Near-Exact Recovery for Tomographic Inverse Problems via Deep Learning

Martin Genzel [* 1] **Ingo Gühring** [* 2] **Jan Macdonald** [* 2] **Maximilian März** [* 2]

Abstract

This work is concerned with the following fundamental question in scientific machine learning: Can deep-learning-based methods solve noise-free inverse problems to near-perfect accuracy? Positive evidence is provided for the first time, focusing on a prototypical computed tomography (CT) setup. We demonstrate that an iterative end-to-end network scheme enables reconstructions close to numerical precision, comparable to classical compressed sensing strategies. Our results build on our winning submission to the recent AAPM DL-Sparse-View CT Challenge. Its goal was to identify the state-of-the-art in solving the sparse-view CT inverse problem with data-driven techniques. A specific difficulty of the challenge setup was that the precise forward model remained unknown to the participants. Therefore, a key feature of our approach was to initially estimate the unknown fanbeam geometry in a data-driven calibration step. Apart from an in-depth analysis of our methodology, we also demonstrate its state-of-the-art performance on the open-access real-world dataset LoDoPaB CT.

1. Introduction

In recent years, deep learning methods have been successfully applied to many problems of the natural sciences (Le-Cun et al., 2015; Schmidhuber, 2015; Goodfellow et al., 2016). A prominent example of such *scientific machine learning* is the development of efficient solutions strategies for inverse problems (Arridge et al., 2019; Ongie et al., 2020), such as those encountered in medical imaging. But despite unprecedented empirical performance in various practical scenarios, a lack of evidence for the relia-

bility of these methods remains. For instance, Sidky et al. (2021a) have recently demonstrated that *post-processing* of filtered backprojection images with the prominent UNet-architecture may not yield satisfactory recovery precision in sparse-view computed tomography (CT). This observation gave rise to the recent AAPM Grand Challenge "Deep Learning for Inverse Problems: Sparse-View Computed Tomography Image Reconstruction", with the goal *"to identify the state-of-the-art in solving the CT inverse problem with data-driven techniques"* (Sidky et al., 2021b). Here, the term 'solving' is used to describe algorithms that provide perfect recovery from incomplete, noiseless measurements.

The study of this desirable property was popularized by the field of *compressed sensing* (Candès et al., 2006; Donoho, 2006; Foucart & Rauhut, 2013). Indeed, high precision in the noiseless, undersampled regime can be used to benchmark reconstruction methods and is a driving factor for their acceptance in practice. Therefore, an important open research question is whether deep-learning-based schemes can achieve such (near-)perfect solutions, comparable to model-based algorithms like total variation (TV) minimization. The present article makes first progress in this direction, building on our winning submission to the AAPM challenge. Our main contributions are as follows:

(i) We show that end-to-end neural networks can achieve *near-perfect accuracy* on the prescribed CT reconstruction task. This underscores the reliability of deep-learning-based solvers for inverse problems, in the sense that they can match the precision of a widely-accepted benchmark (TV minimization) in the noiseless limit.

(ii) We give a *detailed analysis* of our solution strategy, which has significantly outperformed the runner-up teams. Although the challenge amounts to a comparison with 24 competing methods, we also explicitly demonstrate the superiority over several popular baselines in this work. This includes the learned primal-dual algorithm (Adler & Öktem, 2018), which is commonly considered as state-of-the-art for solving inverse problems, cf. Leuschner et al. (2021); Ramzi et al. (2020). In addition, we show the effectiveness of our learning pipeline beyond synthetically generated image data: The proposed neural network scheme produces state-of-the-art results on the LoDoPaB CT dataset (Leuschner et al., 2021), currently ranked first in the public leaderboard.

---

[*]Equal contribution (the authors are ordered alphabetically by last name). [1]Helmholtz-Zentrum Berlin für Materialien und Energie, Germany (work done while at Utrecht University, Netherlands) [2]Technical University Berlin, Germany. Correspondence to: Martin Genzel <martingenzel@gmail.com>.

(iii) We distill several insights of broader interest and conceptual value. Most notably, we found that *simple building blocks* (e.g., end-to-end training, alternation between learned and model-based components, etc.) and a careful *pre-training strategy* already allow for remarkable performance gains.

(iv) A specific difficulty of the AAPM challenge setup was an *unknown (fanbeam) forward model*. Therefore, a crucial step of our approach consists in the data-driven estimation of the underlying fanbeam geometry. We accomplish this by fitting a generic, parameterized fanbeam operator to the provided sinogram-image pairs in a deep-learning-like fashion (i.e., by gradient descent with backpropagation/automatic differentiation). This conception may find further application in the context of geometric calibration and forward operator correction.

Conceptually, our approach stems from the following (debatable) observation:

> *High reconstruction accuracy is only possible if the forward model is explicitly incorporated into the solution map, e.g., by an iterative promotion of data-consistency.*

The vital role of the forward operator in data-driven solutions to inverse problems is by no means a new insight. It is well in line with a central pillar of scientific machine learning, namely that neural networks can be often enriched (or constrained) by physical modeling. Indeed, the seminal works on deep learning techniques for inverse problems are inspired by *unrolling* classical algorithms, e.g., see Gregor & LeCun (2010); Yang et al. (2016); Hammernik et al. (2018); Aggarwal et al. (2018); Adler & Öktem (2018); Chen et al. (2018); Schlemper et al. (2019); Hammernik et al. (2021); Chun et al. (2020); Heaton et al. (2021); Gilton et al. (2021a). At the present time, most state-of-the-art methods rely on *iterative* end-to-end networks and related schemes, e.g., see Knoll et al. (2020); Muckley et al. (2020); Leuschner et al. (2021) for other recent competition benchmarks.

Our contribution to the AAPM challenge is no exception in that respect. We propose a conceptually simple, yet powerful deep learning pipeline, which turns a post-processing UNet (Ronneberger et al., 2015) into an iterative reconstruction scheme. While many of its individual components have been previously reported in the literature, the overall strategy is novel. Our design differs from more common unrolled networks in several aspects, most notably the following two: (a) we make use of a *pre-trained* UNet as the computational backbone, and (b) data-consistency is inspired by an ℓ^2-gradient step, but employs the *filtered* backprojection (FBP) instead of the regular adjoint. In line with most previous works, our unrolled network only involves very

few (five) iteration steps. However, we are the first to show that this is sufficient to match the precision of model-based solvers, which typically need hundreds or thousands of iterations before convergence (and therefore require significantly more computation time).

The Bigger Picture

To put our results into a broader context, it is worth considering an inverse problem in its prototypical, finite-dimensional form:

$$\boldsymbol{y} = \boldsymbol{F}\boldsymbol{x} + \boldsymbol{e},$$

where $\boldsymbol{x} \in \mathbb{R}^N$ denotes the unknown (image) signal, $\boldsymbol{F} \in \mathbb{R}^{m \times N}$ the forward operator, and $\boldsymbol{y}$ are noisy measurements. The goal is to reconstruct $\boldsymbol{x}$ from $\boldsymbol{y}$. The error of any given reconstruction map $\mathcal{R} : \mathbb{R}^m \to \mathbb{R}^N$ can be decomposed as

$$\big\|\boldsymbol{x} - \mathcal{R}(\boldsymbol{y})\big\|_2 \leq \underbrace{\big\|\boldsymbol{x} - \mathcal{R}(\boldsymbol{F}\boldsymbol{x})\big\|_2}_{(a)} + \underbrace{\big\|\mathcal{R}(\boldsymbol{F}\boldsymbol{x}) - \mathcal{R}(\boldsymbol{y})\big\|_2}_{(b)}.$$

The first term (a) is associated with the *accuracy* (or *precision*) of $\mathcal{R}$ and measures how well $\boldsymbol{x}$ can be estimated in the idealistic situation of noiseless measurements. The second term (b) captures the *robustness* of $\mathcal{R}$ against perturbations $\boldsymbol{e}$ of the measurements. Adequate control over both expressions forms the backbone of inverse problem theory and scientific computing in general.

The present paper is primarily concerned with the accuracy term (a), more specifically, the root-mean-square-error (RMSE). We demonstrate that it can become sufficiently close to zero (on the image data distribution) when $\mathcal{R}$ corresponds to a fully data-driven solver. The competitive setting of the AAPM challenge has provided us with the right benchmark to conduct such a case study.

Regarding the robustness term (b), we refer to the recent work by Genzel et al. (2022) for an in-depth case study. It was shown that even standard end-to-end networks can be surprisingly robust against adversarial perturbations (i.e., worst-case noise), comparable to a provably stable benchmark methods. Together with Genzel et al. (2022), the present work provides further evidence for the reliability of deep-learning-based solutions to inverse problems.

Organization of This Article

The rest of this article is organized as follows. Section 2 gives a brief overview of the AAPM challenge setup and the training/test-data. Section 3 is then devoted to a conceptual description of our learning pipeline, while more details on the implementation can be found in Appendix A. Our results and several accompanying experiments are reported in Section 4. We conclude with a short discussion in Section 5. For an evaluation of our method on the LoDoPaB CT dataset, see Appendix C.
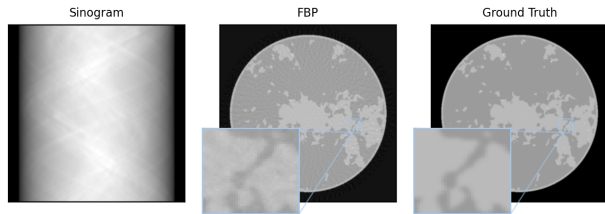
Figure 1. **AAPM challenge data.** Example of a 128-view sinogram, FBP reconstruction, and ground-truth phantom image taken from the AAPM challenge training dataset.

2. AAPM Challenge Setup

The AAPM challenge data is similar to the setting of Sidky et al. (2021a), i.e., it is based on synthetic 2D grayscale images of size 512×512 simulating real-world mid-plane breast CT device scans. Four different tissues were modeled: adipose, skin, fibroglandular tissue, and microcalcifications. To obtain smooth transitions at tissue boundaries, Gaussian smoothing was applied. A fanbeam geometry with 128 projections over 360 degrees was used to create sinograms and FBPs, see Fig. 1 for an example. Notably, the exact fanbeam geometry was not revealed to the participants. No noise was added to the data, neither to the phantom images nor the measurements. The provided training set consisted of 4000 tuples of phantom images, their corresponding 128-view sinograms, and FBP reconstructions. A test set of 100 pairs of sinograms and FBPs (without publicly available ground-truth phantoms) was used for the final challenge evaluation.

Initially, about 50 international teams have participated, out of which 25 have submitted their method to the final evaluation. More details about the challenge setup and results can be found in the official challenge report (Sidky & Pan, 2021).

3. Methodology

This section gives an overview of our (three-step) methodology for the AAPM challenge and motivates our design choices.[1]

Step 1: Data-Driven Geometry Identification

In the first step of our reconstruction pipeline, we estimate the unknown forward operator from the provided training data. The continuous version of *tomographic fanbeam measurements* is based on computing line integrals:

$$p(s, \varphi) = \int_{L(s,\varphi)} x_0(x, y) \, \mathrm{d}(x, y),$$
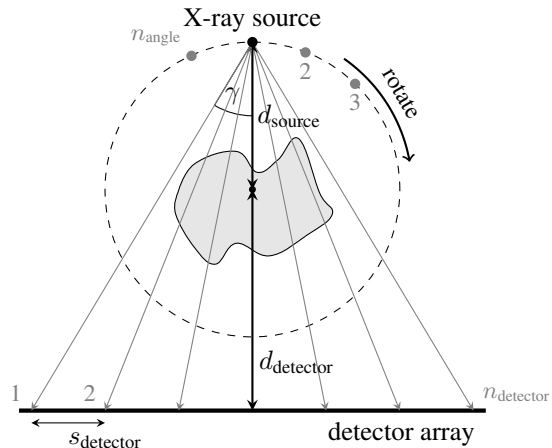
Figure 2. **Fanbeam geometry.** Illustration of the parameters determining the geometry of the fanbeam CT model.

where x_0 is the unknown image and $L(s, \varphi)$ denotes a line in fanbeam coordinates, i.e., φ is the *fan rotation angle* and s encodes the *sensor position*; see Fessler (2017) for more details. In an idealized situation, the fanbeam model is specified by the following geometric parameters[2] (see Fig. 2 for an illustration):

- d_{source} – distance of the X-ray source to the origin,
- d_{detector} – distance of the detector array to the origin,
- n_{detector} – number of detector elements,
- s_{detector} – spacing of detector elements along the array,
- n_{angle} – number of fan rotation angles,
- $\varphi \in [0, 2\pi]^{n_{\text{angle}}}$ – discrete list of rotation angles.

Here, it is assumed that integrals are only measured along a finite number of lines, determined by $m := n_{\text{detector}} \cdot n_{\text{angle}}$. In the *sparse-view* challenge setup, the resulting forward operator is *severely ill-posed*, since only the measurements of a few fan rotation angles n_{angle} are acquired. Furthermore, the geometric setup is not disclosed to the challenge participants—it is only known that fanbeam measurements are used.

We have addressed this lack of information by a data-driven estimation strategy that fits the above set of parameters to the given training data. To this end, we first observe that the previous parametrization is redundant, and without of loss of generality, we may assume that $s_{\text{detector}} = 1$ (by rescaling d_{detector} appropriately). Further, if the field-of-view angle γ

is known, then the relation

$$d_{\text{detector}} = \frac{n_{\text{detector}} \cdot s_{\text{detector}}}{2 \tan \gamma} - d_{\text{source}} \qquad (1)$$

can be used to eliminate another parameter. Thus, the fanbeam geometry is effectively determined by the reduced parameter set $(d_{\text{source}}, n_{\text{detector}}, n_{\text{angle}}, \boldsymbol{\varphi})$. The training data provides pairs of discrete images $\boldsymbol{x} \in \mathbb{R}^{512 \cdot 512 =: N}$ and its simulated fanbeam measurements $\boldsymbol{y} \in \mathbb{R}^{128 \cdot 1024 = m}$, from which the dimensions $n_{\text{angle}} = 128$ and $n_{\text{detector}} = 1024$ can be derived. We determine the field of view as $\gamma = \arcsin(256/d_{\text{source}})$, so that the maximum inscribed circle in the discrete image is exactly contained within each fan of lines, which is a common choice for fanbeam CT. Hence, (1) leads to

$$d_{\text{detector}} = 2 \cdot s_{\text{detector}} \cdot \sqrt{d_{\text{source}}^2 - 256^2} - d_{\text{source}} \ .$$

The main difficulty of Step 1 lies in the estimation of the remaining parameters $(d_{\text{source}}, \boldsymbol{\varphi})$. To that end, we have implemented a discrete fanbeam transform from scratch in `PyTorch` (together with its corresponding FBP). A distinctive aspect of our implementation is the use of a vectorized numerical integration that enables the efficient computation of derivatives with respect to the geometric parameters by means of *automatic differentiation*. This feature can be exploited for a data-driven parameter identification, for instance, by a gradient descent. More precisely, we use a ray-driven numerical integration for the forward model and a pixel-driven and sinogram-reweighting-based FBP (with a Hamming filter), see Fessler (2017, Sec. 3.9.2). In addition to the parameters $(d_{\text{source}}, \boldsymbol{\varphi})$, we also introduce learnable scaling factors s_{fwd} and s_{fbp} for the forward and inverse transform, respectively. They account for ambiguities in chosing the discretization units of distance compared to the actual physical units of distance.

As previously indicated, we estimate the free parameters $\boldsymbol{\theta}_{\text{fan}} = (s_{\text{fwd}}, d_{\text{source}}, \boldsymbol{\varphi}) \in \mathbb{R}^{130}$ of the implemented forward operator $\boldsymbol{F}[\boldsymbol{\theta}_{\text{fan}}] \in \mathbb{R}^{m \times N}$ in a deep-learning-like fashion: The ability to compute derivatives $\frac{\mathrm{d}\boldsymbol{F}}{\mathrm{d}\boldsymbol{\theta}_{\text{fan}}}$ allows us to make use of the $M = 4000$ sinogram-image pairs $\{(\boldsymbol{y}^i, \boldsymbol{x}^i)\}_{i=1}^M$ by solving

$$\min_{\boldsymbol{\theta}_{\text{fan}}} \ \frac{1}{M} \sum_{i=1}^M \left\| \boldsymbol{F}[\boldsymbol{\theta}_{\text{fan}}](\boldsymbol{x}^i) - \boldsymbol{y}^i \right\|_2^2 \qquad (2)$$

with a variant of gradient descent (see Remark 3.1 below for details). Finally, we determine s_{fbp} by solving

$$\min_{s_{\text{fbp}}} \ \frac{1}{M} \sum_{i=1}^M \left\| \boldsymbol{x}^i - \text{FBP}[\boldsymbol{\theta}_{\text{fan}}, s_{\text{fbp}}](\boldsymbol{y}^i) \right\|_2^2 \ , \qquad (3)$$

while keeping the already identified parameters fixed. We will use the short-hand notation $\boldsymbol{F}$ and FBP for the estimated operators $\boldsymbol{F}[\boldsymbol{\theta}_{\text{fan}}]$ and $\text{FBP}[\boldsymbol{\theta}_{\text{fan}}, s_{\text{fbp}}] \colon \mathbb{R}^m \to \mathbb{R}^N$, respectively.

Remark 3.1. (a) Clearly, the formulation (2) is non-convex and therefore it is not clear whether gradient descent enables an accurate estimation of the underlying fanbeam geometry. Indeed, standard gradient descent was found to be very sensitive to the initialization of $\boldsymbol{\theta}_{\text{fan}}$ and got stuck in bad local minima. To overcome this issue, we solve (2) by a *coordinate descent* instead, which alternatingly optimizes over s_{fwd}, d_{source}, and $\boldsymbol{\varphi}$ with individual learning rates. This strategy was found to effectively account for large deviations of gradient magnitudes of the different parameters. Indeed, we observed a fast convergence and a reliable identification of $\boldsymbol{\theta}_{\text{fan}}$, independently of the initialization.

(b) In principle, the strategy of (2) requires only few training samples to be successful. However, when verifying the robustness of the outlined strategy against measurement noise, we observed that it is beneficial to employ more training data.

(c) Subsequent to the estimation of an accurate fanbeam geometry, we still recognized a small systematic error in our forward model. We suspect that it is caused by subtle differences in the numerical integration in comparison to the true forward model of the AAPM challenge. In compensation, we compute the (pixelwise) mean error over the training set, as an additive correction of the model bias.

Step 2: Pre-Training a UNet as Computational Backbone

The centerpiece of our reconstruction scheme is formed by a standard *UNet-architecture* $\text{UNet}[\boldsymbol{\theta}] \colon \mathbb{R}^N \to \mathbb{R}^N$ (Ronneberger et al., 2015) which is employed as a residual network to post-process sparse-view FBP images. The learnable parameters $\boldsymbol{\theta}$ are trained from the collection of $M = 4000$ sinogram-image pairs $\{(\boldsymbol{y}^i, \boldsymbol{x}^i)\}_{i=1}^M$ provided by the AAPM challenge. This is achieved by standard empirical risk minimization, i.e., by (approximately) solving

$$\min_{\boldsymbol{\theta}} \ \frac{1}{M} \sum_{i=1}^M \left\| \boldsymbol{x}^i - [\text{UNet}[\boldsymbol{\theta}] \circ \text{FBP}](\boldsymbol{y}^i) \right\|_2^2 + \mu \cdot \|\boldsymbol{\theta}\|_2^2 \ , \quad (4)$$

where we choose $\mu = 10^{-3}$, and $\text{FBP} \colon \mathbb{R}^m \to \mathbb{R}^N$ is obtained from Step 1. This minimization problem is tackled by 400 epochs of mini-batch stochastic gradient descent and the Adam optimizer (Kingma & Ba, 2014) with initial learning rate 0.0002 and batch size 4.

Remark 3.2. The post-processing strategy of Step 2 was pioneered by Kang et al. (2017); Chen et al. (2017b) and popularized by Jin et al. (2017); Chen et al. (2017a), among many others. Due to the multi-scale encoder-decoder structure with skip-connections, the UNet-architecture is very efficient in handling image-to-image problems. Therefore, solving (4) typically works out-of-the-box without requiring sophisticated initialization or optimization strategies (even in seemingly hopeless situations (Hauptmann & Adler,
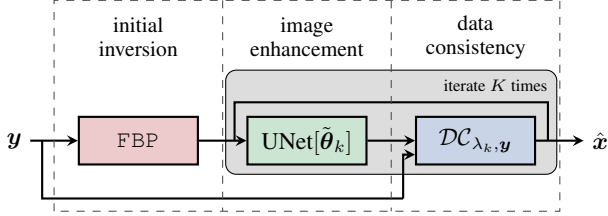
Figure 3. **Constructing an iterative scheme.** Schematic reconstruction pipeline of ItNet$_K[\boldsymbol{\theta}]$ defined in (5).

2020)). Making use of a more powerful or a more memory-efficient network would be beneficial, e.g., see results for the Tiramisu network in Section 4. However, we preferred to keep our workflow as simple as possible and therefore decided to stick to the standard UNet as the main computational building block.

Step 3: Constructing an Iterative Scheme

Our main reconstruction method is called ItNet (short for *iterative network*). It incorporates the estimated forward model $\boldsymbol{F}$ from Step 1 (and the associated inversion FBP) via the following iterative procedure:

$$\text{ItNet}_K[\boldsymbol{\theta}]\colon \mathbb{R}^m \to \mathbb{R}^N,$$

$$\boldsymbol{y} \mapsto \left[\bigcirc_{k=1}^K \left(\mathcal{DC}_{\lambda_k,\boldsymbol{y}} \circ \text{UNet}[\tilde{\boldsymbol{\theta}}_k]\right) \circ \text{FBP}\right](\boldsymbol{y}),\tag{5}$$

for the learnable parameters $\boldsymbol{\theta} = \{\tilde{\boldsymbol{\theta}}_k, \lambda_k\}_{k=1}^K$, $K \in \mathbb{N}$ and the k-th *data-consistency* layer

$$\mathcal{DC}_{\lambda_k,\boldsymbol{y}}\colon \mathbb{R}^N \to \mathbb{R}^N, \ \boldsymbol{x} \mapsto \boldsymbol{x} - \lambda_k \cdot \text{FBP}(\boldsymbol{F}\boldsymbol{x} - \boldsymbol{y}). \tag{6}$$

The ItNet-architecture[3] is illustrated in Fig. 3. We train it by empirical risk minimization analogously to (4) with $\mu = 10^{-4}$. The UNet-parameters $\tilde{\boldsymbol{\theta}}_k$ are initialized by the weights obtained in Step 2. More details on our precise training approach during the challenge submission phase can be found in Appendix A.

We close this section by pointing out several important design choices in our ItNet-architecture:

(i) The centerpiece of ItNet is the *UNet-architecture*. This stands in contrast to earlier generations of unrolled iterative schemes, which rely on basic convolutional blocks instead, e.g., see Adler & Öktem (2018); Yang et al. (2016). We have found that it is advantageous to exploit the efficacy of UNet-like image-to-image networks as enhancement blocks. This is in line with recent state-of-the-art models, which also make use of various advanced sub-networks, e.g., see Knoll et al. (2020); Muckley et al. (2020); Hammernik et al. (2021); Ramzi et al. (2020); Sriram et al. (2020).

---

[3] We drop the subscript K in ItNet$_K$ whenever it is irrelevant.

(ii) The initialization of the UNet-parameters $\tilde{\boldsymbol{\theta}}_k$ with a *pretrained model* from Step 2 has led to significant performance gains, regarding both training speed and reconstruction accuracy. We refer to Section 4 and especially Fig. 8 for a more details.

(iii) Our *data-consistency* layer is inspired by a gradient step on the loss $\boldsymbol{x} \mapsto \frac{\lambda_k}{2}\|\boldsymbol{F}\boldsymbol{x} - \boldsymbol{y}\|_2^2$, which would result in the update $\boldsymbol{x} \mapsto \boldsymbol{x} - \lambda_k \cdot \boldsymbol{F}^{\mathsf{T}}(\boldsymbol{F}\boldsymbol{x} - \boldsymbol{y})$. We depart from this scheme by replacing the unfiltered backprojection $\boldsymbol{F}^{\mathsf{T}}$ by its filtered counterpart FBP; cf. Ding et al. (2020); Tirer & Giryes (2021). This modification leads to significantly improved results for two reasons: (a) it counteracts the fact that the unfiltered backprojection is smoothing, and (b) it produces images with pixel values at the right scale. Therefore, we interpret the ItNet as an industry-like iterative CT-algorithm (e.g., see Willemink & Noël (2019)), rather than a neurally-augmented convex optimization scheme.

4. Results and Analysis

This section presents the main findings of our case study. We begin with several challenge-related experiments, followed by a more in-depth analysis of our method.

Winning the AAPM Challenge and Beyond

In terms of quantitative similarity measures, we restrict ourselves to reporting the average RMSE, which was the main evaluation metric for the AAPM challenge (Sidky et al., 2021b; Sidky & Pan, 2021). With an ensembling of ten ItNet$_5$ (more precisely a variant thereof referred to as ItNet-post, see Appendix A), we were able to achieve near-exact recovery on the test set, thereby winning the challenge with a margin of about an order of magnitude ahead of the runner-up team. The RMSE scores of all participating teams were spread across more than two orders of magnitude in a range between 6.37e-6 (ours) and 7.90e-4. Remarkably, four out of the five top-performing teams have estimated the forward fanbeam operator and made use of the sinogram data. Two of them computed an approximate TV minimization solution that was further processed by a trained neural network. The resulting solution maps involve much higher computational costs than our ItNet-post, due to a significantly larger number of forward-model evaluations. Note that reaching the first place amounts to a direct comparison with 24 competing methods, see the official AAPM challenge report (Sidky & Pan, 2021) for more details.

Nevertheless, for further analysis, we have benchmarked variants of the ItNet with different in-house baselines and other state-of-the-art methods. More specifically, we consider a post-processing of the FBP by the more powerful *Tiramisu-architecture* (Jégou et al., 2017; Bubba et al., 2019; Genzel et al., 2022) (in comparison to the UNet) as well as

Table 1. **Average RMSE scores for further evaluation.** "Challenge FBP" corresponds to the FBP reconstructions included in the challenge dataset. The method "UNet ∘ FBP" corresponds to a post-processing UNet as obtained from Step 2 of Section 3. For more details on our winning-method "ItNet-post ens." (and its pre-steps "ItNet$_4$" and "ItNet-post"), see Appendix A.

	Baselines		Our Network Variants				Comparison Networks	
	Challenge FBP	FBP	UNet ∘ FBP	ItNet$_4$	ItNet-post	ItNet-post ens.	Tiramisu	LPD
RMSE	5.72e-3	3.40e-3	3.50e-4	1.64e-5	1.05e-5	**6.42e-6**	2.24e-4	1.24e-4
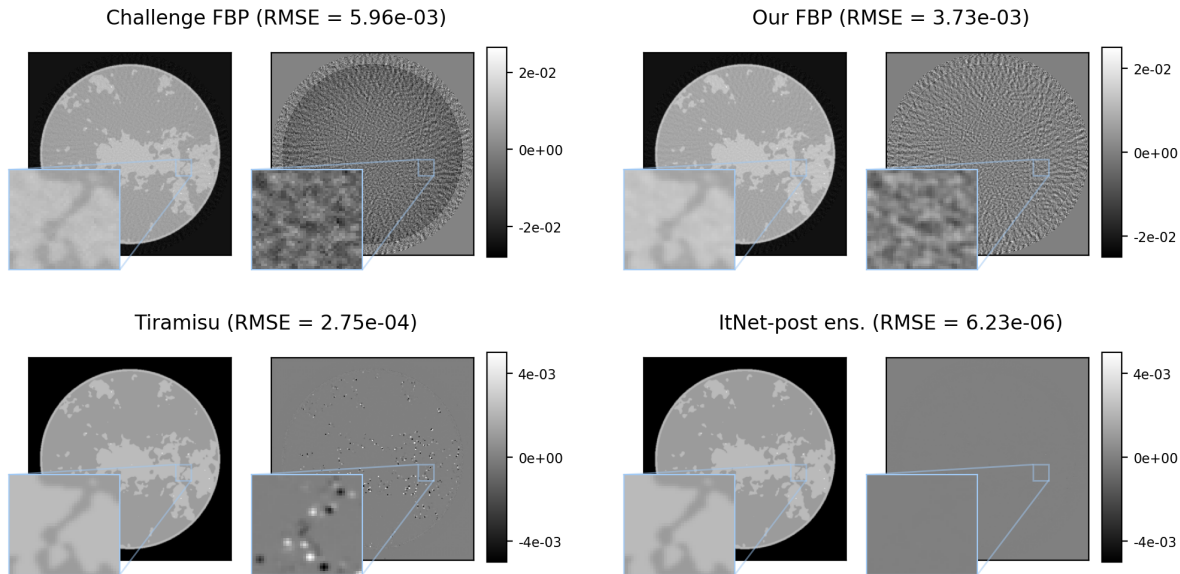
Figure 4. **Reconstruction results.** We display individual reconstructions for an image from the validation set. The first row compares the FBP provided by the AAPM challenge with our FBP (= FBP, see Step 1 of Section 3). The second row compares a post-processing Tiramisu with the (ensemble) ItNet-post. The ground-truth image is omitted because it is visually indistinguishable from the reconstruction of ItNet-post.
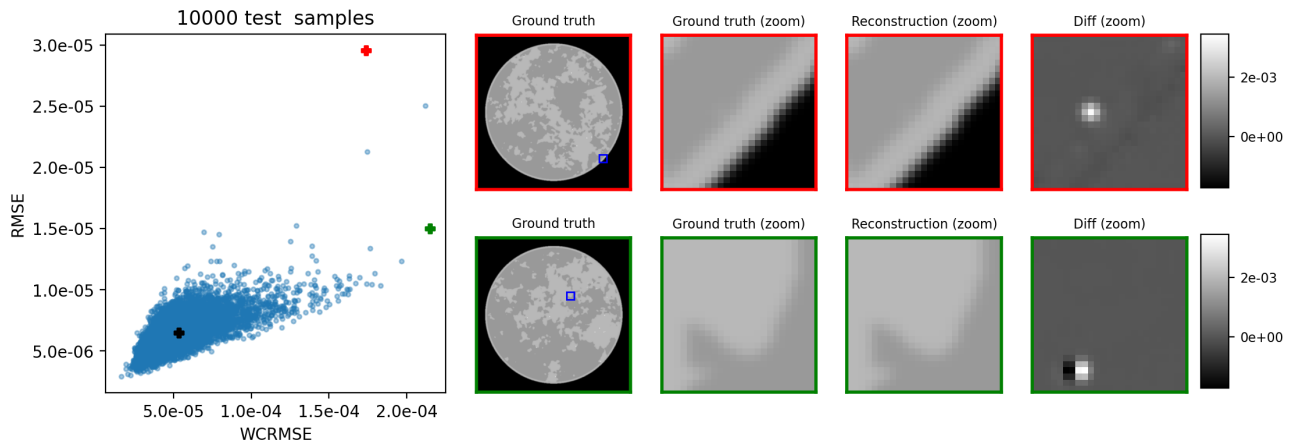

Figure 5. **Consistently accurate?** The plot on the left-hand side visualizes the reconstruction errors of the (ensemble) ItNet-post with respect to the RMSE and the WCRMSE (*worst-case* RMSE) over a set of 10000 test images. Note that the WCRMSE was used as secondary challenge metric, computing the highest RMSE value over all 25×25 sub-patches of each image. The error distribution indicates a low variance in the reconstruction performance of our method. On the right-hand side, we show the worst-case 25×25 sub-patches of the images corresponding to the red point (worst RMSE) and green point (worst WCRMSE). The black point represents the average RMSE and WCRMSE.

the iterative *learned primal-dual* (LPD) algorithm (Adler & Öktem, 2018) (modified by replacing the unfiltered backprojection with the `FBP`). LPD has been recently reported as state-of-the-art in the literature, e.g., see Ramzi et al. (2020); Leuschner et al. (2021). Table 1 shows the average RMSE scores for all methods[4] and Fig. 4 visualizes reconstructions of an image from the validation set.

After the competition period, the challenge organizer has provided us with 10000 additional test samples to increase the statistical significance of our evaluation. The resulting error distribution is visualized in Fig. 5. Although there are very few outliers, even these reconstructions are visually indistinguishable from the corresponding ground-truth phantoms. This underscores that the ItNet-post solves the CT inverse problem on the given data distribution satisfactorily.

Data-Consistency

A crucial feature of a proper solver for an inverse problem is its consistency with the forward model. In our case, this means that the difference $\boldsymbol{y} - \boldsymbol{F} \cdot \mathrm{ItNet}(\boldsymbol{y})$ should be as small as possible. We analyze this aspect in Fig. 6.[5] We observe that the data-consistency error is dominated by the error caused by the estimated forward model $\boldsymbol{F}$ (according to Step 1 of Section 3). This indicates that the performance of the ItNet could be further improved if the exact forward operator would be available. To test this hypothesis, we have trained an ItNet on sinogram data that was simulated from the ground-truth phantoms using our own estimated forward operator. As expected, the resulting ItNetSim is more accurate (about factor 2), and according to Fig. 6 bottom right, implies a much smaller data-consistency error. It is also noteworthy that the loss of data-consistency for the Tiramisu is about a factor 20 larger compared to the ItNet,[6] which highlights a typical downside of simple post-processing approaches.
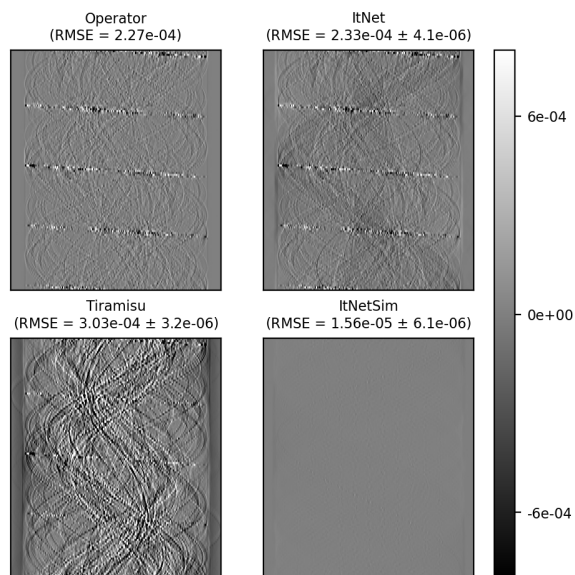
Forward Operator Needed? ... Yes! But How Often?

The previous considerations have particularly demonstrated that incorporating the forward model is key to highly accurate and data-consistent reconstructions. However, invoking the forward operator often forms the computational bottleneck of a given solution method. It is therefore important to analyze the effective number of forward/adjoint operator calls required for satisfactory precision. We address this by

Figure 6. **Data consistency.** We analyze the accuracy of our estimated forward operator by displaying the difference $\boldsymbol{y} - \boldsymbol{F}\boldsymbol{x}$ for a sinogram-image pair $(\boldsymbol{y}, \boldsymbol{x})$ from the validation set (top left); the corresponding error is the RMSE averaged over all differences. The difference $\boldsymbol{y} - \boldsymbol{F} \cdot \mathrm{ItNet}(\boldsymbol{y})$ is visually nearly indistinguishable (top right), showing that ItNet inherits the inaccuracies from the forward model. Indeed, ItNetSim exhibits a much smaller data-consistency error due to a perfectly matching forward model (bottom right). In contrast, post-processing via Tiramisu (cf. Table 1) reveals a clear lack of data-consistency (bottom left). All images are shown within the same dynamical range.

training ItNet_K for different numbers of iterations.[7] In a nutshell, Fig. 7 confirms that only a few forward operator calls are sufficient for near-exact recovery by the ItNet, which is a notable difference to classical model-based methods like TV-minimization. A closer look reveals that (a) not sharing the UNet-weights consistently outperforms weight sharing[8] by a small margin independent of the number of iterations, and (b) there is a sweetspot at about $K = 5$ after which the performance gain due to increasing K is negligible and only the training time increases. For a further discussion of weight sharing, see Appendix B.

Pre-Training Matters

When constructing the ItNet according to Step 3 of Section 3, we have observed that it is crucial to initialize the UNet-parameters $\tilde{\boldsymbol{\theta}}_k$ by the weights from the post-processing network in Step 2. This does not only increase

---

[4]Note that we report the RMSE on a subset of 125 images from the training set used for validation. Hence, values differ slightly from the actual results on the official test set. In the final challenge evaluation, ItNet-post has achieved an RMSE of 6.37e-6.

[5]Here, we have considered an ensemble of five ItNet_4.

[6]This refers to the ratio $\frac{\mathrm{RMSE}(\boldsymbol{y}, \boldsymbol{F} \cdot \mathrm{Tiramisu}(\boldsymbol{y})) - \mathrm{RMSE}(\boldsymbol{y}, \boldsymbol{F}\boldsymbol{x})}{\mathrm{RMSE}(\boldsymbol{y}, \boldsymbol{F} \cdot \mathrm{ItNet}(\boldsymbol{y})) - \mathrm{RMSE}(\boldsymbol{y}, \boldsymbol{F}\boldsymbol{x})}$.

[7]Due the significant computational effort required to conduct such an experiment, this was done on subsampled 256×256 phantom images and simulated 64-view sinograms.

[8]This means that the UNet-parameters are shared between all iterations, i.e., enforcing $\tilde{\boldsymbol{\theta}}_1 = \cdots = \tilde{\boldsymbol{\theta}}_K$ at the training stage.
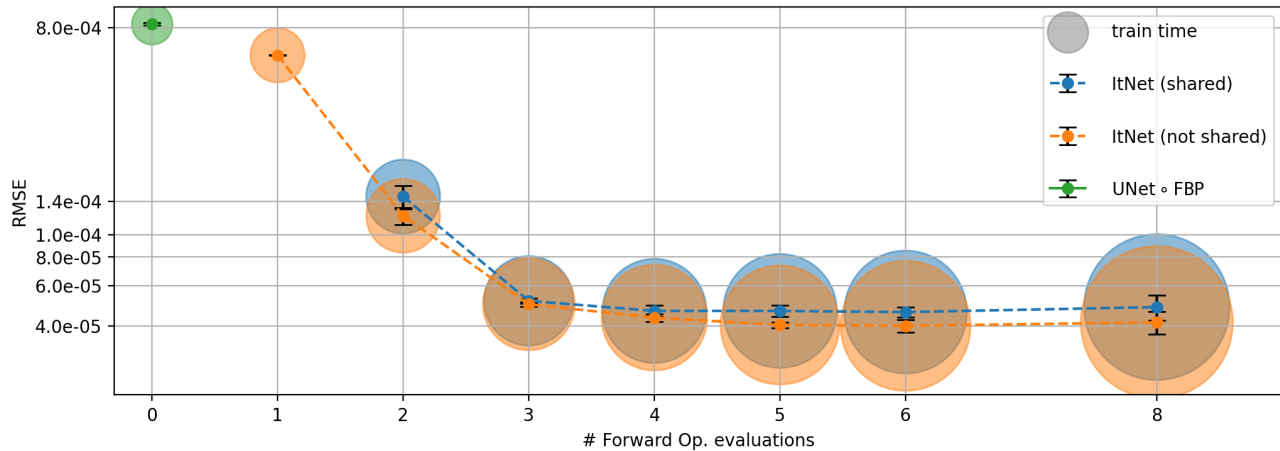
Figure 7. **The deeper the better?** Accuracy of ItNet$_K$ for different K with (blue) and without (orange) UNet weight sharing. The radii of the circles are proportional to the training time. The mean RMSE ($\pm$ std. dev.) on a hold-out evaluation data set is reported over 5 different training/validation splits. The original AAPM challenge data was subsampled to the resolution 256×256 for this experiment.
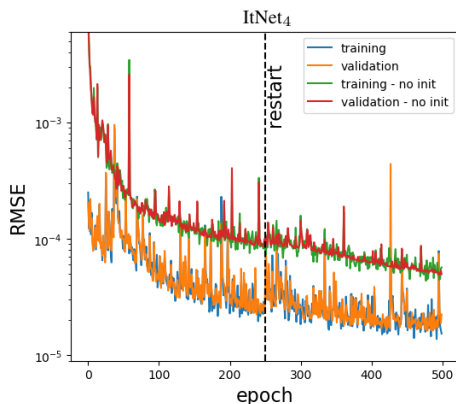

Figure 8. **The power of pre-training.** Loss curves when training the ItNet with and without a pre-initialization from Step 2 of Section 3. Note that the above loss curve only corresponds to a part of our full training pipeline, see Fig. 9 in Appendix A for the complete picture.

the speed of convergence of training ItNet, but it also significantly improves the final accuracy, see Fig. 8 for corresponding loss curves. Thus, our results show that the pre-initialization of the UNet-blocks allows finding better local minima. While the benefits of pre-trained modules are well-known for many standard machine learning tasks, to the best of our knowledge, this has not been reported in the context of inverse problems yet.

5. Conclusion

We have demonstrated that deep-learning-based solvers can produce near-perfect reconstructions for a noise-free CT inverse problem. While our approach provides evidence of feasibility, several aspects are beyond the scope of this article, some of which are pointed out in the following.

How accurate can/should we become? The reconstruction error of ItNet-post reported in Table 1 is not exactly zero, yet comparable to the precision of TV minimization, cf. Sidky et al. (2021a). There is no evidence why even more accurate results should not be achievable, for example, by increasing the internal machine precision of `PyTorch` (which is $\approx$1.19e-7 for float32); but this tweak would certainly also affect model-based algorithms. However, non-perfect recovery is not a severe issue from an applied perspective, since it is typically not required for *practical solutions* to inverse problems. We believe that our submission to the AAPM challenge has obtained satisfactory results in that respect (i.e., reconstructions visually indistinguishable from the ground-truth phantoms; see Fig. 5). Having said this, the term "near-perfect accuracy" should be used with some care when it comes to real-world scenarios. For instance, realistic CT systems involve analog-to-digital conversion processes and measurement noise, which inevitably leads to reconstruction errors. Therefore, the operating regime of the present paper is primarily a testing ground for exploring the potential capabilities of learning-based methods.

Fully data-driven or hybrid method? Although the ItNet-architecture is inspired by unrolling, it is not clear to us how well its internal mechanisms match those of classical iterative algorithms. Indeed, a distinctive feature of our approach is that only very few (five) iterations can achieve near-perfect recovery. This stands in stark contrast to model-based counterparts, which typically require hundreds or thousands of iterations to converge (resulting in significantly increased computation times). Therefore, we prefer viewing the ItNet as fully data-driven pipeline that is *model-aided* by data-consistency terms, rather than a hybrid method; see also Shlezinger et al. (2020). More generally, we suspect that viewing unrolled networks as neurally-enhanced iter-

ative schemes only partially explains the success of deep learning in inverse problems.

Model distortions? Since the purpose of data-driven methods is to adapt to a specific data distribution, the generalization to out-of-distribution features and forward-model distortions cannot be taken for granted. This aspect forms a field of active research, e.g., see Antun et al. (2020); Darestani et al. (2021); Gilton et al. (2021b) for initial results, but clearly goes beyond the scope of this paper.

Generalization to other (inverse) problems? The AAPM challenge has provided an ideal experimental area to test our research hypothesis. Although this has enabled an insightful reliability check, a foundational understanding of learned reconstruction methods is still in its infancy. In particular, it remains speculative to what extent our findings would generalize beyond the sparse-view, mono-energy CT setup. Therefore, similar case studies for different types of inverse problems and real-world data are important steps for future research. Our evaluation on the more realistic LoDoPaB CT dataset in Appendix C can be seen as a first effort in that direction.

Acknowledgements

References

Adler, J. and Öktem, O. Learned primal-dual reconstruction. *IEEE Trans. Med. Imag.*, 37(6):1322–1332, 2018.

Aggarwal, H. K., Mani, M. P., and Jacob, M. MoDL: model-based deep learning architecture for inverse problems. *IEEE Trans. Med. Imag.*, 38(2):394–405, 2018.

Antun, V., Renna, F., Poon, C., Adcock, B., and Hansen, A. C. On instabilities of deep learning in image reconstruction and the potential costs of AI. *Proc. Natl. Acad. Sci.*, 117(48):30088–30095, 2020.

Arridge, S., Maass, P., Öktem, O., and Schönlieb, C.-B. Solving inverse problems using data-driven models. *Acta Numer.*, 28:1–174, 2019.

Bubba, T. A., Kutyniok, G., Lassas, M., März, M., Samek, W., Siltanen, S., and Srinivasan, V. Learning the invisible: A hybrid deep learning-shearlet framework for limited angle computed tomography. *Inverse Probl.*, 35(6):064002, 2019.

Candès, E. J., Romberg, J. K., and Tao, T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory*, 52(2):489–509, 2006.

Chen, H., Zhang, Y., Kalra, M. K., Lin, F., Chen, Y., Liao, P., Zhou, J., and Wang, G. Low-dose CT with a residual encoder-decoder convolutional neural network. *IEEE Trans. Med. Imag.*, 36(12):2524–2535, 2017a.

Chen, H., Zhang, Y., Zhang, W., Liao, P., Li, K., Zhou, J., and Wang, G. Low-dose CT via convolutional neural network. *Biomed. Opt. Express*, 8(2):679–694, 2017b.

Chen, H., Zhang, Y., Chen, Y., Zhang, J., Zhang, W., Sun, H., Lv, Y., Liao, P., Zhou, J., and Wang, G. LEARN: learned experts' assessment-based reconstruction network for sparse-data ct. *IEEE Trans. Med. Imag.*, 37(6):1333–1347, 2018.

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

Darestani, M. Z., Chaudhari, A., and Heckel, R. Measuring robustness in deep learning based compressive sensing. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 2433–2444, 2021.

Ding, Q., Chen, G., Zhang, X., Huang, Q., Ji, H., and Gao, H. Low-dose CT with deep learning regularization via proximal forward–backward splitting. *Phys. Med. Biol.*, 65(12):125009, 2020.

Donoho, D. L. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

Fessler, J. A. Analytical tomographic image reconstruction methods (chapter 3 of book draft). URL: https://web.eecs.umich.edu/~fessler/book/c-tomo.pdf, 2017.

Foucart, S. and Rauhut, H. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser Basel, 2013.

Genzel, M., Macdonald, J., and März, M. Solving Inverse Problems With Deep Neural Networks – Robustness Included? *IEEE Trans. Pattern Anal. Mach. Intell.*, available online: https://doi.org/10.1109/tpami.2022.3148324, 2022.

Gilton, D., Ongie, G., and Willett, R. Deep equilibrium architectures for inverse problems in imaging. *IEEE Trans. Comput. Imag.*, 7:1123–1133, 2021a.

Gilton, D., Ongie, G., and Willett, R. Model adaptation for inverse problems in imaging. *IEEE Trans. Comput. Imag.*, 7:661–674, 2021b.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Gregor, K. and LeCun, Y. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 399–406, 2010.

Hammernik, K., Klatzer, T., Kobler, E., Recht, M. P., Sodickson, D. K., Pock, T., and Knoll, F. Learning a variational network for reconstruction of accelerated MRI data. *Magn. Reson. Med.*, 79(6):3055–3071, 2018.

Hammernik, K., Schlemper, J., Qin, C., Duan, J., Summers, R. M., and Rueckert, D. Systematic evaluation of iterative deep neural networks for fast parallel mri reconstruction with sensitivity-weighted coil combination. *Magn. Reson. Med.*, 86(4):1859–1872, 2021.

Hauptmann, A. and Adler, J. On the unreasonable effectiveness of CNNs. Preprint arXiv:2007.14745, 2020.

Heaton, H., Fung, S. W., Gibali, A., and Yin, W. Feasibility-based fixed point networks. arXiv:2104.14090, 2021.

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., and Bengio, Y. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11–19, 2017.

Jin, K. H., McCann, M. T., Froustey, E., and Unser, M. Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Process.*, 26(9):4509–4522, 2017.

Kang, E., Min, J., and Ye, J. C. A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Med. Phys.*, 44(10):e360–e375, 2017.

Kingma, D. P. and Ba, J. Adam: a method for stochastic optimization. Preprint arXiv:1412.6980, 2014.

Knoll, F., Murrell, T., Sriram, A., Yakubova, N., Zbontar, J., Rabbat, M., Defazio, A., Muckley, M. J., Sodickson, D. K., Zitnick, C. L., and Recht, M. P. Advancing machine learning for MR image reconstruction with an open competition: Overview of the 2019 fastMRI challenge. *Magn. Reson. Med.*, 84(6):3054–3070, 2020.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521(7553):436–444, 2015.

Leuschner, J., Schmidt, M., Ganguly, P. S., Andriiashen, V., Coban, S. B., Denker, A., Bauer, D., Hadjifaradji, A., Batenburg, K. J., Maass, P., and van Eijnatten, M. Quantitative comparison of deep learning-based image reconstruction methods for low-dose and sparse-angle CT applications. *J. Imaging*, 7(3), 2021.

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations (ICLR)*, 2020.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Muckley, M. J., Riemenschneider, B., Radmanesh, A., Kim, S., Jeong, G., Ko, J., Jun, Y., Shin, H., Hwang, D., Mostapha, M., Arberet, S., Nickel, D., Ramzi, Z., Ciuciu, P., Starck, J.-L., Teuwen, J., Karkalousos, D., Zhang, C., Sriram, A., Huang, Z., Yakubova, N., Lui, Y., and Knoll, F. State-of-the-art machine learning MRI reconstruction in 2020: Results of the second fastMRI challenge. Preprint arXiv:2012.06318, 2020.

Ongie, G., Jalal, A., Baraniuk, R. G., Metzler, C. A., Dimakis, A. G., and Willett, R. Deep learning techniques for inverse problems in imaging. *IEEE J. Sel. Areas Inf. Theory*, 1(1):39–56, 2020.

Putzky, P. and Welling, M. Recurrent inference machines for solving inverse problems. Preprint arXiv:1706.04008, 2017.

Ramzi, Z., Ciuciu, P., and Starck, J.-L. XPDNet for MRI reconstruction: an application to the fastMRI 2020 brain challenge. Preprint arXiv:2010.07290, 2020.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F. (eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2015*, pp. 234–241. Springer Cham, 2015.

Schlemper, J., Oksuz, I., Clough, J. R., Duan, J., King, A. P., Schnabel, J. A., Hajnal, J. V., and Rueckert, D. dAUTOMAP: decomposing AUTOMAP to achieve scalability and enhance performance. Preprint arXiv:1909.10995, 2019.

Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.*, 61:85–117, 2015.

Shlezinger, N., Whang, J., Eldar, Y. C., and Dimakis, A. G. Model-based deep learning. Preprint arXiv:2012.08405, 2020.

Sidky, E., Lorente, I., Brankov, J. G., and Pan, X. Do CNNs solve the CT inverse problem? *IEEE Trans. Biomed. Eng.*, 68(6):1799–1810, 2021a.

Sidky, E., Pan, X., Brankov, J., Lorente, I., Armato, S., Drukker, K., Hadjiyski, L., Petrick, N., Farahani, K., Munbodh, R., Cha, K., Kalpathy-Cramer, J., Bearce, B., and AAPM Working Group on Grand challenges. Deep Learning for Inverse Problems: Sparse-View Computed Tomography Image Reconstruction (DL-sparse-view CT). URL: https://www.aapm.org/GrandChallenge/DL-sparse-view-CT/, 2021b.

Sidky, E. Y. and Pan, X. Report on the AAPM deep-learning sparse-view CT (DL-sparse-view CT) Grand Challenge. *Med. Phys.*, accepted, preprint arXiv:2109.09640, 2021.

Sriram, A., Zbontar, J., Murrell, T., Defazio, A., Zitnick, C. L., Yakubova, N., Knoll, F., and Johnson, P. End-to-end variational networks for accelerated MRI reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 64–73, 2020.

Tirer, T. and Giryes, R. On the convergence rate of projected gradient descent for a back-projection based objective. *SIAM J. Imag. Sci.*, 14(4):1504–1531, 2021.

Willemink, M. J. and Noël, P. B. The evolution of image reconstruction for CT – from filtered back projection to artificial intelligence. *Eur. Radiol.*, 29(5):2185–2195, 2019.

Wu, Y. and He, K. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Würfl, T., Ghesu, F. C., Christlein, V., and Maier, A. Deep learning computed tomography. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pp. 432–440, 2016.

Yang, Y., Sun, J., Li, H., and Xu, Z. Deep ADMM-Net for compressive sensing MRI. In *Advances in Neural Information Processing Systems 29*, pp. 10–18, 2016.

A. Exact AAPM Challenge Setup

To ensure reproducability, we give an exact account of how we trained our winning submission to the AAPM challenge (team-name: `robust-and-stable`). Since the systematic investigation of the ItNet-architecture was conducted after the challenge submission phase, it became clear that not all of the substeps outlined below have a notable impact on the performance (see also Section 4).

The following details are related to Step 3 of Section 3 ("Constructing an Iterative Scheme"). We start by training an ItNet$_4$ (with weight sharing) for 500 epochs of mini-batch stochastic gradient descent and Adam with an initial learning rate of $8 \cdot 10^{-5}$ and a batch size of 2 (restarting Adam after 250 epochs). Then, we improve the accuracy by the following *post-training* strategy: First, the ItNet$_4$ is extended by one more iteration:

$$\text{ItNet-post}[\boldsymbol{\theta}] \colon \mathbb{R}^m \to \mathbb{R}^N,$$

$$\boldsymbol{y} \mapsto \left[\bigcirc_{k=1}^{5} \left(\mathcal{DC}_{\lambda_k, \boldsymbol{y}} \circ \text{UNet}[\tilde{\boldsymbol{\theta}}_k] \right) \circ \text{FBP} \right] (\boldsymbol{y}), \tag{7}$$

where $\tilde{\boldsymbol{\theta}}_k$ is initialized with the optimized weights from ItNet$_4$ for $k = 1, \dots, 4$, and we set $\tilde{\boldsymbol{\theta}}_5 := \tilde{\boldsymbol{\theta}}_4$. Next, ItNet-post is fine-tuned by keeping the weights $\tilde{\boldsymbol{\theta}}_1 = \tilde{\boldsymbol{\theta}}_2 = \tilde{\boldsymbol{\theta}}_3$ of the first three UNet-blocks fixed and optimizing only over the weights of the last two iterations (without weight sharing).

Aiming at an additional training speed-up, we use the initialization $\boldsymbol{\lambda} = [1.1, 1.3, 1.4, 0.08]$ for the data-consistency parameters of ItNet$_4$, which was found by pre-training. Similarly, ItNet-post is initialized with the optimized values from ItNet$_4$ for $k = 1, 2, 3$, together with $\lambda_4 = 1.0$ and $\lambda_5 = 0.1$.

To improve the overall performance of our networks, we have additionally applied the following "tricks" for *fine tuning*, which are ordered by their importance:

(i) Due to statistical fluctuations, the networks typically exhibit slightly different reconstruction errors, despite using the same training pipeline. The final reconstructions are therefore computed by an *ensemble* of ten networks, each trained on a different split of the training set.

(ii) Due to the training with small batch sizes, we replace batch normalization of the UNet-architecture by *group normalization* (Wu & He, 2018).

(iii) We equip the UNet-blocks with a few *memory channels*, i.e., one actually has that $\text{UNet}[\boldsymbol{\theta}] \colon \mathbb{R}^N \times (\mathbb{R}^N)^{c_{\text{mem}}} \to \mathbb{R}^N \times (\mathbb{R}^N)^{c_{\text{mem}}}$; cf. Putzky & Welling (2017); Adler & Öktem (2018). While the original image-enhancement channel is not altered, the output of the additional channels is propagated through the ItNet, playing the role of a hidden state (in the spirit of recurrent neural networks). For our experiments, we use $c_{\text{mem}} = 5$.

(iv) It was beneficial to restart occasionally the training of the networks (see also Fig. 9).

The following modifications did not lead to a gain in performance and were omitted:

(i) Improving `FBP` in Step 1 of Section 3 by making some of it components learnable (e.g., the filter), cf. Würfl et al. (2016). Although this is advantageous for the reconstruction quality of `FBP` itself, it leads to worse results for the ItNet. This suggests that a combination of model- and data-based methods benefits most from precise and unaltered physical models.

(ii) Adding additional convolutional-blocks in the measurement domain of ItNet.

(iii) Modifying the standard ℓ^2-loss by incorporating the RMSE or the ℓ^1-norm.

(iv) Utilizing different optimizers such as SGD, RAdam (Liu et al., 2020), or AdamW (Loshchilov & Hutter, 2019).

In Fig. 9, we visualize the RMSE loss curves of our training pipeline, i.e.,

$$\text{UNet} \circ \text{FBP} \to \text{ItNet}_4(+\text{restart}) \to \text{ItNet-post}(+2 \times \text{restart}).$$
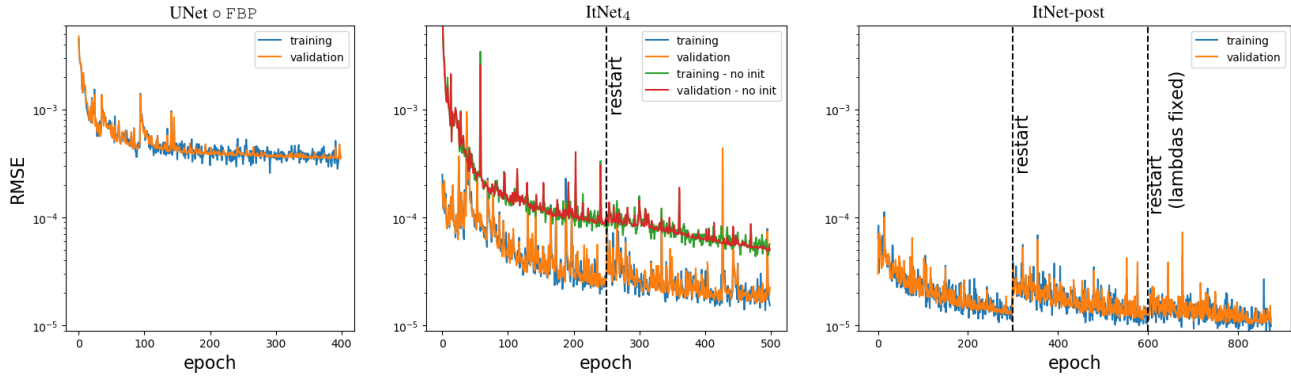
Figure 9. **Loss curves and network training.** The first two plots demonstrate that ItNet$_4$ improves the RMSE by approximately an order of magnitude in comparison to a post-processing by UNet. Furthermore, the gain of our UNet-initialization strategy can be seen in the second graph. The last two plots illustrate the advantages of restarting and of the post-training strategy, respectively. Note that we display the RMSE on the training and validation sets instead of the actual ℓ^2-losses, which behave similarly.

B. The Effect of Weight Sharing

General aspects of weight sharing for unrolled algorithms have been extensively discussed in the literature, e.g., see (Aggarwal et al., 2018; Hammernik et al., 2021). Fig. 10 gives some insights in the context of our specific approach. It clearly indicates that weight sharing also changes the reconstruction dynamic within the neural networks. Earlier iteration steps of the weight-shared ItNets are more effective, while the non-weight-shared counterparts draw most of their performance from the later steps. This suggests a trade-off between increasing the model capacity and the difficulty of optimizing the resulting network, while weight sharing forms a simple remedy; cf. Hammernik et al. (2021). However, we conjecture that an improved training strategy for the non-weight-shared networks might unlock the potential of the early-step UNet-blocks. This could lead to an even larger performance gap between the final reconstruction accuracy of ItNets with and without weight sharing. A systematic study of this aspect is left to future research.
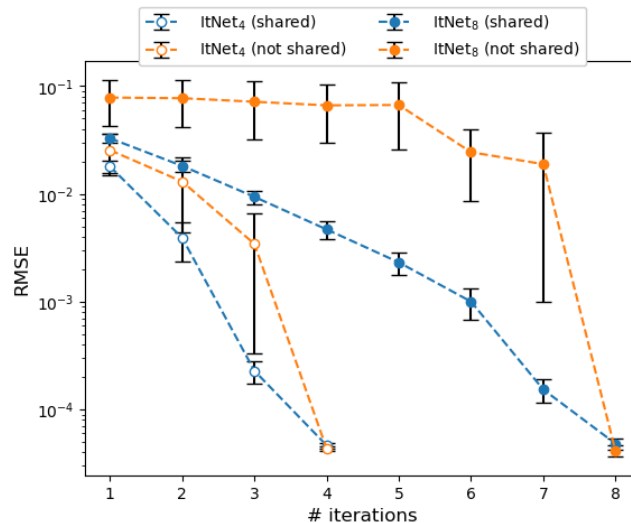

Figure 10. **A look inside.** Accuracy of ItNet$_4$ and ItNet$_8$ with (blue) and without (orange) UNet weight sharing when using only the first k iteration steps and discarding the rest. The mean RMSE ($\pm$ std. dev.) on a hold-out evaluation data set is reported over 5 different training/validation splits. The original AAPM challenge data was subsampled to the resolution 256×256 for this experiment.

C. Performance on Real-World Image Data

In order to assess the effectiveness of our method on real-world images and noisy (but still simulated) measurements, we have applied it to the *low-dose parallel beam (LoDoPaB)* CT dataset (Leuschner et al., 2021). This dataset is part of a past challenge and was successfully used to benchmark various deep-learning-based reconstruction schemes. It consists of 42895 two-dimensional human chest CT slices and their low-intensity measurements, see Leuschner et al. (2021) for details on the low-dose setup. For our case study, we have applied Step 2 and Step 3 of the methodology in Section 3.[9] The resulting ItNet has reached the first place in the public leaderboard (still open for submissions),[10] thereby outperforming various other methods, such as the learned primal-dual algorithm (Adler & Öktem, 2018) (cf. Table 1). A brief analysis and visualization of our reconstructions results can be found in Fig. 11. Overall, we conclude that our solution strategy can also achieve state-of-the-art performance on natural image data.
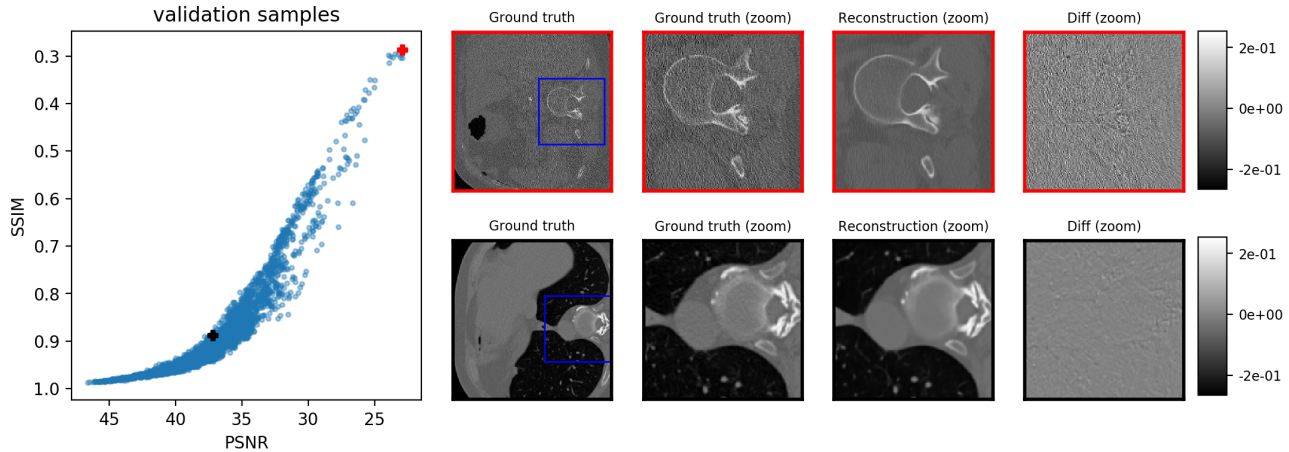

Figure 11. **Results for LoDoPaB CT.** The plot on the left-hand side visualizes the reconstruction performance of the ItNet with respect to the challenge metrics (SSIM and PSNR) where each blue point corresponds to one image in the LoDoPaB CT validation set (3522 images). We also show individual reconstructions for the red point (worst SSIM) and black point (closest to average SSIM and PNSR) on the right-hand side. Most notable is that the poor SSIM value of the red point is rather due to a low-quality ground-truth image, than a low-quality reconstruction.

---

[9]We have trained an ensemble of five ItNet$_3$ with initial learning rate of $8 \cdot 10^{-5}$ and batch size 2. As loss function, a combination of the MSE and SSIM was used.

[10]Team-name: `RobustAndStable`; public leaderboard on `https://lodopab.grand-challenge.org` (accessed on June 7, 2022).