+\mu}{\lambda}\Phi\left(\boldsymbol{p}^{(\boldsymbol{k}+1)}\right)\right)\right) \end{cases}. \qquad \text{(Equation 18)}$$

Let us denote $\frac{\xi}{\lambda} = \varepsilon$ and simplify Equation 18 as

$$\begin{cases} \boldsymbol{p}^{(k+1)} \cong \dfrac{\lambda\left(\mathbf{p}^{(0)} - \mathbf{Af}^{(\boldsymbol{k})}\right)}{1+\lambda+\mu} \\ \boldsymbol{f}^{(\boldsymbol{k}+1)} \cong \boldsymbol{H}^*\boldsymbol{S}_{\varepsilon}\left(\boldsymbol{H}\left(\boldsymbol{f}^{(\boldsymbol{k})} + \dfrac{1+\mu}{\lambda}\Phi\left(\boldsymbol{p}^{(\boldsymbol{k}+1)}\right)\right)\right) \end{cases}. \qquad \text{(Equation 19)}$$

Clearly, Equation 19 agrees with our heuristically derived ACID network by setting $\mu = 0$. In other words, ACID is a special case of Equation 19 after the weighting parameters are properly selected.

Although a unitary property is assumed for the sparse transform $\boldsymbol{H}$ to obtain Equation 19, as is the case of the orthogonal wavelet decomposition, similar results can be also obtained in non-unitary cases. Poon[22] studied the problem of recovering a 1D or 2D discrete signal that is approximately sparse in its gradient transform from an incomplete subset of its Fourier coefficients. To obtain a high-quality reconstruction with high probability, robust to noise and stable to inexact gradient sparsity of order s, Poon proved that it is sufficient to draw $O(s\log N)$ of the available Fourier coefficients uniformly at random.[22] With Poon's results, we can extend Equation 19 to a non-unitary discrete gradient transform for total variation (TV) minimization.

Specifically, the term $\|\boldsymbol{Hf}\|_1$ in Equation 2 is specialized as a total-variation function $g(\boldsymbol{f})$ based on discrete gradient transform

$$g(\boldsymbol{f}) = \sum_{i_w=2}^{I_w}\sum_{i_h=2}^{I_h}(|\boldsymbol{f}(i_w,i_h)-\boldsymbol{f}(i_w-1,i_h)| + |\boldsymbol{f}(i_w,i_h)-\boldsymbol{f}(i_w,i_h-1)|), \quad \text{(Equation 20)}$$

where I_w and I_h represent the width and height of a reconstructed image, and the gradients on the image border are assumed to be zero. An FFT-based algorithm, FTVd,[23] can be employed to find the sparse solution for $\boldsymbol{f}$. Note that the generic TV favors piecewise constant regions, while high-order TV encourages piecewise polynomials.[23] Here, the input to the CS-based sparsifying module is normalized to [0, 1] to facilitate the selection of the regularized parameters, which requires de-normalization of the output of the CS module. In the CS framework, the robust null-space property ensures the stability of sparsity regularized recovery.[6,24]

Let us denote

$$\boldsymbol{f}^{\left(k+\frac{1}{2}\right)} = \boldsymbol{f}^{(k)} + \frac{1+\mu}{\lambda}\Phi\left(\boldsymbol{p}^{(k+1)}\right), \quad \text{(Equation 21)}$$

Equation 19 can be rewritten as

$$\begin{cases} \boldsymbol{p}^{(k+1)} \cong \dfrac{\lambda\left(\mathbf{p}^{(0)} - \mathbf{A}\boldsymbol{f}^{(\boldsymbol{k})}\right)}{1+\lambda+\mu} \\ \boldsymbol{f}^{\left(k+\frac{1}{2}\right)} = \boldsymbol{f}^{(\boldsymbol{k})} + \dfrac{1+\mu}{\lambda}\Phi\left(\boldsymbol{p}^{(\boldsymbol{k}+1)}\right) \\ \boldsymbol{f}^{(\boldsymbol{k}+1)} \cong \boldsymbol{H}^{*}\boldsymbol{S}_{\varepsilon}\left(\boldsymbol{H}\boldsymbol{f}^{\left(k+\frac{1}{2}\right)}\right) \end{cases}. \quad \text{(Equation 22)}$$

The discrete gradient function $g(\boldsymbol{f})$ defined in Equation 20 can be interpreted as $\|\mathbf{Hf}\|_1$ with a non-unitary transform matrix $\boldsymbol{H}$. Because $\boldsymbol{H}$ is not invertible, the adjoint matrix $\boldsymbol{H}^*$ in Equation 22 is not the inverse of $\boldsymbol{H}$. However, due to the fact that both $\boldsymbol{H}^*$ and $\boldsymbol{H}$ work as a pair before and after the soft-thresholding filtering in Equation 22, $\boldsymbol{H}^*$ can be interpreted as a pseudo-inverse of the discrete gradient transform $\boldsymbol{H}$.[25] Hence, each pixel of $\boldsymbol{f}^{(k+1)}$ at the position (i_w, i_h) in Equation 22 can be expressed as follows[25]:

$$\begin{aligned} \boldsymbol{f}^{(k+1)}(i_w,i_h) = \frac{1}{4}\Bigg(& S_{\varepsilon}^{-1}\left(\boldsymbol{f}^{\left(k+\frac{1}{2}\right)}(i_w,i_h), \boldsymbol{f}^{\left(k+\frac{1}{2}\right)}(i_w+1,i_h)\right) \\ & + S_{\varepsilon}^{-1}\left(\boldsymbol{f}^{\left(k+\frac{1}{2}\right)}(i_w,i_h), \boldsymbol{f}^{\left(k+\frac{1}{2}\right)}(i_w,i_h+1)\right) \\ & + S_{\varepsilon}^{-1}\left(\boldsymbol{f}^{\left(k+\frac{1}{2}\right)}(i_w,i_h), \boldsymbol{f}^{\left(k+\frac{1}{2}\right)}(i_w-1,i_h)\right) \\ & + S_{\varepsilon}^{-1}\left(\boldsymbol{f}^{\left(k+\frac{1}{2}\right)}(i_w,i_h), \boldsymbol{f}^{\left(k+\frac{1}{2}\right)}(i_w,i_h-1)\right)\Bigg), \end{aligned} \quad \text{(Equation 23)}$$

where $S_{\varepsilon}^{-1}(\bullet,\bullet)$ is the pseudo-inverse of the soft-thresholding kernel $S_{\varepsilon}(x)$ for a given threshold ε. The pseudo-inverse $S_{\varepsilon}^{-1}(\bullet,\bullet)$ is defined as[25]:

$$S_{\varepsilon}^{-1}(v_a,v_b) = \begin{cases} \dfrac{v_a+v_b}{2}, & \text{if } |v_a-v_b| \le \varepsilon \\ v_a - \dfrac{\varepsilon}{2}, & \text{if } v_a - v_b > \varepsilon \\ v_a + \dfrac{\varepsilon}{2}, & \text{if } v_a - v_b < -\varepsilon \end{cases}. \quad \text{(Equation 24)}$$

with the pseudo-inverse (Equation 24), although the discrete gradient transform is neither unitary nor invertible, the iterative framework (Equation 19) can still be applied for TV minimization using a compressed sensing technique.[22]

Under practically reasonable conditions such as noisy and insufficient data, the ACID iteration will converge to a feasible solution subject to an uncertain range proportional to the noise level (see the convergence analysis in the section "method details").

BREN property

Our theoretical analysis requires the following BREN property of a reconstruction neural network to reconstruct $\boldsymbol{f}$ from measurement $\boldsymbol{p} = \boldsymbol{Af}^*$. If a reconstruction network satisfies the BREN property, we call it a well-designed and well-trained reconstruction network, or a proper network.

Definition: a reconstruction network has the BREN property if the ratio between the L_2 norm of the reconstruction error and the L_2 norm of the corresponding ground truth is less than $(1-\sigma)$ with $0<\sigma<1$. For an s-sparse observable image $\boldsymbol{f}^*$, there are different ways to formulate the Lipschitz continuity such as our BREN property. Let us assume that the function $\Phi(\bullet)$ models a well-trained neural network. Denote the output of the neural network $\Phi(\boldsymbol{Af}^*) = \boldsymbol{f}^* + \boldsymbol{f}^{ob} + \boldsymbol{f}^{nl}$, where the second and third terms are observable and null-space components of the error image associated with the ground truth image $\boldsymbol{f}^*$ and the measurement matrix $\boldsymbol{A}$, the BREN property is defined as

$$\frac{\|\Phi(\mathbf{Af}^*) - \boldsymbol{f}^*\|}{\|\boldsymbol{f}^*\|} = \frac{\|\boldsymbol{f}^{ob} + \boldsymbol{f}^{nl}\|}{\|\boldsymbol{f}^*\|} \le (1-\sigma). \quad \text{(Equation 25)}$$

Equation 25 implies that $\|\boldsymbol{f}^{ob} + \boldsymbol{f}^{nl}\| \le (1-\sigma)\|\boldsymbol{f}^*\|$.

Remark 1: in the literature of deep imaging, including the paper on instabilities of deep reconstruction,[2] a reconstruction network, even if it is unstable, will still produce an output not too far from the ground truth in the sense of the BREN property. The involved errors of types I and II have significant clinical impacts but the norm of these errors in combination is assumed to be small relative to that of the underlying image. This is how a proper reconstruction network is defined and commonly expected in practice. For example, the most popular loss function of a reconstruction network is in the L_2 norm so that a reconstructed image should be close to the ground truth in the sense of the L_2 norm without an adversarial attack. Furthermore, in the adversarial attack cases of our interest, the BREN property is assumed to be valid as the condition for our convergence analysis below.

Table 1. BREN ratios (%) associated with different reconstruction networks

Methods	r_1	r_2	r_3	r_4
Med-50	2.90	x	x	x
AUTOMAP	10.39	23.09	47.85	85.86
Deep MRI	2.73	8.03	13.28	x
MRI-VN	3.53	x	x	x

Remark 2: for deep reconstruction in the supervised mode, a training dataset is typically in the format of $(\boldsymbol{p}(i), \boldsymbol{f}(i))$, $i = 1, 2, \cdots, I_{trn}$. We assume that the imaging model is linear, and we can augment the training dataset to $(\alpha\boldsymbol{p}(i), \alpha\boldsymbol{f}(i))$, $i = 1, 2, \cdots, I_{trn}$, where α is any constant within a reasonable range. With the augmented data, the network will map the input of a small norm to an output of a proportionally small norm. Alternatively, we can include the normalization layer(s) in the reconstruction network so that the network performance is insensitive to the magnitude of data and images.

Remark 3: our assumption of the BREN property is needed for our convergence analysis below, just like the case for CS theory where RIP/rNSP is required for unique image recovery. If the requirement is not met, the theoretical arguments below will not be valid. We have shown that our BREN ratio is substantially less than 1 for the datasets in this PNAS study.[2]

Specifically, all the experimental results with perturbations were repeated in the CT and MRI cases[26–29] reported in Antun et al.[2] Then, the BREN ratios were computed using different reconstruction networks with various perturbations. It is found that all these ratios in CT and MRI experiments are substantially less than 1. As shown in Table 1 and Figures 1, 2, 3, and 4, the AUTOMAP seems more sensitive to the perturbations; i.e., small perturbations cause large changes in the sense of the L_2-norm. Clearly, the BREN property is satisfied in this context. It is easy to observe that the perturbed images contain artifacts; for example, the MED-50, AUTOMAP, and MRI-VN results. In these cases, the sparsity of reconstructed images was corrupted, and the feedforward data estimation based on these reconstruction results are usually not consistent with the original measurement. Searching for a feasible solution within the space of sparse solution is central to the traditional iterative reconstruction. Furthermore, the ACID searches for a reconstruction good in the three aspects: image sparsity, big-data-driven prior, and iterative calibration to eliminate unexplained residual data. When the final image satisfies all these three constraints, it will be our best possible solution.

Lipschitz convergence with perturbations

Let the combination of the measurement matrix $\boldsymbol{A}$ and the neural network $\Phi(\cdot)$ be $\Phi_{\boldsymbol{A}}(\cdot)$. According to the definition of the Lipschitz constant, if we employ the L_2 norm, the Lipschitz constant is the minimal constant that holds for the following inequality:

$$\|\Phi_{\boldsymbol{A}}(\boldsymbol{f}) - \Phi_{\boldsymbol{A}}(\boldsymbol{f}')\| \leq L\|f - f'\|. \quad \text{(Equation 26)}$$

For each fixed $\boldsymbol{f}$, we generated a series of perturbations to obtain $\boldsymbol{f}'$, and computed the value of the ratio $\|\Phi_{\boldsymbol{A}}(\boldsymbol{f}) - \Phi_{\boldsymbol{A}}(\boldsymbol{f}')\|/\|\boldsymbol{f} - \boldsymbol{f}'\|$. Specifically, we computed for many images and found the upper and lower bounds of the Lipschitz constant as our empirically estimated ranges in the CT and MRI cases, respectively. Note that the authors of AUTOMAP did not provide the original data and code for sufficient training and testing, we only performed this experiment on the DAGAN and Ell-50. Here, only 500 pairs of ellipse phantoms were used for Ell-50. Each pair contains $\boldsymbol{f}$ and $\boldsymbol{f}'$, where $\boldsymbol{f}'$ was generated by adding an adversarial attack on the clear image $\boldsymbol{f}$ using the aforementioned adversarial method. Specifically, the lower and upper bounds in the Ell-50 case are 0.4674 and 0.6424, respectively. In contrast, 14,866 pairs of MRI images were used to determine the lower and upper bounds in the DAGAN case, and the corresponding lower and upper bounds are 1.4854 and 12.0737, respectively.

We had shown the convergence of ACID with respect to PSNR in Part A of our work,[3] and here we show the convergence of ACID in terms of the Lipschitz constant with respect to the number of iterations. As representative examples, the convergence curves in the C3 and M4 cases (see the supplemental information of Wu et al.[3] for more details) are given in Figure 5. It can be observed that the Lipschitz constant of ACID for both CT and MRI are monotonically decreasing and finally converge to a constant scale.

ACID against noise

Although some examples about the insensitivity of ACID against noise are reported in Part A of Wu et al.,[3] here we followed up the study in Koonjoo et al.[30] and performed a similar local stability test on ACID with DAGAN and Ell-50 respectively built in. This local robustness was assessed using the maximum ratio between variations in the output space and variations in the imaging object space: $\|\Phi_{\boldsymbol{A}}(\boldsymbol{f}) - \Phi_{\boldsymbol{A}}(\boldsymbol{f}')\|/\|\boldsymbol{f} - \boldsymbol{f}'\|$ for two adjacent images $\boldsymbol{f}$ and $\boldsymbol{f}'$. In this test, 500 pairs of CT phantoms were selected from the Ell-50 test dataset. Then, the additive white Gaussian noise was added, with zero mean and standard deviation 11–30 HU. In this way, we obtained 500 cases. Furthermore, 14,866 image pairs were chosen from the MRI dataset. Similarly, the additive white Gaussian noise was randomly added to each of these images to generate $\boldsymbol{f}'$. A maximum output-input variation ratio of 3.023 was observed for these noisy inputs. The histograms in the CT and MRI cases are given in Figure 6. The results empirically demonstrate the local stability of the ACID reconstruction against noise.

DISCUSSION

Kernel awareness and network stability

The kernel awareness is an important concept. When a reconstruction algorithm lacks the kernel awareness, a "cardinal crime" ("cardinal sin") could be committed,[6] which implies that a well-trained network model would potentially produce highly unstable results, defeating the purpose of medical imaging. In that scenario, the trained network would produce significantly different images from essentially identical input datasets, between which there are subtle differences representing invisible perturbations.

Specifically, a deep network is trained on a dataset D using an optimization technique. The learning procedure would normally

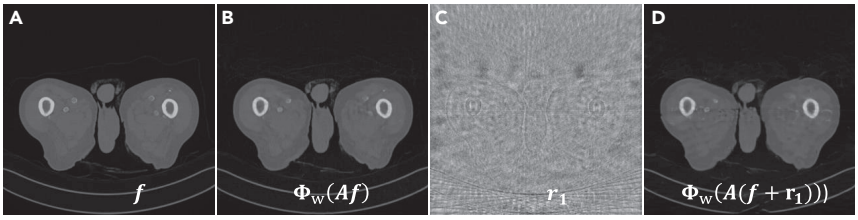

Figure 1. Reconstruction results using MED-50 from Antun et al.

The first to fourth images represent the original image, MED-50 result without perturbation, perturbation image, and MED-50 result with perturbation, respectively.

converge to a network model with optimized parameters, which is usually a continuous transform T such that for $\boldsymbol{f} \in D$,

$$\|T(\boldsymbol{Af}) - \boldsymbol{f}\| < \tau, \qquad \text{(Equation 27)}$$

where $\|\cdot\|$ is a suitable norm, $\boldsymbol{A}$ is a measurement matrix, and a constant τ is a bound. To evaluate the stability of the network model, an ε-Lipschitz metric is defined as follows:

$$L^{\varepsilon}(T,\boldsymbol{p}) = \sup_{0<\|\boldsymbol{p}-\boldsymbol{p}'\|<\varepsilon} \frac{\|T(\boldsymbol{p}) - T(\boldsymbol{p}')\|}{\|\boldsymbol{p} - \boldsymbol{p}'\|} \qquad \text{(Equation 28)}$$

A formula can be derived for a lower bound of the ε-Lipschitz index estimation for $\boldsymbol{p} = \boldsymbol{Af}$:

$$L^{\varepsilon}(T,\boldsymbol{p}) > \frac{1}{\varepsilon}(\|\boldsymbol{f}' - \boldsymbol{f}\| - 2\tau). \qquad \text{(Equation 29)}$$

In fact,

$$\begin{aligned} L^{\varepsilon}(T,\boldsymbol{p}) &\geq \frac{\|T(\boldsymbol{Af}') - T(\boldsymbol{Af})\|}{\|\boldsymbol{Af}' - \boldsymbol{Af}\|} \\ &\geq \frac{\|\boldsymbol{f}' - \boldsymbol{f}\| - \|T(\boldsymbol{Af}) - \boldsymbol{f}\| - \|T(\boldsymbol{Af}') - \boldsymbol{f}'\|}{\|\boldsymbol{Af}' - \boldsymbol{Af}\|} \qquad \text{(Equation 30)} \\ &> \frac{1}{\varepsilon}(\|\boldsymbol{f}' - \boldsymbol{f}\| - 2\tau), \end{aligned}$$

for $\varepsilon \geq \tau$. An inverse problem, such as few-view CT and sparse MRI, involves solving $\boldsymbol{Af} = \boldsymbol{p} + \varepsilon$, where $\boldsymbol{A}$ is an $m \times N$ matrix, $m < N$, and ε is measurement noise. Clearly, the transform $\boldsymbol{A}$ would have a null space (kernel) with $\dim(Null(\boldsymbol{A})) > 1$. Then, there is a nonzero vector $\boldsymbol{f}_0 \in neighborofNull(\boldsymbol{A})$ and a scale factor σ for a large number L such that

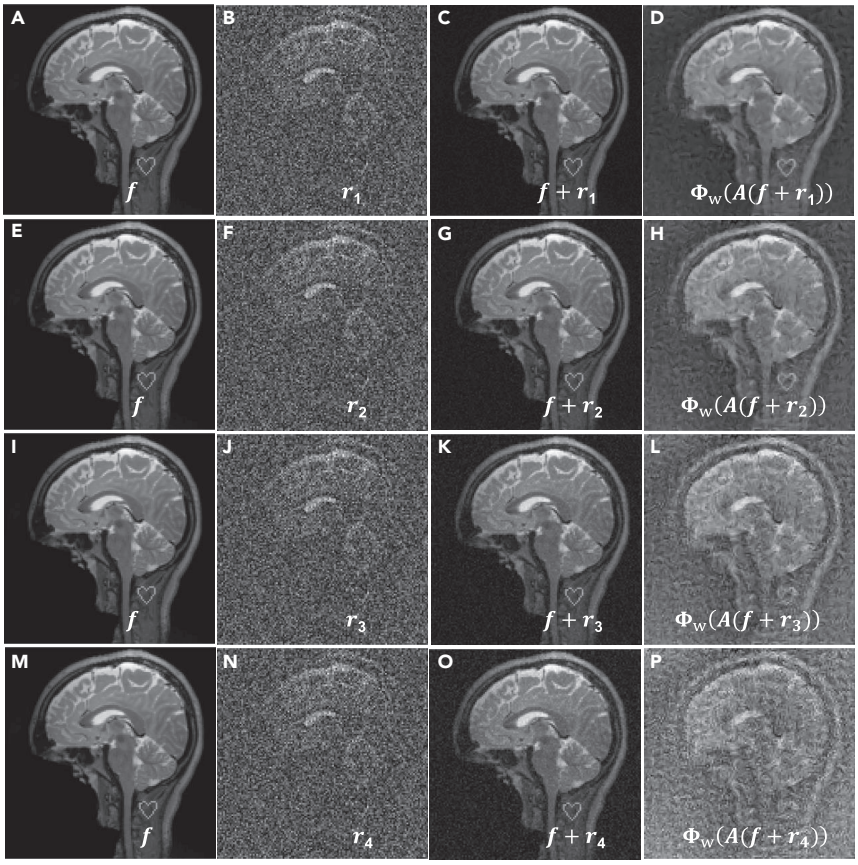

Figure 2. Reconstruction results using AUTOMAP from Antun et al.

The first to fourth columns represent the original, perturbation, original plus perturbation, and perturbed AUTOMAP images, respectively. The first to fourth rows represent different strengths of perturbation, where $r_{1_F}^2 < r_{2_F}^2 < r_{3_F}^2 < r_{4_F}^2$.

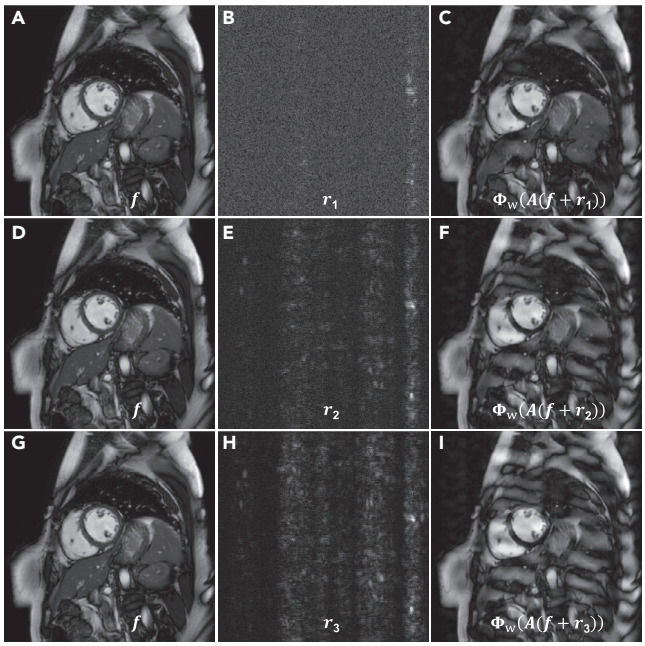

Figure 3. Reconstruction results using deep MRI

These results were adapted from Antun et al. The first-third columns represent the original, perturbation, and perturbed deep MRI (DM) results, respectively. The first to third rows present different strengths of perturbation, where $r_1^2_F < r_2^2_F < r_3^2_F$.

$$\|\boldsymbol{A}\boldsymbol{f}_0\| < \varepsilon, \|\boldsymbol{A}\boldsymbol{f}' - \boldsymbol{p}\| < \varepsilon, \text{ and } \|\boldsymbol{f}' - \boldsymbol{f}\| > L + 2\tau, \quad \text{(Equation 31)}$$

where $\boldsymbol{f}' = \boldsymbol{f} + \sigma \boldsymbol{f}_0$ with $\boldsymbol{f} \in (Null(\boldsymbol{A}))^{\perp}$. If the training set has at least two such elements $(\boldsymbol{A}\boldsymbol{f}, \boldsymbol{f})$ and $(\boldsymbol{A}\boldsymbol{f}', \boldsymbol{f}')$, we have

$$L^{\varepsilon}(T, \boldsymbol{p}) > \frac{L}{\varepsilon}. \quad \text{(Equation 32)}$$

From Equation 32, the instability is intrinsic; that is, when input data are very close to the null space of the associated imaging operator and $\boldsymbol{p}$ is slightly perturbed, a large variation would be induced in the reconstructed image. The instability of the trained network would yield artifacts in reconstructed images, subject to either false-positive or false-negative diagnosis.

BREN and Lipschitz continuity

Assuming the BREN property, our analysis (see the section "method details" and Figures 7–9) shows that ACID is stable against adversarial attacks. In fact, BREN can be viewed as a special case of the Lipschitz continuity; i.e., they are consistent.

Let us first define measurement and reconstruction operators $\mathcal{M}$ and $\mathcal{R}$ on two metric spaces $(\boldsymbol{F}, d_F)$ and $(\boldsymbol{P}, d_{\boldsymbol{P}})$, respectively. Let us measure an image $\boldsymbol{f} \in \boldsymbol{F}$ tomographically to obtain a measurement $\boldsymbol{p} \in \boldsymbol{P}$. We assume that each image in $\boldsymbol{F}$ is non-trivial in that sense that $\boldsymbol{f} > 0$. Let us denote the measurement operator $\mathcal{M} : \boldsymbol{F} \to \boldsymbol{P}$; that is $\boldsymbol{p} = \mathcal{M}(\boldsymbol{f})$. Suppose that the measurement process is totally transparent to us. Thus, we know a 1-to-1 correspondence $\boldsymbol{P} \leftrightarrow \boldsymbol{F}$ perfectly. For example, in our case, the measurement matrix $\boldsymbol{A}$ satisfies the RIP of order s, $\boldsymbol{f}$ is s-sparse, $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{f}$, and there exists a 1-to-1 map. Then, we can define the ideal

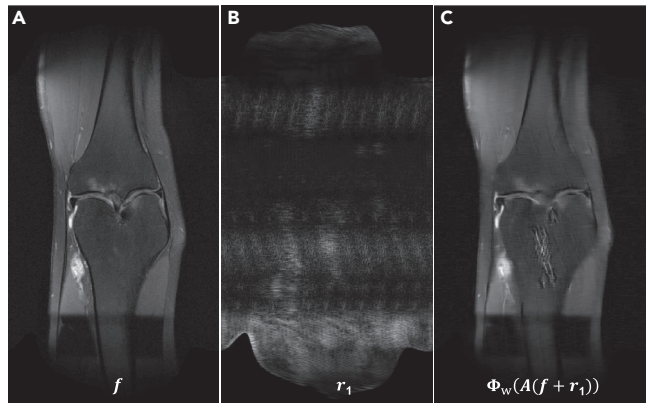

Figure 4. Reconstruction results using MRI-VN from Antun et al.

The first to third images represent the original, perturbation, and perturbed MRI-VN results, respectively.

reconstruction operator $\boldsymbol{f} = \mathcal{R}(\boldsymbol{p})$. Reasonably, we assume that the operator $\mathcal{R}$ is a Lipschitz continuous (LC) function $\mathcal{R} : \boldsymbol{P} \to \boldsymbol{F}$ that satisfies $d_F(\mathcal{R}(\boldsymbol{p}_1), \mathcal{R}(\boldsymbol{p}_2)) \le L_1 d_F(\boldsymbol{f}_1, \boldsymbol{f}_2)$ for a constant $L_1 > 0$.

With a big dataset, we can train a deep network Φ to approximate the ideal reconstruction operator $\mathcal{R}$, $\Phi : \boldsymbol{P} \to \boldsymbol{F}$ is an LC function that satisfies $d_F(\Phi(\boldsymbol{p}_1), \Phi(\boldsymbol{p}_2)) \le L_2 d_F(\boldsymbol{f}_1, \boldsymbol{f}_2)$ for a constant $L_2 > 0$. Furthermore, we assume that network Φ is well designed and well trained so that, for a training tomographic dataset, we have $\Phi(\boldsymbol{p}(i)) - \mathcal{R}(\boldsymbol{p}(i)) < \delta_n$, $i = 1, 2, \cdots, I_{trn}$. For a new dataset $\boldsymbol{p}'$ from an underlying image $\boldsymbol{f}'$, the BREN property requires that $\frac{\|\Phi(\boldsymbol{p}') - \mathcal{R}(\boldsymbol{p}')\|}{\|\mathcal{R}(\boldsymbol{p}')\|} = \frac{\|\Phi(\boldsymbol{A}\boldsymbol{f}') - \boldsymbol{f}'\|}{\|\boldsymbol{f}'\|} \le 1 - \sigma$. Suppose that the image $\boldsymbol{f}'$ is close to an image $\boldsymbol{f}(i_0)$ in the training dataset. In this setting, we have the following relations:

$$\|\Phi(\boldsymbol{p}(i_0)) - \mathcal{R}(\boldsymbol{p}(i_0))\| < \delta_n, \quad \text{(Equation 33)}$$

$$d_F(\mathcal{R}(\boldsymbol{p}(i_0)), \mathcal{R}(\boldsymbol{p}')) \le L_1 d_F(\boldsymbol{f}(i_0), \boldsymbol{f}'), \quad \text{(Equation 34)}$$

$$d_F(\Phi(\boldsymbol{p}(i_0)), \Phi(\boldsymbol{p}')) \le L_2 d_F(\boldsymbol{f}(i_0), \boldsymbol{f}'), \quad \text{(Equation 35)}$$

where Equation 33 is due to the fact the network is well designed and well trained, and Equation 34 and Equation 35 are due to the Lipschitz continuity of $\mathcal{R}$ and Φ. Therefore, we have

$$\begin{aligned}\|\Phi(\boldsymbol{p}') - \mathcal{R}(\boldsymbol{p}')\| = {} & \|\Phi(\boldsymbol{p}') - \Phi(\boldsymbol{p}(i_0)) + \Phi(\boldsymbol{p}(i_0)) - \mathcal{R}(\boldsymbol{p}(i_0)) \\ & + \mathcal{R}(\boldsymbol{p}(i_0)) - \mathcal{R}(\boldsymbol{p}')\| \le \|\Phi(\boldsymbol{p}') - \Phi(\boldsymbol{p}(i_0))\| \\ & + \|\Phi(\boldsymbol{p}(i_0)) - \mathcal{R}(\boldsymbol{p}(i_0))\| + \|\mathcal{R}(\boldsymbol{p}(i_0)) \\ & - \mathcal{R}(\boldsymbol{p}')\| < L_1 d_F(\boldsymbol{f}(i_0), \boldsymbol{f}') + L_2 d_F(\boldsymbol{f}(i_0), \boldsymbol{f}') + \delta_n.\end{aligned} \quad \text{(Equation 36)}$$

Therefore, under the condition that

$$\frac{L_1 d_F(\boldsymbol{f}(i_0), \boldsymbol{f}') + L_2 d_F(\boldsymbol{f}(i_0), \boldsymbol{f}') + \delta_n}{\|\boldsymbol{f}'\|} < 1, \quad \text{(Equation 37)}$$

we have the BREN. The condition can be simplified to $\frac{(L_{1+} + L_2) d_F(\boldsymbol{f}(i_0), \boldsymbol{f}') + \delta_n}{\|\boldsymbol{f}'\|} < 1$, which is roughly $\frac{(L_{1+} + L_2) d_F(\boldsymbol{f}(i_0), \boldsymbol{f}')}{\|\boldsymbol{f}'\|} < 1$. That is, as long as an image is fairly close to the training dataset, the BREN property is satisfied. Heuristically, if the image norm

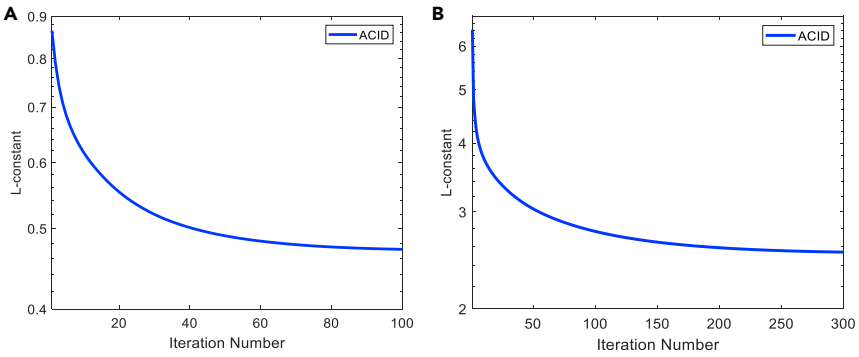

Figure 5. Convergence curves of ACID in terms of the Lipschitz constant

(A and B) The convergence curves of ACID with respect to the number of iterations in the C3 and M4 cases, respectively.

is greater than the product of the LC constant $(L_{1+} + L_2)$ and the distance between an image to be reconstructed and its closest reference point, we have the BREN property. For a big dataset, $d_F(\boldsymbol{f}(i_0), \boldsymbol{f}')$ is small, so that L can be large, which is especially true if we interpret $\boldsymbol{F}$ and $\boldsymbol{P}$ as appropriate low-dimensional manifolds.

Because there is a 1-to-1 correspondence $\boldsymbol{P} \leftrightarrow \boldsymbol{F}$ perfectly, we can treat the combination of the measurement matrix $\boldsymbol{A}$ and the neural network Φ as a new LC function $\Phi_{\boldsymbol{A}}$, which satisfies

$$d_F(\Phi_{\boldsymbol{A}}(\boldsymbol{f}_1), \Phi_{\boldsymbol{A}}(\boldsymbol{f}_2)) \leq L d_F(\boldsymbol{f}_1, \boldsymbol{f}_2). \quad \text{(Equation 38)}$$

The Lipschitz continuity assumption is useful to assess the convergence of a deep reconstruction algorithm. In the section "results", we have verified the BREN property for the data used in Antun et al.[2] Those results support the practical relevance of the BREN property. More importantly, one can calculate the Lipschitz constant directly for both the MRI and CT data using Equation 38.

Unlike the establishment of the instabilities, it is mathematically insufficient to prove the general applicability of ACID using only a finite number of positive experimental results. Hence, a theoretical analysis is desirable on the convergence of the ACID iteration. Although a thorough characterization is rather challenging (since the field of non-convex optimization is still in its infancy), we have assumed an experimentally motivated BREN property of the reconstruction network, which is a special case of the Lipschitz continuity that is widely used for non-convex optimization to establish various converging properties.[31,32] The BREN property means that the relative error of a deep network-based reconstruction is under control in a L_2 norm. Based on BREN, we have made an initial effort to understand the converging mechanism of the ACID iteration. Specifically, we have provided not only (1) a heuristic analysis based on the simplification that the CS module allows a perfect sparsification but also (2) a mathematically denser analysis of the convergence under two approximations (the first approximation to invert an underdetermined system matrix $\boldsymbol{A}$, and the other is to minimize TV with a non-unitary transform $\boldsymbol{H}$).

EXPERIMENTAL PROCEDURES

Resource availability

Lead contact

Hengyong Yu, PhD (e-mail: Hengyong-yu@ieee.org).

Materials availability

The study did not generate new unique reagents.

Data and code availability

The codes, trained networks, test datasets, and reconstruction results are publicly available on Zenode (https://zenodo.org/record/5497811).

Method details

Adversarial attacks to a selected network

In the image reconstruction field, the continuous imaging system[33,34] can be discretized into a linear model $\boldsymbol{p} = \boldsymbol{Af}$, where $\boldsymbol{A} \in R^{m \times N}$ is the system matrix, $\boldsymbol{p}$ represents collected data, and m and N defines the size of the system matrix $\boldsymbol{A}$. The aim of image reconstruction is to reconstruct $\boldsymbol{f}$ from $\boldsymbol{p}$ for a given system matrix $\boldsymbol{A}$. To assess the stability of image reconstruction, it is necessary to compute a tiny perturbation or adversarial attack.[6,35,36] In this context, Antun et al.[2] first computed a tiny perturbation

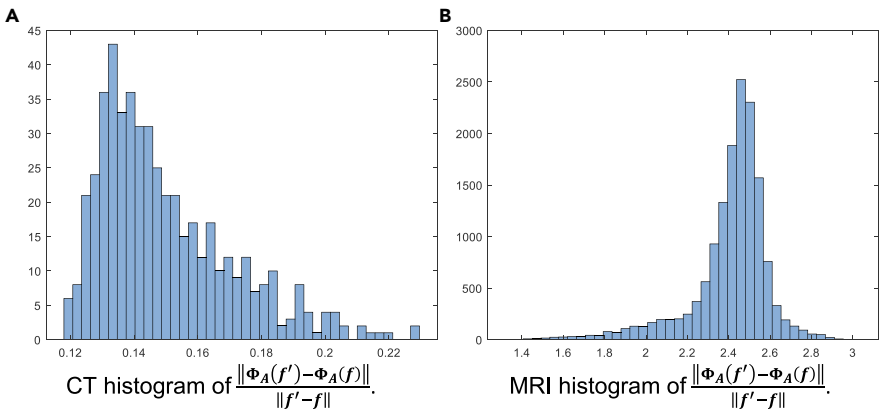

Figure 6. ACID is locally stable with respect to noise

(A) The histogram of the output-to-input ratio between noise-free and Gaussian input data, where ACID has EII-50 built-in, giving the maximum value of 0.229.
(B) The histogram of the output-to-input ratio between noise-free and Gaussian input data, where ACID has DAGAN embedded, with the maximum ratio of 3.023.

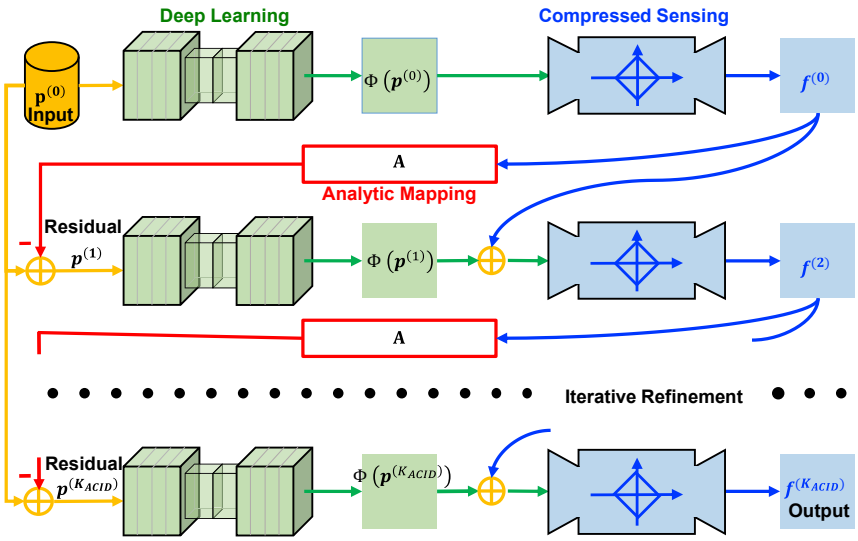

Figure 7. ACID architecture for stabilizing deep tomographic image reconstruction

ACID consists of the following components: deep reconstruction, compressed sensing-based sparsity-promotion, analytic mapping, and iterative refinement. $\boldsymbol{p}^{(0)}$ is original tomographic data, and $\boldsymbol{p}^{(k_{ACID})}$, k_{ACID} = 1, 2, 3, …, K_{ACID}, represents an estimated residual dataset in the k_{ACID}^{th} iteration between $\boldsymbol{p}^{(0)}$ and the currently reconstructed counterpart. $\Phi(\boldsymbol{p}^{(k_{ACID})})$ is an output of the deep reconstruction module, and $\boldsymbol{f}^{(k_{ACID})}$ represents the image after compressed sensing-based regularization.

$$\text{Reconstruct } \boldsymbol{f} \text{ from } \boldsymbol{p}=\boldsymbol{A}\boldsymbol{f},\ \boldsymbol{A}\in R^{m\times N}. \quad \text{(Equation 39)}$$

For a well-trained neural network $\Phi: R^m \rightarrow R^N$ to solve Equation 39, similar to the adversarial attack in image classification,[37] we compute the instabilities by formulating the following problem[2]

$$\widehat{\boldsymbol{e}}(\boldsymbol{f})\in \underset{e}{\operatorname{argmin}}\|\mathrm{e}\|, \quad \text{s.t. } \|\Phi(\boldsymbol{p}+\boldsymbol{A}\boldsymbol{e})-\Phi(\boldsymbol{p})\|\geq\varepsilon. \quad \text{(Equation 40)}$$

With Equation 40, when $\varepsilon>0$, the relationship $\boldsymbol{p}=\boldsymbol{A}\boldsymbol{f}$ might not hold. One can consider the constrained Lasso variant of Equation 40 as follows:

$$\widehat{\boldsymbol{e}}(\boldsymbol{f})\in \underset{e}{\operatorname{argmax}}\|\Phi(\boldsymbol{p}+\boldsymbol{A}\boldsymbol{e})-\Phi(\boldsymbol{p})\|, \quad \text{s.t. } \|\boldsymbol{e}\|\leq\sigma. \quad \text{(Equation 41)}$$

There is no infeasibility issue for Equation 41. An unconstrained Lasso inspired version of Equation 41 is given by

$$\boldsymbol{e}^*(\boldsymbol{p})\in \underset{e}{\operatorname{argmax}}\frac{1}{2}\|\Phi(\boldsymbol{p}+\boldsymbol{A}\boldsymbol{e})-\Phi(\boldsymbol{p})\|_2^2-\frac{\gamma}{2}\|\boldsymbol{e}\|_2^2. \quad \text{(Equation 42)}$$

With $\Phi(\boldsymbol{p})=l(\boldsymbol{f})$, Equation 42 is further converted to

$$\boldsymbol{e}^*(\boldsymbol{p})\in \underset{e}{\operatorname{argmax}}\frac{1}{2}\|\Phi(\boldsymbol{p}+\boldsymbol{A}\boldsymbol{e})-l(\boldsymbol{f})\|_2^2-\frac{\gamma}{2}\|\boldsymbol{e}\|_2^2, \quad \text{(Equation 43)}$$

where $l(\boldsymbol{f})=\boldsymbol{f}$ for image-domain post-processing[38–40] and $l(\boldsymbol{f})=\Phi(\boldsymbol{A}\boldsymbol{f})$ with the end-to-end network (such as AUTOMAP[27] and iRadonMap[41]). Note that Equation 43 works in the image domain to find perturbations. One generates a reconstructed image using an easy way and then compares the original image with a perturbed one to determine whether the perturbed image is acceptable/unacceptable. Now, we describe the details on how to generate perturbations for a single neural network.

Since the neural network $\Phi: R^m \rightarrow R^N$ is a non-linear function. It is difficult to search for a global maximum for Equation 43. Here, we use the same strategy as in Antun et al.[2] to search for tiny perturbations. In other words, one usually can reach the local maxima of Equation 43 using a gradient search method. Especially, one defines the following objective function:

$$D'_{\mathbf{p}}(\boldsymbol{e})=\frac{1}{2}\|\Phi(\boldsymbol{p}+\boldsymbol{A}\boldsymbol{e})-l(\boldsymbol{f})\|_2^2-\frac{\gamma}{2}\|\boldsymbol{e}\|_2^2, \quad \text{(Equation 44)}$$

Regarding the optimization of Equation 44, the gradient ascent search is a very common method.[42] See the supplemental information for details of the algorithm implementation.

Adversarial attacks to ACID as whole

The iterative process of ACID is to find the optimized solution in the intersection of (1) the space of data-driven priors, (2) the space of sparse images, and (3) the space of solutions satisfying the measurement, as shown in Figure 7. With a tiny perturbation to our proposed ACID workflow, the feedforward propagation of the perturbation is illustrated in Figure 8. Specifically, the formula of Equation 44 is converted to

$$D'_{\mathbf{p}^{(0)}}(\boldsymbol{e})=\frac{1}{2}\|\widehat{\boldsymbol{f}}(\boldsymbol{p},\boldsymbol{p}^{(0)})-l(\boldsymbol{f})\|_2^2-\frac{\gamma}{2}\|\boldsymbol{e}\|_2^2, \quad \text{(Equation 45)}$$

where $\widehat{\boldsymbol{f}}$ is different from $\boldsymbol{f}$ as computed by the neural network and stabilized in the ACID framework. $\widehat{\boldsymbol{f}}(\boldsymbol{p},\boldsymbol{p}^{(0)})$ is the solution minimizing the following objective function (Equation 1).

For the optimization problem (Equation 1), we now compute a tiny perturbation via gradient ascent search. Specifically, we compute

$$D'_{\mathbf{p}^{(0)}+\boldsymbol{A}\mathbf{e}}(\mathbf{e})=\frac{1}{2}\|\widehat{\boldsymbol{f}}(\boldsymbol{p},\boldsymbol{p}^{(0)}+\boldsymbol{A}\boldsymbol{e})-\boldsymbol{f}\|_2^2-\frac{\gamma}{2}\|\boldsymbol{e}\|_2^2. \quad \text{(Equation 46)}$$

The backpropagation process for ACID is shown in Figure 9. More clearly, we define the cost function of ACID as

$$L_c=\frac{1}{2}\|\boldsymbol{f}_e-\boldsymbol{f}\|_2^2-\frac{\gamma}{2}\|\boldsymbol{e}\|_2^2, \quad \text{(Equation 47)}$$

where $\boldsymbol{e}$ is the perturbation, $\boldsymbol{f}_e=\widehat{\boldsymbol{f}}(\boldsymbol{p},\boldsymbol{p}^{(0)}+\boldsymbol{A}\boldsymbol{e})$ is the output of the ACID system with the perturbation $\boldsymbol{e}$, and $\boldsymbol{f}$ is the corresponding output without $\boldsymbol{e}$. To find an effective $\boldsymbol{e}$, we need to compute the gradient $dL_c/d\boldsymbol{e}$, and then refine the perturbation $\boldsymbol{e}$ using a gradient ascent algorithm. For clarity, the iteration index for ACID is changed to k_{ACID} ($k_{ACID}=0,\ldots,K_{ACID}$) in this subsection. In Figure 9, there are two branches contributing to $\boldsymbol{f}^{(k_{ACID})}$($k_{ACID}$ = 1, …, K_{ACID} − 1); i.e., branches 1 and 2. To compute the gradient $dL_c/d\boldsymbol{e}$, we take both branches into account.

Because $dL_{\mathrm{ACID}}/d\boldsymbol{e}$ with the loss function $L_{\mathrm{ACID}}=\frac{1}{2}\|\boldsymbol{f}_e-\boldsymbol{f}\|_2^2$ is complicated, we cannot directly compute the gradient of L_c. Fortunately, $dL_c/d\boldsymbol{e}$ can be solved using the backpropagation algorithm,[43,44] which is commonly used in deep learning.[45,46] Then, $dL_c/d\boldsymbol{e}$ can be split as

$$\frac{dL_c}{d\boldsymbol{e}}=\frac{d\left(\frac{1}{2}\|\boldsymbol{f}_e-\boldsymbol{f}\|_2^2\right)}{d\boldsymbol{e}}-\gamma\boldsymbol{e}=\frac{d(L_{ACID})}{d\boldsymbol{e}}-\gamma\mathbf{e}. \quad \text{(Equation 48)}$$

Now, let us start with the backpropagation process for ACID, as shown in Figure 9. First, we can decompose the ACID system into the three modules keyed to $\boldsymbol{p}$, $\boldsymbol{u}$, and $\boldsymbol{f}$ respectively, where $\boldsymbol{u}=\Phi(\boldsymbol{p})$, and the whole procedure is shown in Figure 9. The input and the output of $\boldsymbol{p}$, $\boldsymbol{u}$, and $\boldsymbol{f}$ are denoted as $\boldsymbol{p}_i$, $\boldsymbol{u}_i$, $\boldsymbol{f}_i$, $\boldsymbol{p}_o$, $\boldsymbol{u}_o$, and $\boldsymbol{f}_o$, respectively. Also, the gradient of $\boldsymbol{p}$, $\boldsymbol{u}$, and $\boldsymbol{f}$ can be denoted as $\frac{d\boldsymbol{p}_o}{d\boldsymbol{p}_i}$, $\frac{d\boldsymbol{u}_o}{d\boldsymbol{u}_i}$ and $\frac{d\boldsymbol{f}_o}{d\boldsymbol{f}_i}$, respectively.

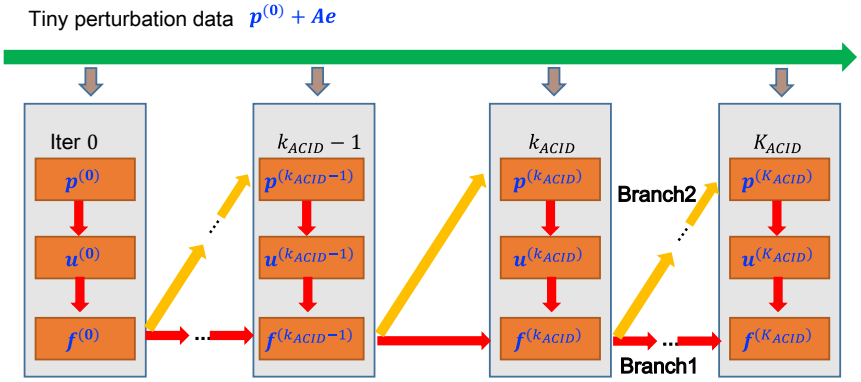

Figure 8. Feedforward propagation of adversarial data in the ACID framework

Following the same steps as for the selected network, we will use the gradient ascent method to iteratively compute adversarial attacks for the whole ACID system, and the target to be attacked will be changed from a single unstable neural network to our whole ACID workflow. There are two iterative loops: the outer loop is for gradient ascent search, and the inner loop is for ACID feedforward and backpropagation. The stopping criteria of finding an adversarial attack for the whole ACID include (1) the number of iterations reaches the maximum number of iterations for computing an adversarial attack (AA) denoted as K_{AA}; or (2) the noise strength of the adversarial attack is greater than that used in attacking the single neural network recorded in our study in terms of the L_2-norm. As mentioned above, the maximum number of iterations of the inner loop is K_{ACID} for the ACID feedforward process. Because each whole inner loop can be considered as an intermediate node, we can use the idea of backpropagation to search for a desirable perturbation. See the supplemental information for details of the algorithm implementation.

Heuristic analysis on the ACID convergence

Let $\boldsymbol{p}^{(0)}$ denote measured data, which is generally incomplete, inconsistent, and noisy. Specifically, the data can be sinogram or k-space data. Then, we need the three key functions in the ACID scheme. First, an imaging model $\boldsymbol{A}$ is the forward model from an underlying image to tomographic data, which is assumed to be linear without loss of generality. Second, the recon-net $\Phi(\cdot)$ consists of a data-enhancement sub-net and a direct-reconstruction sub-net. This network may be unstable. Note that even if the recon-net is unstable, we assume that it respects the BREN property for our convergence analysis. Third, the CS module Θ can be a standard CS algorithm or a network-version of the CS algorithm. This module is an image post-processor that maps an image reconstructed by the recon-net to a refined image within the space of sparse solutions. The loss function of the CS module can be a weighted sum of the fidelity term and the sparsity term. The fidelity term can be in the L_2 norm of the difference of the input and output images. Let k be the index for iteration, $k = 0, 1, 2, \ldots$, and we define the following variables.

$\boldsymbol{d}^{(k)}$: tomographic image produced by the recon-net $\Phi(\cdot)$. $\boldsymbol{d}^{(0)} = \Phi(\boldsymbol{p}^{(0)})$, which is assumed to be a good initial image based on the BREN assumption. $\boldsymbol{d}^{(k)}$, $k = 1, 2, \ldots$ represents successive refinements to $\boldsymbol{d}^{(0)}$.

$\boldsymbol{f}^{(k)}$: tomographic image refined by the CS module Θ, $k = 0, 1, 2, \ldots$, which should be in the space of sparse solutions in the CS framework, and may or may not be the ground truth, depending on if RIP/rNSP is satisfied or not.

$\boldsymbol{m}^{(k)}$: estimated data based on the output of the CS module, $k = 0, 1, 2, \ldots$, which should eventually become as close as possible to the measured data $\boldsymbol{p}^{(0)}$.

$\boldsymbol{p}^{(k)}$: unexplained residual errors based on $\boldsymbol{f}^{(k-1)}$ in reference to the measured data $\boldsymbol{p}^{(0)}$, $k = 1, 2, \ldots$. Specifically, this residual is defined as $\boldsymbol{p}^{(k)} = \boldsymbol{p}^{(0)} - \boldsymbol{A}\boldsymbol{f}^{(k-1)}$. This data residual will be small when k is sufficiently large to obtain a good image quality.

Now, let us analyze the first cycle of the ACID workflow in the following steps.

The first step is to generate $\boldsymbol{d}^{(0)}$ from the original data $\boldsymbol{p}^{(0)}$, which is done by the recon-net $\Phi(\cdot) : \boldsymbol{d}^{(0)} = \Phi(\boldsymbol{p}^{(0)})$. Since the recon-net $\Phi(\cdot)$ may be unstable, $\boldsymbol{d}^{(0)}$ can be generally decomposed into the following three components: (1) $\boldsymbol{f}^{*}$, the ground truth image, which is assumed to be s-sparse; (2) $\boldsymbol{f}^{(sps,0)}$, artifacts in the space of sparse solutions of the CS module Θ, which cannot be eliminated based on the sparsity consideration; and (3) $\boldsymbol{f}^{(nsps,0)}$, artifacts not in the space of sparse solution that can be suppressed by the CS module Θ. That is, $\boldsymbol{d}^{(0)} = \boldsymbol{f}^{*} + \boldsymbol{f}^{(sps,0)} + \boldsymbol{f}^{(nsps,0)}$.

Then, $\boldsymbol{d}^{(0)}$ is processed by the CS module Θ to obtain $\boldsymbol{f}^{(0)}$. That is, $\boldsymbol{f}^{(0)} = \Theta(\boldsymbol{d}^{(0)})$ so that the difference between $\boldsymbol{d}^{(0)}$ and $\boldsymbol{f}^{(0)}$ is minimized subject to that $\boldsymbol{f}^{(0)}$ is in the space of sparse solutions of Θ under the constraint of the measurement. As a result, $\boldsymbol{f}^{(0)} = \boldsymbol{f}^{*} + \boldsymbol{f}^{(sps,0)}$ (without loss of generality, here we assume that the sparsity can be perfectly achieved). Without loss of generality, let us take CT as an example.

Based on $\boldsymbol{f}^{(0)}$, $\boldsymbol{m}^{(0)}$ can be estimated with the forward imaging model as $\boldsymbol{m}^{(0)} = \boldsymbol{A}\boldsymbol{f}^{(0)}$. Generally, we consider $\boldsymbol{f}^{(sps,0)} = \boldsymbol{f}^{(ob,0)} + \boldsymbol{f}^{(nl,0)}$ where two components $\boldsymbol{f}^{(ob,0)}$ and $\boldsymbol{f}^{(nl,0)}$ are observable and unobservable, respectively (an unobservable image $\boldsymbol{f}^{(nl,0)}$ is in the null space of $\boldsymbol{A}$). When $\boldsymbol{f}^{(ob,0)}$ is nonzero, the estimated data and the measured data $\boldsymbol{p}^{(0)}$ must be inconsistent. This discrepancy is quantified as the data residual $\boldsymbol{p}^{(1)} = \boldsymbol{p}^{(0)} - \boldsymbol{A}\boldsymbol{f}^{(0)}$. When $\boldsymbol{A}$ does not satisfy RIP/rNSP, the intersection of the data constrained space and the data prior space may contain many solutions, and thus it could be possible that $\boldsymbol{p}^{(1)} = 0$ but $\boldsymbol{f}^{(0)} = \boldsymbol{f}^{*} + \boldsymbol{f}^{(nl,0)}$ (that is, $\boldsymbol{f}^{*} + \boldsymbol{f}^{(nl,0)}$ and $\boldsymbol{f}^{*}$ explain the

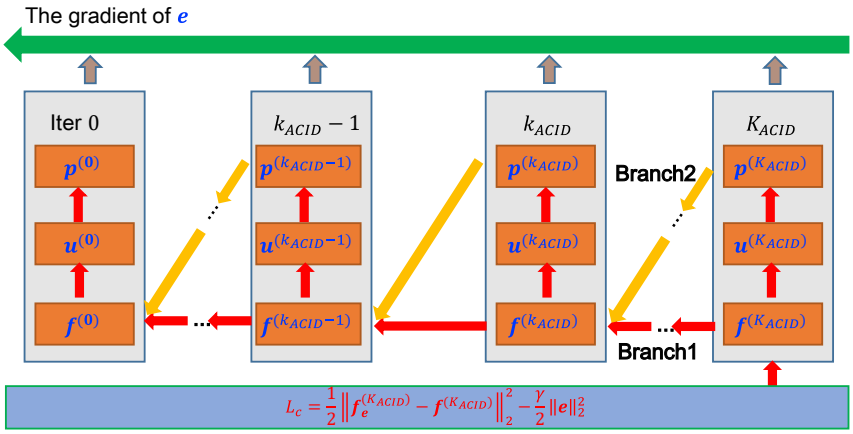

Figure 9. Backpropagation process of ACID

data $\boldsymbol{p}^{(0)}$ equally well). Nevertheless, it is highly unlikely in practice that the residual data become zero, and the ACID iterative process will not converge immediately.

The nonzero data residual can be further reconstructed into the image increment $\Delta\boldsymbol{f}^{(1)}$ using the recon-net $\Phi(\cdot)$; that is, $\Delta\boldsymbol{f}^{(1)} = \Phi(\boldsymbol{p}^{(1)})$. Then, the current tomographic image is updated to $\boldsymbol{d}^{(1)} = \boldsymbol{f}^{(0)} + \Delta\boldsymbol{f}^{(1)}$ (the sum of two prior-consistent images are assumed to be still consistent with data-driven prior, which can be alternatively achieved by applying the recon-net to the augmented data $\boldsymbol{p}^{(0)} + \boldsymbol{p}^{(k)}$). Generally speaking, $\boldsymbol{d}^{(1)}$ will be closer to the ground truth $\boldsymbol{f}^*$ than the previous image $\boldsymbol{d}^{(0)}$, since our reconstructed image should explain as much as possible data $\boldsymbol{p}^{(0)}$. With $\boldsymbol{d}^{(1)}$, the residual error in the data domain will be reduced.

Now, we can describe the converging mechanism of ACID, assuming that the recon-net $\Phi(\cdot)$ satisfies the BREN property. Our key arguments include the following three steps:

1. After we have $\boldsymbol{f}^{(0)} = \Theta(\Phi(\boldsymbol{p}^{(0)}))$, by BREN we have $\frac{\|\boldsymbol{f}^{(sps,0)} + \boldsymbol{f}^{(nsps,0)}\|}{\|\boldsymbol{f}^*\|} < 1$. That is, $\frac{\|\boldsymbol{f}^{(ob,0)}\|}{\|\boldsymbol{f}^*\|} = \alpha^{(0)} < 1$ and $\frac{\|\boldsymbol{f}^{(nl,0)}\|}{\|\boldsymbol{f}^*\|} = \beta^{(0)} < 1$, because $\boldsymbol{f}^{(sps,0)} = \boldsymbol{f}^{(ob,0)} + \boldsymbol{f}^{(nl,0)}$ is orthogonal to $\boldsymbol{f}^{(nsps,0)}$, and $\boldsymbol{f}^{(ob,0)}$ is orthogonal to $\boldsymbol{f}^{(nl,0)}$. That is, both $\|\boldsymbol{f}^{(ob,0)}\|$ and $\|\boldsymbol{f}^{(nl,0)}\|$ are fractions of $\|\boldsymbol{f}^*\|$.
2. We use the forward model $\boldsymbol{A}$ to synthesize the unexplained residual data $\boldsymbol{p}^{(1)}$. Then, $\boldsymbol{p}^{(1)}$ is fed to the recon-net to reconstruct $\Delta\boldsymbol{f}^{(1)} = \Phi(\boldsymbol{p}^{(1)})$. Since $\boldsymbol{p}^{(1)}$ is due to $\boldsymbol{f}^{(ob,0)}$. Then, $-\boldsymbol{f}^{(ob,0)}$ can be reconstructed up to a new artifact image $\boldsymbol{f}^{(ob,1)} + \boldsymbol{f}^{(nl,1)} + \boldsymbol{f}^{(nsps,1)}$. By BREN again, $\frac{\|\boldsymbol{f}^{(ob,1)}\|}{\|\boldsymbol{f}^{(ob,0)}\|} = \alpha^{(1)} < 1$ and $\frac{\|\boldsymbol{f}^{(nl,1)}\|}{\|\boldsymbol{f}^{(ob,0)}\|} = \beta^{(1)} < 1$. That is, both $\|\boldsymbol{f}^{(ob,1)}\|$ and $\|\boldsymbol{f}^{(nl,1)}\|$ are fractions of $\|\boldsymbol{f}^{(ob,0)}\|$.
3. We can repeat this process for $k \to \infty$, we have that $\boldsymbol{d}^{(\infty)} = \boldsymbol{f}^{(\infty)} = \boldsymbol{f}^* + \boldsymbol{f}^{(ob,\infty)} + \sum_{k=0}^{\infty} \boldsymbol{f}^{(nl,k)}$. Because the norm of $\boldsymbol{f}^{(ob,\infty)}$ is less than $\|\boldsymbol{f}^*\| \prod_{k=0}^{\infty} \alpha^{(k)} \to 0$, $\boldsymbol{f}^{(ob,\infty)} \to 0$. Meanwhile $\boldsymbol{f}^{(nl,\infty)} \to 0$ since

$$\|\boldsymbol{f}^{(nl,k)}\| = \|\boldsymbol{f}^{(ob,k-1)}\|\beta^{(k)} = \|\boldsymbol{f}^*\| \prod_{k'=0}^{k-1} \alpha^{(k')}\beta^{(k)}.$$

Noting that both $\alpha^{(k)}$ and $\beta^{(k)}$ are smaller than 1, we have

$$\|\sum_{k=0}^{\infty} \boldsymbol{f}^{(nl,k)}\| < \sum_{k=0}^{\infty} \|\boldsymbol{f}^{(nl,k)}\| = \sum_{k=0}^{\infty} \|\boldsymbol{f}^*\| \prod_{k'=0}^{k-1} \alpha^{(k')}\beta^{(k)} < \infty.$$

That is, the ACID will converge to $\boldsymbol{f}^* + \sum_{k=0}^{\infty} \boldsymbol{f}^{(nl,k)}$, which will be in the intersection of the space of solutions satisfying measured data, the space of sparse solutions, and the space of data-driven solutions. While this ACID scheme may converge to an image still containing a nonzero null-space component when RIP/rNSP is not satisfied, the key point is that under the same condition (i.e., RIP/rNSP is not satisfied) a sparsity-promoting algorithm cannot eliminate such a nonzero null space component either, and more importantly ACID has enforced the powerful deep prior so that the space of feasible solutions is greatly reduced relative to that permitted with a sparsity-promoting algorithm alone. In our experiments, we have shown that ACID with the kernel awareness embedded consistently outperforms the selected sparsity-promoting algorithms that do not utilize big-data-driven prior. In other words, the data prior is instrumental in recovering the nonzero null-space component that cannot be measured by the system matrix.

Although the above analysis is not mathematically rigorous, it indeed sheds light on the inner working of ACID. This analysis adds value, especially in the current situation that a general non-convex optimization theory is yet to be developed. In the above analysis, we have assumed the BREN property of the recon-net. As a result, even if the network is not ideal (which means producing a substantial nonzero artifact image), the convergence is still guaranteed, as long as the relative error is under control (less than 100%) in the L_2 norm, which is a practically motivated condition. On the other hand, it is underlined that if the network is indeed optimized or nearly optimized so that artifact image is small in the first place, the iterative process will converge rapidly, and in that case the whole ACID workflow can be unrolled into a compact feedforward network.

Mathematical analysis on the ACID convergence

In the theoretical iterative framework (Equation 19) and with the BREN property of the neural network, we will show that the final solution $\boldsymbol{f}^{(k+1)}$ will converge to an optimal image, and particularly the ground truth $\boldsymbol{f}^*$ assuming that RIP/rNSP is satisfied, subject to a noise-induced uncertainty distance in terms of the L_2 norm and the null-space component. While this convergence analysis is not rigorous, it helps rationalize the ACID workflow, and in this context the convergence to the optimal solution implies stability.

Now, let us analyze the convergence of our ACID scheme. Denoting $M_1 = \frac{\lambda}{1+\lambda+\mu}$ and $M_2 = \frac{1+\mu}{\lambda}$, we have $M = M_1 M_2 = \frac{\lambda}{1+\lambda+\mu}\frac{1+\mu}{\lambda} = \frac{1+\mu}{1+\lambda+\mu} < 1$. Equation 19 can be simplified to our heuristically designed ACID iteration:

$$\begin{cases} \boldsymbol{p}^{(k+1)} = M_1\left(\boldsymbol{p}^{(0)} - \boldsymbol{A}\boldsymbol{f}^{(k)}\right) \\ \boldsymbol{f}^{(k+1)} = \boldsymbol{H}^* \boldsymbol{S}_\varepsilon\left(\boldsymbol{H}\left(\boldsymbol{f}^{(k)} + M_2\Phi(\boldsymbol{p}^{(k+1)})\right)\right) \end{cases}. \quad \text{(Equation 49)}$$

In this subsection, we replace $\cong$ in Equation 19 with $=$ in Equation 49, abusing the notation a bit. Let us analyze the convergence of our ACID network for noise-free measurement as follows. Assuming an initial image $\Phi(\boldsymbol{p}^{(0)}) = \boldsymbol{f}^* + \boldsymbol{f}^{(ob,0)} + \boldsymbol{f}^{(nl,0)}$. By the BREN property, we have

$$\|\boldsymbol{f}^{(ob,0)} + \boldsymbol{f}^{(nl,0)}\| < (1-\sigma)\|\boldsymbol{f}^*\|. \quad \text{(Equation 50)}$$

Since $\boldsymbol{H}\boldsymbol{f}^{(ob,0)}$ and $\boldsymbol{H}\boldsymbol{f}^{(nl,0)}$ are orthogonal, we have

$$\|\boldsymbol{f}^{(ob,0)} + \boldsymbol{f}^{(nl,0)}\| = \|\boldsymbol{H}\boldsymbol{f}^{(ob,0)} + \boldsymbol{H}\boldsymbol{f}^{(nl,0)}\| = \|\boldsymbol{H}\boldsymbol{f}^{(ob,0)}\| + \|\boldsymbol{H}\boldsymbol{f}^{(nl,0)}\| = \|\boldsymbol{f}^{(ob,0)}\| + \|\boldsymbol{f}^{(nl,0)}\|. \quad \text{(Equation 51)}$$

This implies that

$$\|\boldsymbol{f}^{(ob,0)}\| < (1-\sigma)\|\boldsymbol{f}^*\|. \quad \text{(Equation 52)}$$

Since $\boldsymbol{f}^{(0)}$ is the output of the soft-thresholding filtering, it can be expressed as

$$\boldsymbol{f}^{(0)} = \Phi(\boldsymbol{p}^{(0)}) - \boldsymbol{H}^*\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}, \quad \text{(Equation 53)}$$

where $\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}$ is a noise background in the transform domain. If we denote $\bar{f}_n^{(\boldsymbol{e},0)}$ as the n^{th} component of $\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}$, there will be $\left|\bar{f}_n^{(\boldsymbol{e},0)}\right| \le \varepsilon$, which is a noise floor. Without loss of generality, in the transform domain we assume the first s components span the s-sparse space of $\boldsymbol{H}\boldsymbol{f}^*$. Because only the first s components of $\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}$ is observable, let us decompose $\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}$ into two parts $(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)})_{\boldsymbol{n}\le s}$ and $(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)})_{\boldsymbol{n}>s}$, where $(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)})_{\boldsymbol{n}\le s}$ is observable and $(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)})_{\boldsymbol{n}>s}$ is in the null space of $\boldsymbol{A}$. Then, Equation 53 can be rewritten as

$$\boldsymbol{f}^{(0)} = \boldsymbol{f}^* + \boldsymbol{f}^{(ob,0)} - \boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}\right)_{\boldsymbol{n}\le s} + \boldsymbol{g}^{(nl,0)}, \quad \text{(Equation 54)}$$

where $\boldsymbol{g}^{(nl,0)} = \boldsymbol{f}^{(nl,0)} - \boldsymbol{H}^*(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)})_{\boldsymbol{n}>s}$ is in the null space of $\boldsymbol{A}$.

For the case $k = 0$:

From Equations 49 and 54, we have

$$\boldsymbol{p}^{(1)} = M_1\left(\boldsymbol{p}^{(0)} - \boldsymbol{A}\boldsymbol{f}^{(0)}\right) = -M_1\boldsymbol{A}\boldsymbol{f}^{(ob,0)} + M_1\boldsymbol{A}\boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}\right)_{\boldsymbol{n}\le s}, \quad \text{(Equation 55)}$$

$$\Phi(\boldsymbol{p}^{(1)}) = -M_1\boldsymbol{f}^{(ob,0)} + M_1\boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}\right)_{\boldsymbol{n}\le s} + \boldsymbol{f}^{(ob,1)} + \boldsymbol{f}^{(nl,1)}, \quad \text{(Equation 56)}$$

$$\|\boldsymbol{f}^{(ob,1)}\| < (1-\sigma)\|M_1\boldsymbol{f}^{(ob,0)} - M_1\boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}\right)_{\boldsymbol{n}\le s}\|, \quad \text{(Equation 57)}$$

$$\boldsymbol{f}^{(1)} = \boldsymbol{f}^{(0)} + M_2\Phi(\boldsymbol{p}^{(1)}) - \boldsymbol{H}^*\bar{\boldsymbol{f}}^{(\boldsymbol{e},1)} = \boldsymbol{f}^* + (1-M)\left(\boldsymbol{f}^{(ob,0)} - \boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}\right)_{\boldsymbol{n}\le s}\right) + M_2\boldsymbol{f}^{(ob,1)} - \boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},1)}\right)_{\boldsymbol{n}\le s} + \boldsymbol{g}^{(nl,1)}, \quad \text{(Equation 58)}$$

where $\boldsymbol{g}^{(nl,1)} = \boldsymbol{g}^{(nl,0)} + M_2(\boldsymbol{f}^{(nl,1)}) - \boldsymbol{H}^*(\bar{\boldsymbol{f}}^{(\boldsymbol{e},1)})_{\boldsymbol{n}>s}$ is in the null space of $\boldsymbol{A}$.

For the case $k = 1$:

From Equations 49 and 58, we have

$$\boldsymbol{p}^{(2)} = M_1\left(\boldsymbol{p}^{(0)} - \boldsymbol{A}\boldsymbol{f}^{(1)}\right) = -(1-M)M_1\boldsymbol{A}\left(\boldsymbol{f}^{(ob,0)} - \boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}\right)_{\boldsymbol{n}\le s}\right) - M\boldsymbol{A}\boldsymbol{f}^{(ob,1)} + M_1\boldsymbol{A}\boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},1)}\right)_{\boldsymbol{n}\le s}, \quad \text{(Equation 59)}$$

$$\Phi\left(\boldsymbol{p}^{(2)}\right) = -(1-M)M_1\left(\boldsymbol{f}^{(ob,0)} - \boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}\right)_{\boldsymbol{n}\le s}\right) - M\boldsymbol{f}^{(ob,1)} + M_1\boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},1)}\right)_{\boldsymbol{n}\le s} + \boldsymbol{f}^{(ob,2)} + \boldsymbol{f}^{(nl,2)}, \quad \text{(Equation 60)}$$

$$\|\boldsymbol{f}^{(ob,2)}\| < (1-\sigma)\|(1-M)M_1\left(\boldsymbol{f}^{(ob,0)} - \boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}\right)_{\boldsymbol{n}\le s}\right) + M\boldsymbol{f}^{(ob,1)} - M_1\boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},1)}\right)_{\boldsymbol{n}\le s}\|, \quad \text{(Equation 61)}$$

$$\boldsymbol{f}^{(2)} = \boldsymbol{f}^{(1)} + M_2\Phi\left(\boldsymbol{p}^{(2)}\right) - \boldsymbol{H}^*\bar{\boldsymbol{f}}^{(\boldsymbol{e},2)} = \boldsymbol{f}^* + (1-M)^2\left(\boldsymbol{f}^{(ob,0)} - \boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}\right)_{\boldsymbol{n}\le s}\right) + (1-M)\left(M_2\boldsymbol{f}^{(ob,1)} - \boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},1)}\right)_{\boldsymbol{n}\le s}\right) + M_2\boldsymbol{f}^{(ob,2)} - \boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},2)}\right)_{\boldsymbol{n}\le s} + \boldsymbol{g}^{(nl,2)}, \quad \text{(Equation 62)}$$

with $\boldsymbol{g}^{(nl,2)} = \boldsymbol{g}^{(nl,1)} + M_2\boldsymbol{f}^{(nl,2)} - \boldsymbol{H}^*(\bar{\boldsymbol{f}}^{(\boldsymbol{e},2)})_{\boldsymbol{n}>s}$.

If we continue the above procedure, for $k>1$, it is easy to obtain that

$$\boldsymbol{p}^{(\boldsymbol{k}+1)} = -(1-M)^{\boldsymbol{k}}M_1\boldsymbol{A}\left(\boldsymbol{f}^{(ob,0)} - \boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}\right)_{\boldsymbol{n}\le s}\right) - \sum_{\boldsymbol{k}'=1}^{\boldsymbol{k}}(1-M)^{\boldsymbol{k}-\boldsymbol{k}'}\boldsymbol{A}\left(M\boldsymbol{f}^{(ob,\boldsymbol{k}')} - M_1\boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},\boldsymbol{k}')}\right)_{\boldsymbol{n}\le s}\right) = (1-M)\boldsymbol{p}^{(\boldsymbol{k})} - \boldsymbol{A}\left(M\boldsymbol{f}^{(ob,k)} - M_1\boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},\boldsymbol{k})}\right)_{\boldsymbol{n}\le s}\right). \quad \text{(Equation 63)}$$

Denoting the ground truth image of $\boldsymbol{p}^{(k)}$ as $\boldsymbol{f}^{(*,k)}$, that is $\boldsymbol{p}^{(k)} = \boldsymbol{A}\boldsymbol{f}^{(*,k)}$, we have

$$\boldsymbol{f}^{(*,k+1)} = (1-M)\boldsymbol{f}^{(*,k)} - M\boldsymbol{f}^{(ob,k)} + M_1\boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},k)}\right)_{\boldsymbol{n}\le s}. \quad \text{(Equation 64)}$$

$$\|\boldsymbol{f}^{(ob,k+1)}\| < (1-\sigma)\|\boldsymbol{f}^{(*,k+1)}\|. \quad \text{(Equation 65)}$$

Because each component of $(\bar{\boldsymbol{f}}^{(\boldsymbol{e},k)})_{\boldsymbol{n}\le s}$ is bounded by ε, we have

$$\|\boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},k)}\right)_{\boldsymbol{n}\le s}\| = \|\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},k)}\right)_{\boldsymbol{n}\le s}\| \le \varepsilon\sqrt{s}. \quad \text{(Equation 66)}$$

Equations 64–66 imply:

$$\|\boldsymbol{f}^{(*,1)}\| = \|M_1\boldsymbol{f}^{(ob,0)} + M_1\boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},0)}\right)_{\boldsymbol{n}\le s}\| \le M_1\|\boldsymbol{f}^{(ob,0)}\| + M_1\varepsilon\sqrt{s}, \quad \text{(Equation 67)}$$

$$\begin{aligned}\|\boldsymbol{f}^{(*,2)}\| &= \|(1-M)\boldsymbol{f}^{(*,1)} - M\boldsymbol{f}^{(ob,1)} + M_1\boldsymbol{H}^*\left(\bar{\boldsymbol{f}}^{(\boldsymbol{e},1)}\right)_{\boldsymbol{n}\le s}\| \le (1-M)\|\boldsymbol{f}^{(*,1)}\| \\ &+ M\|\boldsymbol{f}^{(ob,1)}\| + M_1\varepsilon\sqrt{s} \le (1-M)\|\boldsymbol{f}^{(*,1)}\| + M(1-\sigma)\|\boldsymbol{f}^{(*,1)}\| \\ &+ M_1\varepsilon\sqrt{s} \le (1-M\sigma)\left(M_1\|\boldsymbol{f}^{(ob,0)}\| + M_1\varepsilon\sqrt{s}\right) \\ &+ M_1\varepsilon\sqrt{s} \le (1-M\sigma)M_1\|\boldsymbol{f}^{(ob,0)}\| + ((1-M\sigma)+1)M_1\varepsilon\sqrt{s}.\end{aligned} \quad \text{(Equation 68)}$$

If we continue this process, we can reach

$$\|\boldsymbol{f}^{(*,k+1)}\| \le (1-M\sigma)^kM_1\|\boldsymbol{f}^{(ob,0)}\| + \sum_{\boldsymbol{k}'=0}^{\boldsymbol{k}}(1-M\sigma)^{\boldsymbol{k}'}M_1\varepsilon\sqrt{s} = (1-M\sigma)^kM_1\|\boldsymbol{f}^{(ob,0)}\| + \frac{1-(1-M\sigma)^{k+1}}{M\sigma}M_1\varepsilon\sqrt{s}, \quad \text{(Equation 69)}$$

$$\|\boldsymbol{f}^{(ob,k+1)}\| < (1-\sigma)\left((1-M\sigma)^kM_1\|\boldsymbol{f}^{(ob,0)}\| + \frac{1-(1-M\sigma)^{k+1}}{M_2\sigma}\varepsilon\sqrt{s}\right). \quad \text{(Equation 70)}$$

When $k\to\infty$, Equation 70 shows

$$\|\boldsymbol{f}^{(ob,\infty)}\| < \frac{(1-\sigma)\sqrt{s}}{M_2\sigma}\varepsilon. \quad \text{(Equation 71)}$$

Because the parameter ε for the soft-thresholding kernel should match the system tolerance level, it is a noise floor. Equation 71 implies that $\boldsymbol{f}^{(ob,k)}$ will converge to a noise-induced uncertainty range of the imaging system. For an ideal noise-free case, the matching $\varepsilon\to 0$ and $\|\boldsymbol{f}^{(ob,\infty)}\|\to 0$. The bound (Equation 69) will monotonously decrease if it satisfies $\frac{\varepsilon\sqrt{s}}{\|\boldsymbol{f}\|} < \frac{(1-\sigma)M\sigma}{(1-M\sigma)}$. In other words, if the image is not too noisy, the ACID algorithm will converge to a solution in the intersection of the space constrained by measured data, the space of sparse solutions, and the space of deep priors.

In the above analysis, we have assumed that the input to the neural network is noise free; that is, $\boldsymbol{p}^{(0)} = \boldsymbol{A}\boldsymbol{f}^*$. When there is a noise component in the projection data, this noise $\boldsymbol{e}$ can be decomposed into two parts, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, where $\boldsymbol{e}_1$ satisfying $\boldsymbol{e}_1 = \boldsymbol{A}\boldsymbol{n}^*$ with $\boldsymbol{n}^*$ being the observable image corresponding to the noise so that $\boldsymbol{f}^* + \boldsymbol{n}^*$ is still consistent to both the data-driven prior and the sparsity condition, and $\boldsymbol{e}_2 = \boldsymbol{e} - \boldsymbol{e}_1$ as a complement of $\boldsymbol{e}_1$. Because the image $\boldsymbol{n}^*$ can be absorbed by $\boldsymbol{f}^*$, we can ignore $\boldsymbol{e}_1$ and only consider $\boldsymbol{e}_2$. Because $\boldsymbol{e}_2$ is outside the intersection of the three spaces constrained by (1) data-driven prior, (2) sparsity condition, and (3) measurement data, and thus makes no contribution to the final image, we can just modify the system tolerance level ε accordingly to accommodate the effect of the noise $\boldsymbol{e}$ without affecting the above convergence analysis.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.patter.2022.100475.

ACKNOWLEDGMENTS

W.W. was partially supported by the Li Ka Shing Medical Foundation, Hong Kong. V.V. is partially supported by the Li Ka Shing Medical Foundation, Hong Kong. W.W., H.S., W.C., C.N., H.Y., and G.W. are partially supported by NIH grants U01EB017140, R01EB026646, R01CA233888, R01CA237267, and R01HL151561, USA.

AUTHOR CONTRIBUTIONS

G.W. initiated the project and supervised the team in collaboration with H.Y. and V.V. W.W., H.Y., and G.W. designed the ACID network. W.W. and D.H. conducted the experiments. H.Y., W.C., and G.W. established the mathematical model and performed the theoretical analysis. W.W., H.Y., and G.W. drafted the manuscript. W.W., D.H., and H.S. worked on user-friendly codes/data sharing. All co-authors participated in discussions, contributed technical points, and revised the manuscript iteratively.

DECLARATION OF INTERESTS

G.W. is an advisory board member of *Patterns*. An invention disclosure was filed to the Office of Intellectual Property Organization of Rensselaer Polytechnic Institute in August, 2020, and the US Non-provisional Patent Application was filed in August, 2021. The authors declare no competing interests.

Received: November 26, 2021
Revised: December 24, 2021
Accepted: March 1, 2022
Published: April 6, 2022

REFERENCES

1. Targonski, C., Bender, M.R., Shealy, B.T., Husain, B., Paseman, B., Smith, M.C., and Feltus, F.A. (2020). Cellular state transformations using deep learning for precision medicine applications. Patterns *1*, 100087. https://doi.org/10.1016/j.patter.2020.100087.
2. Antun, V., Renna, F., Poon, C., Adcock, B., and Hansen, A.C. (2020). On instabilities of deep learning in image reconstruction and the potential costs of AI. Proc. Natl. Acad. Sci. U S A *117*, 30088–30095. https://doi.org/10.1073/pnas.1907377117.

3. Wu, W.W., Hu, D.L., Cong, W.X., Shan, H.M., Wang, S.Y., Niu, C., Yan, P.K., Yu, H.Y., Vardhanabhuti, V., and Wang, G. (2022). Stabilizing deep tomographic reconstruction– Part A: hybrid framework and experimental results. Patterns *3*, 100474.

4. Smale, S. (1998). Mathematical problems for the next century. The Math. Intelligencer *20*, 7–15. Available from: https://doi.org/10.1142/9789812792815_0045.

5. Bastounis, A., Hansen, A.C., and Vlacic, V. (2021). The extended Smale's 9th problem - on computational barriers and paradoxes in estimation, regularisation, computer-assisted proofs and learning. https://www.mins.ee.ethz.ch/pubs/files/smales9th.pdf.

6. Gottschling, N.M., Antun, V., Adcock, B., and Hansen, A.C. (2020). The troublesome kernel: why deep learning for inverse problems is typically unstable. Preprint at arXiv, 2001.01258.

7. Antun, V., Colbrook, M.J., and Hansen, A.C. (2021). Can stable and accurate neural networks be computed?–On the barriers of deep learning and Smale's 18th problem. Preprint at arXiv, 2101.08286.

8. Chen, H., Zhang, Y., Chen, Y., Zhang, J., Zhang, W., Sun, H., Lv, Y., Liao, P., Zhou, J., and Wang, G. (2018). LEARN: learned experts' assessment-based reconstruction network for sparse-data CT. IEEE Trans. Med. Imaging *37*, 1333–1347.

9. Genzel, M., Macdonald, J., and März, M. (2020). Solving inverse problems with deep neural networks–robustness included?. Preprint at arXiv, 2011.04268.

10. Chun, I.Y., Huang, Z., Lim, H., and Fessler, J.A. (2020). Momentum-Net: fast and convergent iterative neural network for inverse problems. IEEE Trans. Pattern Anal. Mach. Intell. https://doi.org/10.1109/TPAMI.2020.3012955.

11. Schwab, J., Antholzer, S., and Haltmeier, M. (2019). Deep null space learning for inverse problems: convergence analysis and rates. Inverse Probl. *35*. https://doi.org/10.1088/1361-6420/aaf14a.

12. Gilton, D., Ongie, G., and Willett, R. (2021). Deep equilibrium architectures for inverse problems in imaging. Preprint at arXiv. 2102.07944. https://doi.org/10.48550/arXiv.2102.07944.

13. Pan, J., Wu, W.W., Gao, Z., and Zhang, H.Y. (2021). Multi-domain integrative Swin transformer network for sparse-view tomographic reconstruction. Preprint at arXiv. 2111.14831. https://doi.org/10.48550/arXiv.2111.14831.

14. Wu, W.W., Hu, D.L., Niu, C., Yu, H.Y., Vardhanabhuti, V., and Wang, G. (2021). DRONE: dual-domain residual-based optimization network for sparse-view CT reconstruction. IEEE Trans. Med. Imaging *40*, 3002–3014. https://doi.org/10.1109/TMI.2021.3078067.

15. Jiang, M., and Wang, G. (2003). Convergence studies on iterative algorithms for image reconstruction. IEEE Trans. Med. Imaging *22*, 569–579. https://doi.org/10.1109/TMI.2003.812253.

16. Jiang, M., and Wang, G. (2003). Convergence of the simultaneous algebraic reconstruction technique (SART). IEEE Trans. Image Process. *12*, 957–961. https://doi.org/10.1109/TIP.2003.815295.

17. Wu, W.W., Zhang, Y.B., Wang, Q., Liu, F.L., Chen, P.J., and Yu, H.Y. (2018). Low-dose spectral CT reconstruction using image gradient ℓ0–norm and tensor dictionary. Appl. Math. Model. *63*, 538–557. https://doi.org/10.1016/j.apm.2018.07.006.

18. Wu, W.W., Hu, D.L., An, K., Wang, S.Y., and Luo, F.L. (2020). A high-quality photon-counting CT technique based on weight adaptive total-variation and image-spectral tensor factorization for small animals imaging. IEEE Trans. Instrumentation Meas. *70*. https://doi.org/10.1109/TIM.2020.3026804.

19. Katsevich, A. (2002). Analysis of an exact inversion algorithm for spiral cone-beam CT. Phys. Med. Biol. *47*, 2583–2597. https://doi.org/10.1088/0031-9155/47/15/302.

20. Axel, L., Summers, R., Kressel, H., and Charles, C. (1986). Respiratory effects in two-dimensional Fourier transform MR imaging. Radiology *160*, 795–801. https://doi.org/10.1148/radiology.160.3.3737920.

21. Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. J. Optimization Theor. Appl. *109*, 475–494. https://doi.org/10.1023/A:1017501703105.

22. Poon, C. (2015). On the role of total variation in compressed sensing. SIAM J. Imaging Sci. *8*, 682–720. https://doi.org/10.1137/140978569.

23. Wang, Y., Yang, J., Yin, W., and Zhang, Y. (2008). A new alternating minimization algorithm for total variation image reconstruction. SIAM J. Imaging Sci. *1*, 248–272. https://doi.org/10.1137/080724265.

24. Foucart, S., and Rauhut, H. (2013). An invitation to compressive sensing. In A mathematical introduction to compressive sensing (Springer), pp. 1–39. https://doi.org/10.1007/978-0-8176-4948-7_1.

25. Yu, H.Y., and Wang, G. (2010). A soft-threshold filtering approach for reconstruction from a limited number of projections. Phys. Med. Biol. *55*, 3905–3916. https://doi.org/10.1088/0031-9155/55/13/022.

26. Jin, K.H., McCann, M.T., Froustey, E., and Unser, M. (2017). Deep convolutional neural network for inverse problems in imaging. IEEE Trans. Image Process. *26*, 4509–4522. https://doi.org/10.1109/TIP.2017.2713099.

27. Zhu, B., Liu, J.Z., Cauley, S.F., Rosen, B.R., and Rosen, M.S. (2018). Image reconstruction by domain-transform manifold learning. Nature *555*, 487–492. https://doi.org/10.1038/nature25988.

28. Schlemper, J., Caballero, J., Hajnal, J.V., Price, A.N., and Rueckert, D. (2018). A deep cascade of convolutional neural networks for dynamic MR image reconstruction. IEEE Trans. Med. Imaging *37*, 491–503. https://doi.org/10.1109/TMI.2017.2760978.

29. Hammernik, K., Klatzer, T., Kobler, E., Recht, M.P., Sodickson, D.K., Pock, P., and Knoll, F. (2018). Learning a variational network for reconstruction of accelerated MRI data. Magn. Reson. Med. *79*, 3055–3071. https://doi.org/10.1002/mrm.26977.

30. Koonjoo, N., Zhu, B., Bagnall, G.C., Bhutto, B., and Rosen, M. (2021). Boosting the signal-to-noise of low-field MRI with deep learning image reconstruction. Sci. Rep. *11*. https://doi.org/10.1038/s41598-021-87482-7.

31. Wang, X., Yan, J., Jin, B., and Li, W. (2021). Distributed and parallel ADMM for structured nonconvex optimization problem. IEEE Trans. Cybernetics *51*, 4540–4552. https://doi.org/10.1109/TCYB.2019.2950337.

32. Barber, R.F., and Sidky, E.Y. (2016). MOCCA: mirrored convex/concave optimization for nonconvex composite functions. J. Mach Learn Res. *17*, 1–51.

33. Liu, J., Hu, Y., Yang, J., Chen, Y., Shu, H., Luo, L., Feng, Q., Gui, Z., and Coatrieux, G. (2016). 3D feature constrained reconstruction for low-dose CT imaging. IEEE Trans. Circuits Syst. Video Technol. *28*, 1232–1247. https://doi.org/10.1109/TCSVT.2016.2643009.

34. Valkonen, T., Bredies, K., and Knoll, F. (2013). Total generalized variation in diffusion tensor imaging. SIAM J. Imaging Sci. *6*, 487–525. https://doi.org/10.1137/120867172.

35. Kurakin, A., Goodfellow, I., and Bengio, S. (2016). Adversarial examples in the physical world. Preprint at arXiv, 1607.02533.

36. Finlayson, S.G., Bowers, D.D., Ito, J., Zittrain, J.L., Beam, A.L., and Kohane, I.S. (2019). Adversarial attacks on medical machine learning. Science *363*, 1287–1289. https://doi.org/10.1126/science.aaw4399.

37. Fawzi, A., Moosavi-Dezfooli, S.M., and Frossard, P. (2017). The robustness of deep networks: a geometrical perspective. IEEE Signal Process. Mag. *34*, 50–62. https://doi.org/10.1109/MSP.2017.2740965.

38. Kang, E., Min, J., and Ye, J.C. (2017). A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. Med. Phys. *44*, e360–e375. https://doi.org/10.1002/mp.12344.

39. Wu, D., Kim, K., El Fakhri, G., and Li, Q.Z. (2017). Iterative low-dose CT reconstruction with priors trained by artificial neural network. IEEE Trans. Med. Imaging *36*, 2479–2486. https://doi.org/10.1109/TMI.2017.2753138.

40. Hyun, C.M., Kim, H.P., Lee, S.M., Lee, S., and Seo, J.K. (2018). Deep learning for undersampled MRI reconstruction. Phys. Med. Biol. *63*. https://doi.org/10.1088/1361-6560/aac71a.

41. He, J., Wang, Y., and Ma, J. (2020). Radon inversion via deep learning. IEEE Trans. Med. Imaging *39*, 2076–2087. https://doi.org/10.1109/TMI.2020.2964266.

42. Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In Proceedings of the 20th international conference on machine learning (icml-2003) (CMU), pp. 928–935.

43. Leung, H., and Haykin, S. (1991). The complex backpropagation algorithm. IEEE Trans. Signal Process. *39*, 2101–2104. https://doi.org/10.1109/78.134446.

44. Chauvin, Y., and Rumelhart, D.E. (1995). Backpropagation: Theory, Architectures, and Applications (Psychology press).

45. Ganin, Y., and Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In International conference on machine learning (PMLR), pp. 1180–1189.

46. LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature *521*, 436–444. https://doi.org/10.1038/nature14539.