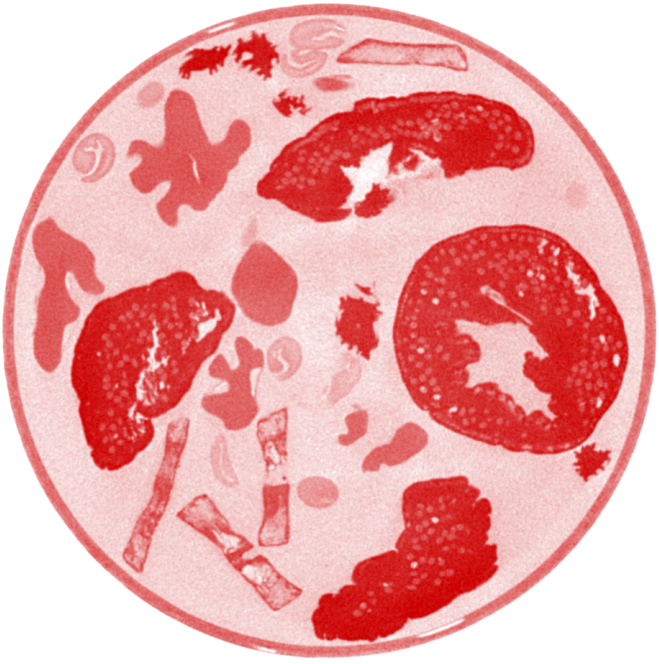
Advancing learned algorithms for 2D X-ray computed tomography

Kiss, M.B.



6

Benchmarking Learned Algorithms for Computed Tomography Image Reconstruction Tasks

Computed tomography (CT) is a widely used non-invasive diagnostic method which is applied in various fields such as medicine, materials science, industrial testing, and cultural heritage research. Based on X-ray projection images acquired from 360 degrees, cross-sectional images of an object or patient can be calculated using computer algorithms. For reconstructions of CT data that is, for example, badly sampled (limited- or sparse-angle), low-dose, or exhibits artifacts caused by e.g. metals or other dense materials, typically, a variety of image processing techniques are necessary for quality improvement.

The rise of deep learning [122] and widespread availability of large scale computing systems have led to substantial advances in computer vision, including tasks such as object detection, classification, segmentation or image denoising. A key enabler was the release of corresponding large-scale, open-source datasets such as MNIST [123], CIFAR [119] and ImageNet [46], which helped the research community to hill-climb on standardized benchmarks and continuously advance the state-of-the-art. The last five

This chapter is based on:

M. B. Kiss, A. Biguri, Z. Shumaylov, F. Sherry, K. J. Batenburg, C.-B. Schönlieb, and F. Lucka. "Benchmarking learned algorithms for computed tomography image reconstruction tasks". *Applied Mathematics for Modern Challenges* 3.0 (2025), pp. 1–43.

years have also seen a rapid development of machine learning approaches specifically for CT image reconstruction [4, 11, 40, 88, 125, 126, 157, 179, 199, 213, 218], which hold great potential for further reducing patient dose, speeding up acquisitions, and improving image quality in challenging acquisition settings [29, 71].

Despite increasing research activity at the intersection of CT and machine learning, the field of CT image reconstruction still lacks large-scale, open-access, real-world datasets that employ standardized evaluation metrics and benchmarking baselines. Many CT studies use datasets that are either not openly available to the research community or largely consist of synthetic data, which may suffer from the broadly observed sim-to-real gap [152, 205]. Furthermore, most of them use different pre-processing pipelines and datasets of various sizes. This hinders the comparison of different state-of-the-art methods and makes reproducing as well as validating results a cumbersome and challenging task.

Early computer vision algorithms were often developed using small-scale datasets under lab conditions and showed a significant lack of generalization in the real world. The ability to generalize improved significantly with the emergence of large-scale datasets, which initially consisted of images sourced from the internet and later expanded to encompass increasingly unstructured data. Today, we have access to massive multi-modal (visual and language) datasets scrapped off the internet, enabling foundational models and large language models (LLMs) to achieve unprecedented levels of information processing and synthesis. Accordingly, developing a large-scale benchmarking dataset for 2D computed tomography may be a first step to enabling similar breakthroughs in data-driven CT image reconstruction.

In this paper, we utilize real-world experimental data instead of simulated CT data and design standardized experiments for various common CT reconstruction tasks, allowing for more systematic and standardized comparisons between learned algorithms on one unified dataset. The contributions of this paper are: (i) a benchmarking study for a fixed set of data-driven methods on a recently published dataset of real-world experimental measurements, the 2DeteCT dataset [98]; (ii) a toolbox for benchmarking that enables seamless addition of new methods; (iii) an option to load the 2DeteCT dataset differently within the toolbox for extensions to other problems and different CT reconstruction tasks. This work provides a starting point for the community to develop, test, and compare new methods on real-world experimental data in a straightforward and reproducible way, which can shorten the overall development time of new data-driven CT image reconstruction algorithms considerably.

In the remainder of this paper, we give a brief overview of related work in the field of CT datasets and present details about the mathematical foundation of data-driven CT reconstruction. We provide a short categorization of learning-based methods for solving inverse problems such as CT image reconstruction to give context for the information content of the benchmarking framework. Afterwards, we introduce the benchmarking design, including the various CT image reconstruction tasks, the data pipeline, and the performance metrics. Subsequently, we elaborate on the employed pre-processing of the benchmarking dataset, the evaluated methods of our numerical

experiments, and their training details. After presenting the benchmarking results we discuss the limitations of the dataset, its broader impact, and the code and data availability.

Table 6.1: A summary of publicly available CT datasets, supported tasks, their size, and their raw data availability. (✓) = possible through data generation.

Dataset	CT Image Reconstruction Tasks				Size (>100 samples)	Raw Data
	Low-Dose	Limited-Angle	Sparse-Angle	Beam-hardening reduction		
Mayo [141, 144]	✓	(✓)	(✓)	✗	✗[141] / ✓[144]	✗
LoDoPaB [124]	✓	(✓)	(✓)	✗	✓	✗
ICASSP GC8 [20]	✓	(✓)	(✓)	✗	✓	✗
Walnut CBCT [47]	✗	✓	✓	✗	✗	✓
2DeteCT [98]	✓	✓	✓	✓	✓	✓

6.1 Related work

Computer science has played a vital role in overcoming limitations of traditional imaging systems such as CT and magnetic resonance imaging (MRI). In combination with applied mathematics and advanced engineering, computer science forms the field of computational imaging. The main goal of computational imaging is to improve image quality, enhance resolution, enable novel imaging capabilities, and extract valuable information or hidden details and features that may not be directly visible by traditional imaging methods.

The field has undergone many technological advances throughout the last 15 years [68, 84, 171, 200] but the most recent focus has been on employing machine learning techniques [199]. Despite their clear necessity, the computational imaging field to date offers few large-scale datasets and benchmarks on real-world experimental data. Researchers of NYU and Facebook recently addressed this need for the case of MRI scans by publishing raw measurement data in their fastMRI [116] dataset, while the field of CT still lacked an open-access dataset of comparable scope (cf. Table 6.1). Acquiring such data in the medical sector presents particular challenges, including the radiation exposure patients receive from multiple CT scans and the lack of access to raw measurement data from commercial CT scanners. Previous attempts in the field of low-dose CT, such as the Mayo Clinic low-dose CT challenge of 2016 [141] and 2021 [144], the LoDoPaB dataset [124], and the IEEE ICASSP Grand Challenge 8 [20] have sought to bridge this gap, but relied on simulated data. These issues are further exacerbated due to the lack of raw projection data along with corresponding reconstructed image slices. Although the second release of the Mayo Clinic low-dose CT challenge of 2021 [144] already released raw projection data, the noise for these low-dose datasets remains simulated.

Only recently, the 2DeteCT dataset [98] overcame these shortcomings by providing raw measurement data with complementary features that can be used for a wide range of imaging tasks such as supervised or unsupervised denoising, limited- and sparse-angle scanning, beam-hardening reduction, super-resolution, region-of-interest

tomography or segmentation. In contrast to the clinical, in-vivo datasets such as the LIDC-IDRI [10] or the Mayo Clinic [141, 144], the 2DeteCT dataset would be categorized as an in-vitro dataset that only simulates the behavior of natural tissue.

Having one joint dataset instead of individual datasets of various research groups helps to train algorithms on a uniform set of data, to test them in a standardized way, and to compare them against other algorithms for different imaging tasks. Particularly, the problem of defining a ground truth or "gold standard" is prevalent in CT imaging. Usually, it involves some sort of choice or trade-off with respect to the image acquisition or generation whereas for the 2DeteCT dataset the "mode 2" acquisition provides clean data since its acquisition was designed in a high-resolution setting with an over-sampling in the number of angular projections, a high-dose tube setting, and with a beam filtration in place. Therefore, we treat the reference reconstructions of the 2DeteCT dataset as a ground truth or "gold standard" in this work. They utilize a Nesterov accelerated gradient descent (AGD) algorithm on a bigger reconstruction plane and subsequently crop the resulting reconstructions to their center region. These reconstructions can be used as target images for matching noisy or artifact-inflicted measurements of "mode 1" and "mode 3" respectively and for limited- or sparse-angle measurement data extracted from "mode 2". A more detailed description of these acquisition modes can be found in section 6.5.1 and a visualization is presented in Figure 6.1. This figure also illustrates types of artifacts present in each reconstruction, highlighting the diverse challenges that data-driven CT reconstruction must address.

6.2 Data-driven CT reconstruction

In this section, we present the mathematical background of data-driven CT reconstruction, specifically focusing on 2D tomography, i.e. reconstructing 2D slices from 1D projection data. Furthermore, we briefly introduce how the four classes of methods explored in this work fit within this framework.

6.2.1 Tomographic reconstruction as a linear inverse problem

Tomographic reconstruction is an inverse problem which can be described as an image recovery task based on measurements obtained through the Radon transform: $y(\ell) = \int_\ell x(z)\,\mathrm{d}z, \ell \in \mathcal{L}$. In this equation, $\mathcal{L}$ represents the lines in $\mathbb{R}^2$ from the X-ray source to each detector pixel, defined by the scanner geometry and rotation. Typically, this problem is linearized and discretized as

$$Ax + \tilde{e} = y \tag{6.1}$$

where A represents the so-called *forward operator* which encapsulates the integral computations over these lines. Here, A is a matrix where each row corresponds to a line integral over the pixel grid of the object. In this context, x is a vector representing the pixel values of the image, y is a vector representing the measured sinogram values, and $\tilde{e}$ accounts for the noise or error, which may arise from the measurements themselves

or from the linearization of the operator.

Classically, to solve the inverse problem in Eq. 6.1 in a robust manner, a variational regularization approach [55, 176] is employed. The reconstruction is defined by the following minimization problem of the variational objective:

$$\hat{x} = \arg\min_{x} \{\mathcal{D}(y, Ax) + \mathcal{R}(x)\}, \tag{6.2}$$

where $\mathcal{D}$ measures the data fidelity between the measurement and the reconstructed image (most commonly the L^2-distance in CT) and $\mathcal{R}$ is a regularization function that promotes images of desired properties. The data fidelity term $\mathcal{D}$ is usually chosen according to the noise distribution, and a good choice of regularizer $\mathcal{R}$ is important for achieving accurate results. Traditionally, regularization functionals were hand-crafted to encourage the reconstruction x to have structures known to be realistic.

In practice, Eq. 6.2 is solved using iterative optimization schemes, and the quality of reconstructions is largely influenced by the choice of the regularization functional $\mathcal{R}$. A variety of methods have been proposed in the optimization literature to solve Eq. 6.2 with particular choices for the data fidelity term $\mathcal{D}$ and the regularization term $\mathcal{R}$, often under the assumption that these functions are convex. In certain cases, these optimization methods yield superior reconstructions compared to the standard analytical approach of the inverse Radon transform, known as filtered backprojection, especially when suitable functions and parameters are selected.

While convexity of $\mathcal{R}$ is analytically desirable for providing efficient optimization schemes with various guarantees, it is often observed that non-convex regularizers yield superior reconstructions in practice. However, this advantage comes at a price: finding global minima becomes generally infeasible, and sometimes even finding stationary points cannot be guaranteed.

6.2.2 Data-driven methods for tomographic reconstruction

In response to the limitations of classical knowledge-driven approaches, data-driven methods have rapidly advanced over the past decades. These methods can be categorized in different ways, including based on the amount of expert knowledge involved, which components are parameterized as neural networks, the domain of application, and the methodological approach employed (cf. Table 6.2).

In this work, we follow the general categorization of supervised learning methods of Arridge et al. [11] and consider the following methods in our benchmarking design: post-processing methods, learned/unrolled iterative methods, learned regularizer methods, plug-and-play methods.
While any strict categorization may overlook or misrepresent certain methodologies from the literature, such as self-supervised learning strategies, this framework effectively encompasses the majority of techniques found in the (weakly) supervised learning literature on data-driven CT reconstructions.

In the following section, we briefly introduce and describe each of the method cate-

Table 6.2: Method Categorizations

Article	Categorization
Zhang et al., 2020 [225]	Amount of expert knowledge involved: hand-crafted, hybrid approaches, mostly learned
Ye et al., 2023 [219]	Point of learned processing: pre-processing, post-processing, and raw-to-image
Ravishankar et al., 2019 [166]	Domain of application: image-domain, hybrid-domain, AUTOMAP [230], sensor-domain
Arridge et al., 2019 [11]	Methodological: post-processing methods, Learned / Unrolled Iterative Methods, Learned Regularizer Methods, Plug-and-Play Methods

gories for historical and methodological context. It is not intended as a state-of-the-art review of methods in the literature.

Direct solvers: Directly learning a reconstruction from measurement y as $\hat{x} = \mathcal{N}_\theta(y)$ has been proposed for MRI in AUTOMAP [230]. Recently, a similar direct method has been proposed for CT [64], but due to the limited success of this approach, we do not explore it further.

Post-processing methods: A straightforward approach to incorporating data-driven methods to overcome the ill-posed nature of CT reconstruction is to initially use classical method for reconstruction and subsequently train a network to learn the mapping from the manifold of inadequate reconstructions to the manifold ground truth reconstructions[75, 78, 88, 91, 228]:

$$\hat{x} = \mathcal{N}_\theta(\mathcal{F}(y)), \tag{6.3}$$

where $\mathcal{F}$ is a reconstruction (e.g. FBP in this work), and $\mathcal{N}_\theta$ an appropriately parameterized neural network (NN).

Learned/Unrolled iterative methods: This method category arises from the realization that many iterative solvers, such as FISTA [14], have a resemblance to convolutional neural networks (CNNs) [74]. Notably, Barbu [12] introduced the idea of unrolling iterative methods. By unrolling a handcrafted iterative algorithm and using it as a building block of a deep neural network (DNN), the parameterized mathematical operators inherit hyperparameters, image priors, and data consistency constraints from the iterative methods [219]. A common way to design these methods is to find a particular step of an iterative solver, e.g. a *proximal* step, and replace it with an iteration-dependent shallow CNN. A broader overview of these methods is given in the review by Monga et al. [145].

The resulting unrolled iterative methods have more interpretable architectures compared to traditional "black-box" denoisers, commonly exhibit much fewer trainable parameters than standard DNNs, and enable combining domain knowledge with deep learning [225]. However, it is important to acknowledge that while these methods may

be more interpretable, they lose their mathematical guarantees when incorporating learned networks. Since these methods utilize the operator within the network, they are often referred to as model-based networks. In this context, the physics of the model, represented by the operator A, is provided to the network rather than learned.

Learned regularizer methods: This approach is based on learning the regularization functional $\mathcal{R}$ in Eq. 6.2 from data. Traditionally, the regularization functional was hand-crafted for the problem, and over the past decades many hand-crafted functionals have been proposed; see Benning and Burger [16] for an overview. In contrast to standard DNNs, the minimization of the variational objective can be analyzed mathematically[149]. Examples include dictionary learning [39], generative [203], network Tikhonov [126], and adversarial regularization [183]. See Habring and Holler [76] or Dimakis et al. [48] for an overview.

Plug-and-Play methods: A special sub-case of learned regularizers is the field of Plug-and-Play (PnP) methods where proximal algorithms are used to optimize the inverse problem when either the data fidelity or regularization term is non-smooth. Two widely used iterative algorithms minimizing such composite functionals are the Alternating Direction Method of Multipliers (ADMM) [27] and the FISTA [14] which use proximal operators to avoid differentiating the non-smooth function. The proximal step in these algorithms can be replaced by a more general black-box denoiser ("plugged"-in) while the optimization algorithms run ("play") as before. This approach of PnP methods was developed by Venkatakrishnan et al. [194] and an overview of theory, algorithms, and applications can be found in a recent review by Kamilov et al. [90]. Although PnP methods are heavily inspired by variational approaches, the study of their properties as convergent regularization methods is an area that is still under active development, with some initial work establishing results in this direction [51, 80].

6.3 Benchmark design

Following best practices on reproducibility for benchmarks [204] we define the purpose and scope of our benchmark as providing the research community with a benchmarking framework based on a real-world experimental dataset under CC BY 4.0 license. The framework consists of a versatile toolbox and a pipeline to evaluate and compare different algorithms. The toolbox can be used to set up reproducible, reusable experiments for different image reconstruction and processing tasks in X-ray computed tomography. The methods selected cover the full range of common categories of supervised learning methods for solving inverse problems. For each category we implement three well-established methods and evaluate their respective performance. The parameters for the CT image reconstruction tasks represent common choices in the field and all methods are implemented in recent Python and PyTorch versions. The evaluation is done with key quantitative performance metrics such as the structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR). The results of the different methods are documented in Tables 6.5, 6.6, and 6.7 as well as in a visual overview for a selected image slice in Figure 6.2. All trained models are saved and made

available on GitHub. The LION toolbox https://github.com/CambridgeCIA/LION/, used for setting up the benchmarking experiments, enables future extensions by implementing other CT experiments or other ML-based methods. The codebase is open-source and licensed under GNU General Public License v3.0. The aim of the benchmarking design is threefold. Firstly, we give an overview of different categories of data-driven methods for CT image reconstruction. Secondly, we set-up an easy-to-use pipeline for implementing and testing algorithms on real-world experimental data. Thirdly, we provide a baseline comparison of the aforementioned data-driven methods on the most common CT image reconstruction tasks [219].

The benchmark design described in this work is unique in its combination of realism, dataset scale, variety of measurement settings and variety of reconstruction methods considered. Our major contribution is that we have developed a benchmarking framework that for the first time relies completely on real-world experimental data and investigates the whole range of common CT image reconstruction tasks. Existing studies on data-driven CT reconstruction usually focus on one of the method categories and a singular task for which a newly developed algorithm is compared to the most recent state-of-the-art and classical reconstruction methods. Often the data used for these assessments are not the same as the data the other method was tested on, e.g. the acquisition geometry, the sub-sampling, or the pre-processing might differ. All these factors limit their comparability and necessitate a benchmarking design with one common dataset and standardized CT reconstruction tasks as outlined below and visualized in Figure 6.1.

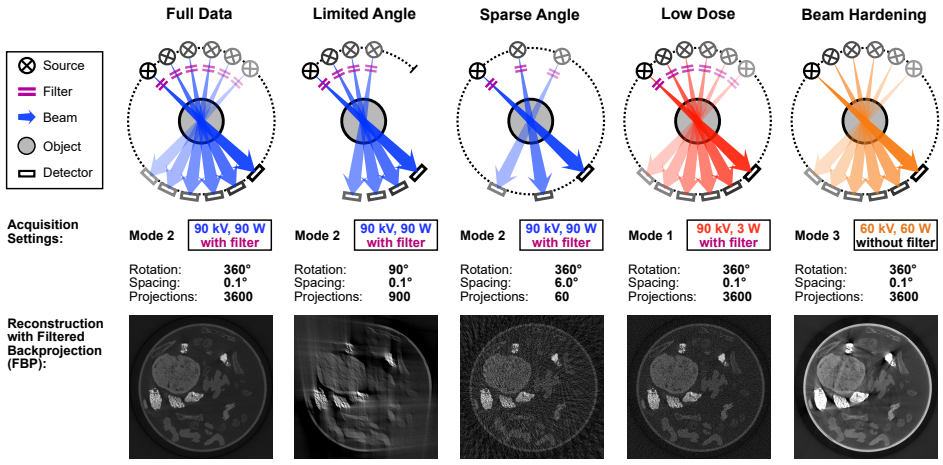


Figure 6.1: CT Image Reconstruction Tasks.

6.3.1 CT Image reconstruction tasks

2DeteCT encompasses raw experimental data of various acquisition modes for different CT image reconstruction tasks unified in one dataset. Defining and setting up CT image reconstruction tasks from the 2DeteCT dataset requires loading subsets of the experimental measurement data and defining the geometry for the reconstructions (see Table 6.3). For this work, we use the 2DeteCT sinograms of shape 956×3600 and reference reconstructions of shape 1024×1024.

Table 6.3: Summary of the acquisition parameters of the 2DeteCT dataset, adapted from [98]. The Thoraeus filter is a compound filter made of Sn 0.1mm, Cu 0.2mm, Al 0.5mm. The SOD and SDD values are based on the motor readings of the FleX-ray scanner which get translated into physical quantities and are subject to alignment errors.

Acquisition parameter	**Mode 1**	**Mode 2**	**Mode 3**
Tube voltage	90.0 kV	90.0 kV	60.0 kV
Tube power	3.0 W	90.0 W	60.0 W
Filters used	Thoraeus	Thoraeus	No Filter
Exposure time	50.0 ms		
Binned detector pixel size	149.6 µm		
Number of binned detector pixels	956		
Source to object distance (SOD)	431.020 mm		
Source to detector distance (SDD)	529.000 mm		
Number of projections	3601		
Angular increment	0.1 deg		

Each of the tasks utilizes different properties of the 2DeteCT dataset, necessitates a corresponding data pairing and requires a pre-processing which we will discuss in Section 6.4.1. The "gold standards" set as target images for all tasks are the reference reconstructions of "mode 2" of the 2DeteCT dataset. The details of these tasks are described in the following and visualized in Figure 6.1.

Full data reconstruction: We use the complete raw projection data of the "mode 2" acquisition (full sinograms) and use the corresponding iterative reference reconstructions of the 2DeteCT dataset as target images. Since these were generated through an AGD algorithm on a bigger reconstruction plane and subsequently cropping the resulting reconstructions to their center region, the evaluated methods actually learn to mimic the final output of AGD for this task. However, the full data reconstruction serves as a reference for the performance of the evaluated methods on other CT image reconstruction tasks.

Limited-angle reconstruction: We limit the raw projection data of the "mode 2" acquisition to a smaller angular range. Depending on the used wedge of 120°, 90°, or 60° only the first 1200, 900 or 600 projection lines of the 956×3600 sinograms are extracted. With this missing information a standard reconstruction will show streaking, elongation, ghost tail, and missing boundaries artifacts of increasing severity with decreasing angular size of the wedge [13, 62]. The "gold standard" is a reconstruction of a complete sinogram and therefore we use the iterative reference reconstructions of the 2DeteCT dataset as target images.

Sparse-angle reconstruction: We sub-sample the raw projection data of the "mode 2" acquisition. Depending on the number of angles used 360, 120, or 60 projection lines evenly distributed over the full angular range are extracted from the original 3600 projections of the sinograms. Undersampling of this kind violates the Nyquist-Shannon sampling theorem and introduces aliasing artifacts [89]. The "gold standard" is a reconstruction of a fully-sampled sinogram and therefore we use the iterative reference reconstructions of the 2DeteCT dataset as target images.

Low-dose reconstruction: We use the complete raw projection data of the "mode 1" acquisition (full sinograms). Due to the low-dose setting of the acquisition, the measurements show very low photon counts and their corresponding iterative reconstruction slices show streaking artifacts and high granularity as manifestations of image noise. The "gold standard" for this task is a reconstruction of the corresponding high-dose acquisition of "mode 2" and we use the iterative reference reconstructions of the 2DeteCT dataset as target images.

Beam-hardening corrected reconstruction: We use the complete raw projection data of the "mode 3" acquisition (full sinograms). Due to the unfiltered beam spectrum of the X-ray source during the acquisition of the measurements, the corresponding iterative reconstruction slices show streaking and shadowing-like "cupping" artifacts as manifestations of beam-hardening and photon starvation [35, 158]. The "gold standard" for this task is a reconstruction of the corresponding filtered acquisition of "mode 2" and we use the iterative reference reconstructions of the 2DeteCT dataset as target images.

6.3.2 Pipeline

The benchmark framework was set up using LION (Learned Iterative Optimization Networks), an open-source Python toolbox for learned tomographic reconstruction. There are other open-source libraries that focus on providing a robust set of forward (and backward) operators, like ODL [2] or tomosipo [81] or libraries that focus on providing variational (non-ML) reconstructions, such as ASTRA [1, 154, 190] or TIGRE [18, 19]. Given such existing libraries, obtaining an operator for the experiments described above, can be straightforward, but creating such operators for real measured datasets is not always trivial and loading this data accordingly can be challenging.

LION focuses on using libraries such as tomosipo to build a toolbox for loading, preprocessing, simulating, and reconstructing CT data as well as training and evaluating data-driven methods in CT. As forward operators, LION currently supports the aforementioned tomosipo, which uses ASTRA's ray-driven forward projector and an unmatched voxel-driven backprojector.

To our knowledge there is only one other maintained open-source library for deep learning that supports CT reconstruction, DeepInv [187]. DeepInv is a comprehensive library for inverse problems and learning. However, it does have limitations in terms of specificity, which is often overlooked by more generalist libraries. For instance, it does

not provide complex dataset definitions, topic-specific experiments, or application specific noise models that are crucial for certain applications.

Through this benchmarking study LION now features a designated data loader for the 2DeteCT dataset [98] and defines tasks and modes such that the data loader sources the corresponding data from this dataset. It performs a split into training (79.4%, 3930 slices), validation (11.11%, 550 slices), and test data (9.49%, 470 slices), defines the CT geometry and parameters of CT data processing, creates a forward operator, and loads and pre-processes sinograms and reconstructions. Furthermore, it contains designated experiment classes for the above CT image reconstruction tasks and PyTorch implementations of various models from the different method categories outlined above. A significant development effort was undertaken to ensure a consistent implementation of all deep learning methods using the same data and operator framework for the variety of CT image reconstruction experiments. Additionally, the toolbox defines metrics for training and evaluation, contains an optimizer for supervised learning settings and allows for saving all relevant information in a parameter file to completely reproduce models. These parameter files store among others, the used dataset parameters, training parameters, loss, epochs, optimizer, and the CT geometry. Lastly, the toolbox allows for saving trained models to compare against them and for storing scripts that have been used in papers to specifically reproduce experiments of that particular study.

To summarize, the motivation for this benchmarking design is providing the community with an easy pipeline to load real-world experimental data and conduct standardized experiments. It lays the foundation for the computational imaging community to easily implement and test methods on the 2DeteCT dataset using a custom data loader as well as tailored and standardized CT benchmarking experiments. This greatly extends the utility of the 2DeteCT dataset since researchers do not need to spend time on the implementation of their own data loaders or reconstruction tasks and can easily compare against other methods. It makes it easier than ever to start experimenting with deep-learning-based (and non-deep-learning-based) CT reconstruction in realistic settings, without the need for expert knowledge or simulating data. In particular, it allows users to avoid many of the pitfalls of trying to simulate appropriate measurement data and focus instead on the development of reconstruction methods.

6.3.3 Performance metrics

One common metric for evaluating CT reconstructions, especially in the case of limited or noisy data, is the peak signal-to-noise ratio (PSNR)[69]. It quantifies the ratio of the maximum possible value of a signal to the power of corrupting noise that affects the fidelity of the image. Furthermore, the structural similarity (SSIM) [180, 195, 202] indicates in a range from 0.0 to 1.0 how similar an evaluated image is to a reference image, where 1.0 means they are identical. For both metrics, higher scores indicate a better algorithm performance and a ground truth reference image is necessary. The ground truth reference images used in this work are the reference reconstructions

of "mode 2" of the 2DeteCT dataset which utilize an AGD algorithm on a bigger reconstruction plane which is then cropped to its center.

6.4 Numerical experiments

6.4.1 Pre-processing

All numerical experiments for the different CT image reconstruction tasks are set up as sinogram-to-reconstruction experiments. They do not perform sinogram-to-sinogram or reconstruction-to-reconstruction experiments such as sinogram denoising, artifact reduction, inpainting, beam-hardening reduction. This means that the ML-based algorithms take a sinogram as input data and corresponding iterative reference reconstructions from acquisition "mode 2" as target data. In principle, it is possible to also perform sinogram-to-sinogram or reconstruction-to-reconstruction experiments within the LION toolbox but this would make the comparison between e.g. post-processing methods and learned/unrolled iterative methods less fair. Therefore, we chose to only use sinogram-to-reconstruction experiments in this benchmarking study. The sinograms are pre-processed with LION using modules such as ASTRA [1, 154, 190] and tomosipo [81] according to the description in the original dataset publication and as outlined below.

The sinograms are pre-processed into a beam intensity loss image by subtracting detector offset counts ("dark currents") from the measured photon counts per detector pixel and by dividing by so-called "flat fields", the pixel-dependent sensitivities of the detector. To perform a CT reconstruction, the data is then transformed with the negative logarithm to follow the Beer-Lambert law. For more details please refer to the original dataset publication [98].

6.4.2 Evaluated methods

The method selection in this work focuses on (weakly) supervised learning methods, excluding self-supervised and unsupervised approaches to establish a foundation for benchmarking. To this end, we prioritize established supervised learning methods that can serve as reliable baselines, omitting some newer techniques based on transformers and generative models, as explained at the end of this subsection. In the following, we introduce which methods from the literature will be used in this benchmark. To limit the broad scope of this work, we use three methods from each subclass presented in Section 6.2.2.

Table 6.4 lists the different methods that have been evaluated for the benchmarking and in which of the described method categories they fall. The details of the networks' training can be found in Section 6.4.3 and in the GitHub repository mentioned in the section "Code and Data Availability". Additionally, we evaluate classical reconstruction methods on the test data. Analytical methods such as filtered backprojection (FBP) or iterative methods such as AGD [151] or regularized methods such as the Chambolle-

Pock (PDHG) [37] solver with total variation (TV) regularization are highly effective and still widely used in practice [15].

Table 6.4: Evaluated methods

Category	Method (Year and Reference)
Classical Methods	FBP [79], AGD [151], PDHG [37]
Post-Processing Methods	U-Net [173], MSD-Net [160], DnCNN [227]
Learned / Unrolled Iterative Methods	Learned Gradient [4], TV-regularized Learned Gradient, Learned Primal Dual [3]
Learned Regularizer Methods	AR [134], TDV [117], ACR [147, 148]
Plug-and-Play Methods	DnCNN-PnP [227], DRUNet-PnP [226], GS-PnP [85]

Post-processing methods: In this work, the evaluated post-processing methods are the U-Net [173], the MSD-Net [160], and the DnCNN [227]. The U-Net, originally designed for image segmentation tasks, is well known in many fields of machine learning and consists of a contracting path, which captures both low and high-frequency features, a bottleneck layer, and a symmetric expanding path. The expanding path integrates information from corresponding layers in the contracting path, effectively translating learned features back into the image space at each resolution. The MSD-Net has proven itself particularly effective for CT image reconstruction problems in the literature. Its neural network architecture, incorporating dense connections between layers at different scales, helps to effectively capture both local and global information in images. The DnCNN is a deep learning architecture specifically designed for image denoising tasks. It uses a series of convolutional layers with batch normalization to learn noise patterns and remove them from images. It has achieved state-of-the-art results in image denoising benchmarks.

Learned / Unrolled iterative methods: The first unrolled method we consider is Learned Gradient (LG) [4], which seeks to directly learn the update step in the gradient descent solver rather than relying solely on an additive step. The LG method parameterizes a fixed number of gradient steps to approximate the optimal direction based on the true gradient of the variational objective. The second method is an extension of this model including a TV regularization in its update rule (LGTV). In both of these methods, a small four-layer CNN is employed to replace the update step, taking the current image estimate and gradient(s) as inputs. The last method considered is the Learned Primal Dual (LPD) algorithm [3]. LPD solves the variational optimization problem by learning proximal-like steps in both primal and dual variables simultaneously. By jointly training two networks to update primal and dual variables, this algorithm can efficiently solve tasks such as image reconstruction and denoising and is known to produce high-quality results, using a minimal set of learned parameters. In LPD, each gradient step is substituted by a shallow four-layer CNN.

Learned regularizer methods: Learned regularization methods, which directly parameterize the regularization functional using a neural network, typically differ in either their training strategy, network architecture, or variational objective optimization scheme. For evaluation, the adversarial regularizer (AR)[134] and its convex coun-

terpart (ACR)[147, 148] are considered alongside total deep variation (TDV)[117]. Both adversarial regularizers are trained using a Wasserstein-1 distance-based loss, while TDV is trained by minimizing the distance between the ground truth and the reconstruction achieved via a fixed number of gradient steps on the variational objective. AR is parameterized using a standard CNN with a single dense layer, and the variational objective is optimized via accelerated gradient descent with early stopping. ACR utilizes an input convex neural network, and the variational objective is optimized with accelerated gradient descent and backtracking. TDV is parameterized using a multiscale convolutional neural network.

Plug-and-Play methods: There are various axes along which the settings of PnP methods can be varied, including the choice of splitting method and the architecture of the denoiser. In this work, we will fix the splitting to be a forward-backward splitting of a variational objective, and consider the effect of varying the denoiser architectures: two of them will be "unconstrained", differing mainly in model capacity, while the last one has a structural constraint that allows for provable convergence. To be more specific, the first method (DnCNN-PnP) replaces the proximal operator of the regularization functional by DnCNN [227], while the second method (DRUNet-PnP) uses a DRUNet [226] instead, which gives improved denoising performance at the cost of significantly more parameters and increased computational time. Finally, the third method (GS-PnP) splits the variational objective in the opposite way, taking a gradient step on the regularization functional and a proximal step on the data discrepancy functional. It has been shown that it is possible to obtain high-quality PnP reconstructions in this way, while retaining the interpretation of minimizing a variational objective [85]. To compute the output of the denoiser in GS-PnP, it is necessary to perform an intermediate backpropagation on the backbone denoiser, resulting in significant extra computational cost, both in terms of memory and time. As in the work of Hurault, Leclaire, and Papadakis [85], we deal with this by scaling down (both in number of blocks and width of the blocks) the backbone DRUNet, as compared to the DRUNet used in DRUNet-PnP.

The field of deep learning methods is rapidly evolving, with new architectures and methods constantly being released. For this reason, we necessarily have had to omit some methods from consideration in this benchmark such as the most recent strides using transformers and diffusion models for CT reconstruction.

In the context of CT reconstruction, transformer-based reconstructions [198] generally take the form of what we have called a post-processing method in the benchmark, the only difference being the architecture of the "denoiser" used. In our benchmark, all of the architectures considered were convolutional. On the other hand, diffusion-model-based approaches [133, 186] are most similar to what we have called plug-and-play methods, as they alternately implement data-consistency steps, utilizing the forward model, and prior-consistency steps. In deterministic plug-and-play the latter involves applying a denoiser or executing a sampling step in stochastic restoration based on diffusion models.

In summary, we believe that our selection of method classes encompasses many methods of interest, even if some specific methods mentioned are not included in the presented comparison of twelve exemplary well-established approaches.

6.4.3 Training details

The basis for this benchmarking framework is the 2DeteCT dataset [98]. This dataset was split in a sophisticated way to ensure that no scanned sample mixes are shared between the training, validation, and test data. The data split is as follows: training data (79.4%, 3,930 slices), validation data (11.11%, 550 slices), test data (9.49%, 470 slices). The training was carried out without extensive hyperparameter tuning to achieve as good a result as possible. We prioritized adequate performance over extensive hyperparameter tuning to produce baseline results for a comparative analysis among techniques. The reported total training times for each method vary significantly, reflecting the differences in data size associated with each CT image reconstruction task.

Post-processing methods

The post-processing methods have all been trained with the same parameters: Adam optimizer [97] for 100 epochs with a learning rate of 10^{-4} and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The final models were chosen based on the minimum loss in the validation set. The total training times are dependent on the used machine, the CT image reconstruction task and the evaluated method:

- FBP+U-Net, total training time per CT image reconstruction task ranges between $26 - 73$ hours,
- FBP+MSD-Net, total training time per CT image reconstruction task ranges between $83 - 120$ hours,
- FBP+DnCNN, total training time per CT image reconstruction task ranges between $56 - 93$ hours,

Learned / Unrolled iterative methods

These methods have been trained exactly the same way as the post-processing methods, however, LG and LGTV required a learning rate of 10^{-5} for stable training. The final models were again chosen based on the minimum loss in the validation set. The total training times are dependent on the used machine, the CT image reconstruction task and the evaluated method:

- LG, total training time per CT image reconstruction task ranges between $24 - 117$ hours,
- LGTV, total training time per CT image reconstruction task ranges between $19 - 116$ hours,

- LPD, total training time per CT image reconstruction task ranges between $23 - 153$ hours,

Learned regularizer methods

All models were trained using an Adam optimizer with a learning rate of 10^{-4} and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$. These hyperparameters were picked according to the original methods [117, 148], and were fixed for all tasks. Order of magnitude for the hyperparameters in minimization of the variational objective were found for one tasks, the sparse-angle reconstruction with 90 projections, and were adjusted for other tasks based on the operator norm. The adversarial regularization methods have been trained for 25 epochs, with a reduced validation set. Minimum validation loss model was then chosen. TDV was trained for 10 epochs, due to its computationally expensive training. The number of steps was chosen to be the maximal number allowing for the network to fit on a 24 GB GPU. Due to the relatively low SSIM numbers we hypothesize that the number of steps ideally would need to be increased for all experiments, but due to the sizes, remains infeasible. The total training times are dependent on the used machine, the CT image reconstruction task and the evaluated method:

- AR, total training time per CT image reconstruction task ranges between $50 - 75$ hours,
- ACR, total training time per CT image reconstruction task ranges between $50 - 75$ hours,
- TDV, total training time per CT image reconstruction task ranges between $90 - 110$ hours,

Plug-and-Play methods

As above, we trained the denoisers for the PnP methods using the Adam optimizer, with a learning rate of 10^{-4} and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We normalized the inputs by rescaling with the maximum pixel value found on the training set. We trained denoisers on Gaussian denoising tasks with a range of noise levels $\{0.001, 0.005, 0.01, 0.02, 0.03, 0.05, 0.07\}$, corresponding to (average) PSNRs of approximately 50 dB, 36 dB, 30 dB, 24 dB, 20 dB, 17 dB and 13 dB respectively. After training, we plugged the denoisers into the methods described in Section 6.4.2, selecting separately for each experiment the denoiser that performed best on the PnP reconstruction task on the validation set. Each denoiser was trained for 25 epochs, with total training times per denoiser being as follows:

- DnCNN, total training time 6 hours,
- DRUNet, total training time 10 hours,
- GS-DRUNet, total training time 15 hours.

6.5 Results and discussion

We report on the performances of different data-driven methods on the most common CT image reconstruction tasks in both a quantitative (see Tables 6.5, 6.6, and 6.7) and qualitative analysis (see Figure 6.2). The metrics are averaged over the whole test dataset and include their standard deviation whereas the qualitative analysis is for one specific slice (index 182) of the test dataset. While an extensive quantitative analysis of the performance describing trends in different performances is out of the scope of this paper, we provide a detailed but short quantitative and qualitative analysis below. In CT image reconstruction the qualitative analysis, i.e. the visual inspection of reconstructed images, is a crucial tool to augment the quantitative results reported in Tables 6.5, 6.6, and 6.7 and shall serve as a starting point for further analysis.

6.5.1 Relevance and difficulty of CT image reconstruction tasks

The basis of the benchmarking framework of this work are the selected CT image reconstruction tasks. Namely, Full Data, Limited-Angle, Sparse-Angle, Low-Dose and Beam-Hardening corrected reconstruction. In the following, we want to give a better insight about their respective relevance and difficulty.

Full data reconstruction: The Full Data CT image reconstruction task can be considered purely as a reference for each of the algorithms and does not pose any particular challenges. This is due to the fact that this task uses the full data of the "mode 2" acquisition of the 2DeteCT dataset which was designed in a high-resolution setting with an over-sampling in the number of angular projections, a high-dose tube configuration, and with a beam filtration in place. Classical methods such as FBP, AGD, and PDHG will perform well in these settings and there is no need for learned reconstruction methods.

Limited-angle reconstruction: If this data is limited or sparsified in the angular range, undersampling artifacts occur. Theoretically, acquisitions from 180° with sufficient angular sampling can produce an artifact-free image. When limiting the angular range further, the missing information causes more visible image artifacts such as streaking, elongation, ghost tail, and missing boundaries. The challenge of limited angle acquisition occurs for example in industrial product inspection and medical imaging mammography. During the selection of the angular span of the wedge, we tested a limited angle of 150° for a few algorithms and decided that the task at hand is not yet challenging enough. Therefore, we chose wedges of 120°, 90°, or 60° for our limited angle reconstruction tasks. Since classical reconstructions of 60° are already dominated by artifacts, a further undersampling was omitted. The difficulty of these tasks increases with decreasing available angular range.

Sparse-angle reconstruction: For undersampling in terms of sparsity, the Nyquist-Shannon sampling theorem [89] can give an approximation of how many projections are necessary for a well-sampled CT scan. For the experimental setup of the 2DeteCT dataset a minimal number of approximately 3000 projections is required for sufficient

sampling. Noticeable differences, however, only occur when undersampling by factors of five or more. Decreasing the number of projections is often used to speed up the CT acquisition process or to reduce dose for the scanned subject or sample. We tested undersamplings based on 720, 360, 180, 120, 90, and 60 projections for one classical, one post-processing, and one unrolled algorithm to decide which experiments to include in the benchmark. Since 720 and 360 projections both showed a relatively similar severity in artifacts we chose to include only the 360 projections case in our benchmarking. For the lower end of this undersampling range we concluded that 60 projections, so an undersampling of a factor 60 in comparison to the full data, was still feasible for some of the tested learned algorithms and should be considered as a challenging task for our benchmarking. To distribute the number of projections for our sparse-angle reconstruction we chose 360, 120, and 60 projections for our final experiments. Again, the difficulty of these tasks increases with decreasing the number of available projections.

Low-dose reconstruction: For the low-dose CT image reconstruction task, we use the acquisition data of "mode 1" which uses a 1/30 tube current compared to "mode 2" acquisition that has been optimized for the best image quality. In medical imaging, a lower dose is typically chosen to achieve images of adequate quality for clinical purposes while minimizing radiation exposure. The "tube current"-"exposure time" products range from 50 to 400 mAs in clinical practice. "Mode 2" has a high-dose acquisition with a "tube current"-"exposure time" product of 18 mAs, while "mode 1" has a low-dose acquisition with a product of 0.6 mAs. It is important to note that the size of the scanning object and the setup geometry differ from a traditional medical CT scan, as the scanning object is much smaller with a circumference of approximately 35 cm compared to 50-100 cm for standard abdominal circumferences in children and adolescents [168]. Nevertheless, the low-dose CT reconstruction task can be viewed as having a similar or even higher noise-level than medical (extreme) low-dose CT data.

Beam-hardening corrected reconstruction: For the beam-hardening corrected CT image reconstruction task, we use the acquisition data of "mode 3". Since the beam spectrum of the X-ray source remains unfiltered during the acquisition process, the corresponding iterative reconstruction slices show streaking and shadowing-like "cupping" artifacts as manifestations of beam-hardening and photon starvation [35, 158]. These artifacts are of a highly non-local and non-linear nature and corresponding data of beam-hardening afflicted CT images and physically filtered and corrected acquisition data are a novelty introduced by the 2DeteCT dataset [98]. In both medical and industrial settings, having high-attenuating areas in the region-of-interest causes severe artifacts and poses challenges. The severity of the challenge of learning to map between these data distributions was to date unknown and is reported in this work for the first time.

Overall, the limited-angle reconstruction from 60° and the beam-hardening corrected reconstruction can be considered the most difficult tasks.

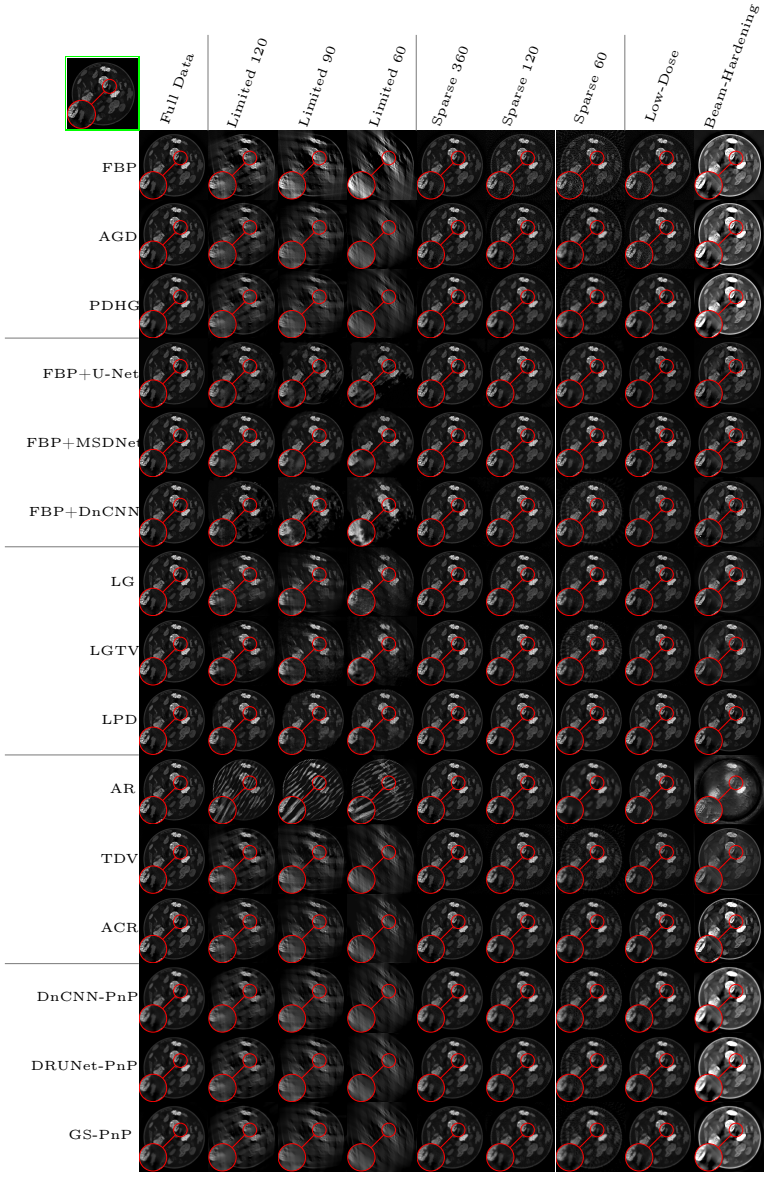


Figure 6.2: Qualitative analysis of all evaluated methods for slice 182 of the test dataset in comparison to the "gold standard" iterative reference reconstruction of the 2DeteCT dataset (green box).

Table 6.5: Quantitative analysis of all evaluated methods with respect to PSNR and SSIM on their performance in the CT image reconstruction tasks: Full Data "mode 2", Low-Dose "mode 1", and Beam-Hardening "mode 3".

Method	**Metric**	**CT Image Reconstruction Task**		
		Full Data	Low-Dose	Beam-Hardening
Classical Methods				
FBP	SSIM	0.7463 ± 0.0296	0.0838 ± 0.0212	0.3367 ± 0.0464
	PSNR	35.0285 ± 2.0907	18.6437 ± 2.0508	14.5594 ± 1.9056
AGD	SSIM	0.7753 ± 0.0380	0.0727 ± 0.0182	0.3483 ± 0.0650
	PSNR	**35.1006 ± 2.2801**	17.7062 ± 2.0517	14.4294 ± 1.9255
PDHG	SSIM	0.7689 ± 0.0498	0.6820 ± 0.0749	0.4492 ± 0.0616
	PSNR	34.3251 ± 1.9703	31.9637 ± 1.9824	15.0090 ± 1.9094
Post-Processing Methods				
FBP+U-Net	SSIM	0.6499 ± 0.0681	0.7632 ± 0.0780	0.6336 ± 0.0760
	PSNR	32.6998 ± 1.9512	27.6772 ± 3.0133	28.0629 ± 3.8439
FBP+MSDNet	SSIM	0.8481 ± 0.0384	0.7253 ± 0.0864	0.7991 ± 0.0665
	PSNR	33.2999 ± 1.9492	31.6910 ± 1.9559	31.4389 ± 2.0988
FBP+DnCNN	SSIM	0.8324 ± 0.0403	0.7127 ± 0.0806	0.6328 ± 0.0751
	PSNR	32.2575 ± 1.9506	29.7645 ± 1.9748	28.2405 ± 2.0374
Learned / Unrolled Iterative Methods				
LG	SSIM	0.7498 ± 0.0708	0.6685 ± 0.0772	0.6025 ± 0.0747
	PSNR	32.4409 ± 1.9527	30.4679 ± 1.9945	28.4148 ± 1.9584
LGTV	SSIM	0.8221 ± 0.0532	0.7008 ± 0.0797	0.6656 ± 0.0774
	PSNR	33.3312 ± 1.9493	30.9991 ± 1.9328	29.5372 ± 2.0008
LPD	SSIM	0.8447 ± 0.0370	**0.8282 ± 0.0519**	**0.8352 ± 0.0568**
	PSNR	33.3086 ± 1.9466	**32.6685 ± 1.9656**	**33.1382 ± 1.9759**
Learned Regularizer Methods				
AR	SSIM	0.8196 ± 0.0385	0.8039 ± 0.0505	0.3125 ± 0.0479
	PSNR	32.3583 ± 1.9621	31.0472 ± 1.9763	19.5692 ± 1.9612
TDV	SSIM	0.7282 ± 0.0652	0.6047 ± 0.0815	0.5494 ± 0.0635
	PSNR	33.1204 ± 1.9496	28.2429 ± 1.9433	25.7832 ± 2.0109
ACR	SSIM	0.8518 ± 0.0362	0.8163 ± 0.0522	0.6742 ± 0.0505
	PSNR	33.7131 ± 1.9450	32.2621 ± 1.9689	18.5708 ± 1.7824
Plug-and-Play Methods				
DnCNN-PnP	SSIM	**0.8585 ± 0.0402**	0.7795 ± 0.0523	0.5989 ± 0.0375
	PSNR	32.8506 ± 1.9306	31.3986 ± 1.9424	15.4667 ± 1.9107
DRUNet-PnP	SSIM	0.8573 ± 0.0405	0.7984 ± 0.0517	0.5945 ± 0.0375
	PSNR	32.8935 ± 1.9327	31.5762 ± 1.9480	15.4543 ± 1.9102
GS-PnP	SSIM	0.7856 ± 0.0590	0.7727 ± 0.0683	0.5131 ± 0.0562
	PSNR	32.5734 ± 1.9331	31.7931 ± 1.9444	15.3466 ± 1.9128

Table 6.6: Quantitative analysis of all evaluated methods with respect to PSNR and SSIM on their performance in the CT image reconstruction tasks of Limited-Angle.

Method	Metric	CT Image Reconstruction Task		
		Limited-Angle 120	Limited-Angle 90	Limited-Angle 60
Classical Methods				
FBP	SSIM	0.3418 ± 0.0354	0.2369 ± 0.0323	0.1557 ± 0.0288
	PSNR	22.4188 ± 1.9240	19.4251 ± 1.9405	16.9057 ± 1.9579
AGD	SSIM	0.4904 ± 0.0550	0.4411 ± 0.0555	0.4146 ± 0.0559
	PSNR	25.7508 ± 1.9690	24.1128 ± 1.9528	22.8848 ± 1.9531
PDHG	SSIM	0.5923 ± 0.0550	0.5194 ± 0.0547	0.4658 ± 0.0510
	PSNR	26.3646 ± 1.9443	24.4392 ± 1.9321	22.7556 ± 1.9428
Post-Processing Methods				
FBP+U-Net	SSIM	0.7251 ± 0.0519	0.6338 ± 0.0659	0.5892 ± 0.0678
	PSNR	28.7931 ± 2.0052	27.3875 ± 2.0441	23.8511 ± 2.8370
FBP+MSDNet	SSIM	0.7840 ± 0.0641	0.7695 ± 0.0579	0.7148 ± 0.0726
	PSNR	30.7850 ± 1.9784	29.0111 ± 1.9740	27.4624 ± 1.9661
FBP+DnCNN	SSIM	0.5829 ± 0.0521	0.5661 ± 0.0528	0.5174 ± 0.0559
	PSNR	20.4433 ± 2.5345	23.4066 ± 2.2764	21.2132 ± 2.5140
Learned / Unrolled Iterative Methods				
LG	SSIM	0.6740 ± 0.0652	0.5746 ± 0.0655	0.5378 ± 0.0706
	PSNR	28.1639 ± 1.9380	26.3188 ± 1.9508	24.7228 ± 1.9630
LGTV	SSIM	0.6804 ± 0.0647	0.5590 ± 0.0639	0.5091 ± 0.0643
	PSNR	28.4131 ± 1.9300	26.0867 ± 1.9545	25.0225 ± 1.9687
LPD	SSIM	**0.8296 ± 0.0410**	**0.8049 ± 0.0444**	**0.7724 ± 0.0571**
	PSNR	**31.1723 ± 1.9607**	**29.2534 ± 1.9941**	**28.0734 ± 1.9589**
Learned Regularizer Methods				
AR	SSIM	0.6869 ± 0.0505	0.6100 ± 0.0543	0.5742 ± 0.0620
	PSNR	23.8496 ± 2.1578	21.1830 ± 2.1772	22.2350 ± 2.0260
TDV	SSIM	0.5940 ± 0.0595	0.5459 ± 0.0572	0.5282 ± 0.0584
	PSNR	26.3233 ± 1.9315	24.8127 ± 1.9399	23.3939 ± 1.9662
ACR	SSIM	0.7114 ± 0.0543	0.6575 ± 0.0541	0.5515 ± 0.0529
	PSNR	27.1792 ± 1.9441	25.3342 ± 1.9442	23.4915 ± 1.9539
Plug-and-Play Methods				
DnCNN-PnP	SSIM	0.7617 ± 0.0410	0.6981 ± 0.0441	0.6200 ± 0.0475
	PSNR	26.9997 ± 1.9330	25.0658 ± 1.9248	23.4108 ± 1.9521
DRUNet-PnP	SSIM	0.7634 ± 0.0411	0.7002 ± 0.0443	0.6149 ± 0.0511
	PSNR	27.0262 ± 1.9334	25.0829 ± 1.9254	23.4362 ± 1.9520
GS-PnP	SSIM	0.6396 ± 0.0670	0.5668 ± 0.0663	0.4989 ± 0.0625
	PSNR	26.3318 ± 1.9328	24.5129 ± 1.9282	22.8225 ± 1.9481

Table 6.7: Quantitative analysis of all evaluated methods with respect to PSNR and SSIM on their performance in the CT image reconstruction tasks of Sparse-Angle.

Method	Metric	CT Image Reconstruction Task		
		Sparse-Angle 360	Sparse-Angle 120	Sparse-Angle 60
Classical Methods				
FBP	SSIM	0.2947 ± 0.0453	0.1231 ± 0.0225	0.0611 ± 0.0112
	PSNR	24.9674 ± 2.0415	19.8769 ± 2.0124	16.6451 ± 1.9972
AGD	SSIM	0.3867 ± 0.0563	0.4142 ± 0.0630	0.4333 ± 0.0664
	PSNR	26.9629 ± 2.0444	27.5127 ± 1.9948	27.2796 ± 1.9553
PDHG	SSIM	0.6998 ± 0.0685	0.6712 ± 0.0718	0.5952 ± 0.0728
	PSNR	32.9158 ± 1.9739	31.7980 ± 1.9798	29.8020 ± 1.9326
Post-Processing Methods				
FBP+U-Net	SSIM	0.7449 ± 0.0801	0.7518 ± 0.0657	0.7728 ± 0.0592
	PSNR	30.4766 ± 3.4844	26.0785 ± 6.3779	19.5421 ± 9.9613
FBP+MSDNet	SSIM	0.8392 ± 0.0473	0.7993 ± 0.0650	0.7626 ± 0.0789
	PSNR	33.1188 ± 1.9820	32.2993 ± 1.9928	30.9931 ± 1.9875
FBP+DnCNN	SSIM	0.7864 ± 0.0618	0.6701 ± 0.0864	0.6180 ± 0.0796
	PSNR	31.7575 ± 2.0213	29.1817 ± 2.3053	28.6079 ± 2.3389
Learned / Unrolled Iterative Methods				
LG	SSIM	0.7846 ± 0.0628	0.6795 ± 0.0790	0.6428 ± 0.0777
	PSNR	32.5946 ± 1.9764	31.1950 ± 1.9658	29.9360 ± 1.9603
LGTV	SSIM	0.7811 ± 0.0659	0.7100 ± 0.0745	0.7081 ± 0.0689
	PSNR	32.9404 ± 1.9666	31.4072 ± 1.9647	29.9021 ± 1.9357
LPD	SSIM	**0.8433 ± 0.0479**	**0.8300 ± 0.0500**	**0.8206 ± 0.0508**
	PSNR	**33.3809 ± 1.9513**	**32.7032 ± 1.9685**	**32.0583 ± 1.9789**
Learned Regularizer Methods				
AR	SSIM	0.8309 ± 0.0447	0.8117 ± 0.0553	0.7949 ± 0.0595
	PSNR	32.8067 ± 1.9714	32.1030 ± 1.9577	30.8378 ± 1.9532
TDV	SSIM	0.6815 ± 0.0736	0.6235 ± 0.0741	0.5725 ± 0.0728
	PSNR	32.2673 ± 1.9641	30.6585 ± 1.9357	28.9451 ± 1.8995
ACR	SSIM	0.8271 ± 0.0494	0.8074 ± 0.0539	0.7849 ± 0.0524
	PSNR	33.1537 ± 1.9632	31.9181 ± 1.9666	30.5147±1.9338
Plug-and-Play Methods				
DnCNN-PnP	SSIM	0.8405 ± 0.0432	0.8021 ± 0.0465	0.7637 ± 0.0484
	PSNR	32.4627 ± 1.9309	31.3847 ± 1.9271	29.9350 ± 1.8980
DRUNet-PnP	SSIM	0.8398 ± 0.0433	0.8000 ± 0.0465	0.7658 ± 0.0498
	PSNR	32.5065 ± 1.9324	31.3949 ± 1.9266	29.9518 ± 1.9085
GS-PnP	SSIM	0.7622 ± 0.0628	0.7588 ± 0.0684	0.6937 ± 0.0701
	PSNR	32.1977 ± 1.9353	31.2728 ± 1.9370	29.5791 ± 1.8960

6.5.2 Quantitative and qualitative analysis of the evaluated methods

For this benchmarking we compared a range of algorithms representative for different categories of learned reconstruction methods in several CT image reconstruction tasks and reported their performance with respect to SSIM and PSNR. However, it is particularly important to note that the quantitative analysis presented in Tables 6.5, 6.6, and 6.7 does not capture the nuances of the quality of the reconstruction upon visual inspection. This is particularly true in cases where the task at hand is more challenging, e.g. limited-angle reconstruction from 60°. In this case, the quantitative analysis indicates that both post-processing methods and learned/unrolled iterative methods perform similarly to the learned regularizer and PnP methods (e.g. ACR has better SSIM than LG). However, the qualitative analysis (visual inspection of the reconstructions) shows that for PnP and Learned Regularizer methods, limited angle reconstructions are not performing well, arguably producing images as bad as FBP. Thus, one should not fully trust the performance metrics when comparing models solving such challenging inverse problem scenarios.

Along the same lines, the performance of all evaluated methods on the Full Data reconstruction task should be considered carefully. The "gold standard" or ground truth for this reconstruction task are iterative reference reconstructions computed with an AGD algorithm on a larger field-of-view (2048×2048) and cropped to its center region of (1024×1024). Given that this is a relatively well-posed problem, AGD should have converged to the true minimizer of the data fidelity functional, and thus the resulting image should be a good target for data-driven methods. However, this is not strictly true, as noise in CT acquisition is much more complex than just Gaussian, and this target is not really the exact ground truth. One could argue that a solution from an explicitly regularized algorithm (e.g. Chambolle-Pock with TV) would be also an appropriate target to use, and this would change the numerical results of this work. This choice of using AGD as target, is likely not highly impacting the conclusions of this work, but it is important to clarify that this is a choice, and not a definition of the ground truth.

This explains why AGD performs excellently, and if the same scale of the reconstruction would be kept, the SSIM would be 1 and PSNR infinity. But, as explained, this means that all evaluated methods in the Full Data reconstruction are learning to produce AGD-like results, not the actual ground truth. PnP and learned regularizer methods produce a high SSIM because they are also optimized using Gradient Descent, but with a learned regularization step. Therefore, they can mimic AGD more appropriately.

The beam-hardening corrected reconstruction is a particularly interesting case study as the errors caused by beam-hardening are very non-local and non-linear. This presents a greater challenge than tasks which are more similar to "denoising" such as sparse-angle. Therefore all methods that do not learn to imitate the final results directly (classical methods, PnP and adversarial regularizers) fail to produce a good image. Beam-hardening corrected reconstruction appears to be a more challenging task for variational regularization methods and learned regularizers and PnP methods. This can be explained by the linearization of the forward operator which assumes

monochromatic X-ray sources. However, beam-hardening is ultimately a non-linear effect caused by wide beam spectra and their non-linear absorption. This operator mismatch pushes the minimization into the wrong direction, causing artifacts, as the forward model deviates from the physical process of acquiring the measurements.

As an overall discussion on performance, it is worth noting that post-processing methods, albeit lacking mathematical guarantees, consistently produce quantitatively and visually relatively good results on all CT image reconstruction tasks. As is seen in Figure 6.1, for example for FBP+DnCNN in the Limited Angle reconstruction from 60°, these methods, however, may suffer from "hallucinations". This should not come as a surprise as post-processing methods do not enforce data consistency.

Generally, learned/unrolled iterative methods tend to be better at ensuring consistency in both data and image space, while (learned) regularization approaches provably achieve consistency [176]. However, we emphasize that data consistency is necessary but not sufficient for preventing hallucinations, especially as the ill-posedness of the reconstruction problem increases, and as a result these methods may also suffer from hallucinations. Indeed, even classical, model-based, approaches with theoretical guarantees of data consistency have been seen to exhibit a sensitivity to adversarial perturbations [67]. At the same time, training iterative unrolled methods requires much more time than post-processing methods, although the number of parameters are orders of magnitude smaller than standard networks like the U-Net.

Furthermore, we find that adversarial regularization and PnP approaches both tend to be well-performing in both sparse and full-data settings, oftentimes reaching performance of supervised learned/unrolled iterative methods, by the virtue of relying on the variational formulation and ultimately interpolating the missing image data. However, in the limited angle setting it is no longer an interpolation problem, but instead an in-painting problem. Neither PnP nor AR have been designed for in-painting directly and often rely on local image information. However, in-painting inherently requires non-local information in order to fill-in the missing data.

6.5.3 Limitations and broader impact

In our benchmarking study, we aimed to establish a foundational understanding of how learned algorithms of different method categories perform on standardized CT reconstruction tasks with real-world experimental data. We prioritized adequate performance over extensive hyperparameter tuning to produce baseline results for a comparative analysis. However, we recognize that the interplay of hyperparameters such as architecture, learning rates, regularization strengths, and iteration counts can significantly affect performance.

Consequently, the results shown in Figure 6.2 and Tables 6.5, 6.6, and 6.7 should be interpreted cautiously, as they reflect performance under limited tuning. Future users must conduct thorough hyperparameter optimization tailored to their specific applications to fully leverage each method's potential.

While other CT image reconstruction tasks such as region-of-interest tomography, super-resolution, or segmentation are supported by the 2DeteCT dataset in principle, they have not yet been fully implemented in the benchmarking framework. Furthermore, while the dataset was designed to resemble abdominal CT scans, there are several remaining differences. However, 2DeteCT does provide realistic experimental data for a range of research fields such as manufacturing industry, food industry, and materials science. In an ideal case scenario, it would be possible to have raw measurement data for medical CT scanners and medically relevant subjects. However, medical CT manufacturers claim this data as proprietary and ethical concerns on both patients' privacy and radiation dose prohibit acquiring matching data pairs of e.g. high-dose and low-dose scans or other acquisition modes. Moreover, the mismatch between the 2DeteCT dataset and medical CT image characteristics and morphology could create a significant performance gap that remains unexamined both in this study and in the existing literature. As a result, there is no guarantee that the overall performance trends observed in this work are transferable to medical CT cases. To address this, future research will need to focus on acquiring medical datasets and investigating retraining or transfer learning techniques to confirm or challenge these findings.

Additionally, the dataset has been acquired using a specific acquisition geometry and a non-medical micro-CT scanner, which could limit the generalization of trained algorithms to other CT data. For that, the performance of models trained on 2DeteCT could be evaluated under out-of-distribution (OOD) conditions in both image distribution and forward operator. For instance, applying the trained models to a different dataset would allow for assessing the models' generalization capacity. This type of experiment is particularly critical in the medical field, where out-of-distribution changes are common [95].

Such an evaluation under OOD conditions should be conducted thoroughly, involving a wide range of OOD cases rather than just one or a few specific instances. While the extensive combinations of OOD tests and generalization scenarios pose significant computational challenges, making them impractical for the current study, we release the trained models and accompanying code on GitHub (see Section 6.5.4). We hope this will facilitate future research in this area and believe that our comparison framework lays a solid foundation for conducting comprehensive OOD studies.

There are also slight remaining beam-hardening artifacts in the filtered, clean acquisition of "mode 2", indicating a reduction but not complete removal of beam-hardening. The chosen performance metrics of SSIM and PSNR are common quality assessments, but meaningful quality metrics for reconstructed (medical) CT images should be clinically relevant, task dependent, and aware of unaltered image content [225]. Despite these challenges, trained models on this dataset could potentially be applied to other data through transfer learning, with potential benefits for the medical sector. Researchers must be aware of potential distribution shifts and validate their algorithms on suitable data for the intended application. However, unlike other existing datasets, the 2DeteCT dataset provides raw experimental measurement data and the presented

benchmarks show how well existing algorithms work on real-world experimental data. With this, we take a step towards closing the gap between developed algorithms and real-world applications by utilizing real-world experimental data instead of simulated data.

6.5.4 Code and data availability

The 2DeteCT dataset which serves as the foundation of this benchmarking paper is hosted on zenodo [102] and the LION toolbox is hosted on GitHub by the Cambridge Image Analysis group and maintained by Ander Biguri. This open-source toolbox allows for easy access to a variety of methods, tools, and resources. Additionally, new methods can be seamlessly added to the toolbox, with demos available to showcase how code should be organized and set up within the LION framework. This holds also true for future extensions of the available CT image reconstruction tasks mentioned in the paragraph "Limitations and broader impact". By regularly updating and expanding the toolbox the benchmarking project remains relevant, efficient, and effective in advancing the field of ML-based CT image reconstruction. The models and the scripts to train and evaluate the models presented in this benchmark, will be made available in the GitHub repository above and tagged with 'AMMC_benchmark' upon publication of this work.

6.6 Conclusion

Our benchmarking study provides a comparison of a fixed set of data-driven CT reconstruction algorithms with real-world experimental data in a reproducible and reusable way. It provides a starting point to develop new methods significantly faster as time-consuming implementations of data loaders, reconstruction tasks, comparison methods and evaluation protocols do not have to be redone. The open-source toolbox allows for seamless addition of new state-of-the-art methods and for extensions towards other problems and different CT reconstruction tasks.