

QN-Mixer: A Quasi-Newton MLP-Mixer Model for Sparse-View CT Reconstruction

Anonymous CVPR submission

Paper ID 15740

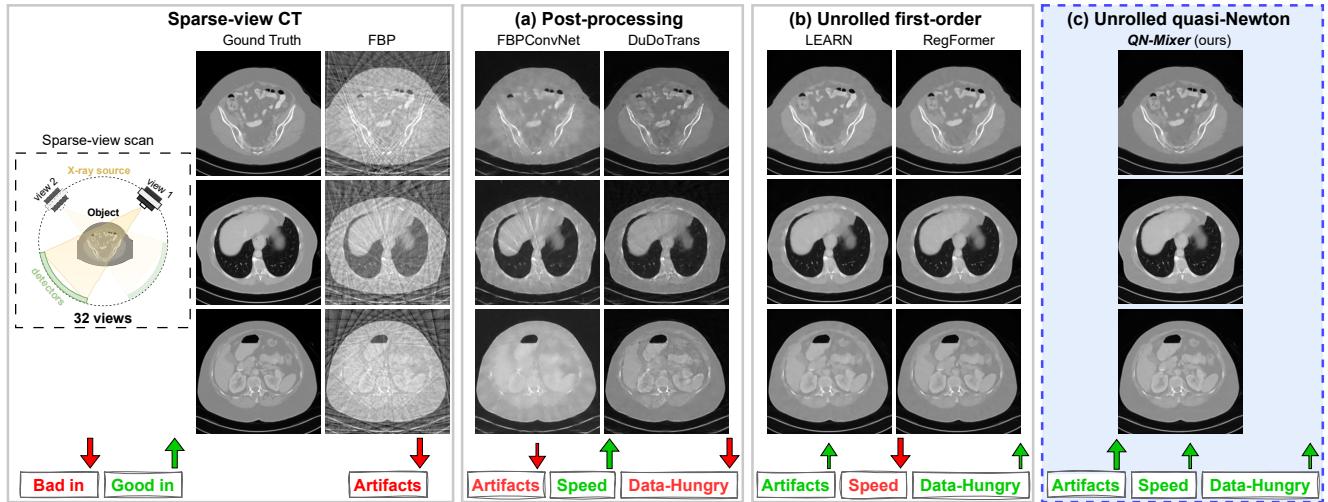


Figure 1. **CT Reconstruction of State-of-the-Art Methods.** We present the results of sparse-view CT reconstruction employing our QN-Mixer approach with 32 views. A comparative analysis against state-of-the-art post-processing and first-order unrolling networks reveals the superiority of QN-Mixer in artifact removal, training time, and data efficiency.

Abstract

001 *Inverse problems span across diverse fields. In medical contexts, computed tomography (CT) plays a crucial 002 role in reconstructing a patient's internal structure, presenting challenges due to artifacts caused by inherently ill- 003 posed inverse problems. Previous research advanced image 004 quality via post-processing and deep unrolling algorithms 005 but faces challenges, such as extended convergence times 006 with ultra-sparse data. Despite enhancements, resulting 007 images often show significant artifacts, limiting their 008 effectiveness for real-world diagnostic applications. We 009 aim to explore deep second-order unrolling algorithms for 010 solving imaging inverse problems, emphasizing their faster 011 convergence and lower time complexity compared to common 012 first-order methods like gradient descent. In this paper, 013 we introduce QN-Mixer, an algorithm based on the quasi- 014 Newton approach. We use learned parameters through the 015 BFGS algorithm and introduce Incept-Mixer, an efficient 016 neural architecture that serves as a non-local regularization 017 term, capturing long-range dependencies within images.*

020 *To address the computational demands typically associated 021 with quasi-Newton algorithms that require full Hessian 022 matrix computations, we present a memory-efficient alterna- 023 tive. Our approach intelligently downsamples gradient 024 information, significantly reducing computational require- 025 ments while maintaining performance. The approach is val- 026 idated through experiments on the sparse-view CT problem, 027 involving various datasets and scanning protocols, and is 028 compared with post-processing and deep unrolling state- 029 of-the-art approaches. Our method outperforms existing 030 approaches and achieves state-of-the-art performance in 031 terms of SSIM and PSNR, all while reducing the number 032 of unrolling iterations required.*

1. Introduction

033 Computed tomography (CT) is a widely used imaging 034 modality in medical diagnosis and treatment planning, de- 035 livering intricate anatomical details of the human body 036 with precision. Despite its success, CT is associated with 037 high radiation doses, which can increase the risk of can- 038

cer induction [46]. Adhering to the ALARA principle (As Low As Reasonably Achievable) [34], the medical community emphasizes minimizing radiation exposure to the lowest level necessary for accurate diagnosis. Numerous approaches have been proposed to reduce radiation doses while maintaining image quality. Among these, sparse-view CT emerges as a promising solution, effectively lowering radiation doses by subsampling the projection data, often referred to as the sinogram. Nonetheless, reconstructed images using the well-known Filtered Back Projection (FBP) algorithm [31], suffer from pronounced streaking artifacts (see Fig. 1), which can lead to misdiagnosis. The challenge of effectively reconstructing high-quality CT images from sparse-view data is gaining increasing attention in both the computer vision and medical imaging communities.

With the success of deep learning spanning diverse domains, initial image-domain techniques [5, 17, 23, 25, 54] have been introduced as post-processing tasks on the FBP reconstructed images, exhibiting notable accomplishments in artifact removal and structure preservation. However, the inherent limitations of these methods arise from their constrained receptive fields, leading to challenges in effectively capturing global information and, consequently, sub-optimal results.

To address this limitation, recent advances have seen a shift toward a dual-domain approach [16, 24, 26, 45], where post-processing methods turn to the sinogram domain. In this dual-domain paradigm, deep neural networks are employed to perform interpolation tasks on the sinogram data [14, 22], facilitating more accurate image reconstruction. Despite the significant achievements of post-processing and dual-domain methods, they confront issues of interpretability and performance limitations, especially when working with small datasets and ultra-sparse-view data, as shown in Fig. 1. To tackle these challenges, deep unrolling networks have been introduced [1, 6, 7, 10, 15, 18, 47, 50]. Unrolling networks treat the sparse-view CT reconstruction problem as an optimization task, resulting in a first-order iterative algorithm like gradient descent, which is subsequently unrolled into a deep recurrent neural network in order to learn the optimization parameters and the regularization term. Like post-processing techniques, unrolling networks have been extended to the sinogram domain [48, 52] to perform interpolation task.

Unrolling networks, as referenced in [11, 33, 41], exhibit remarkable performance across diverse domains. However, they suffer from slow convergence and high computational costs, as illustrated in Fig. 1, necessitating the development of more efficient alternatives [13]. More specifically, they confront two main issues: *Firstly*, they frequently grapple with capturing long-range dependencies due to their dependence on locally-focused regularization terms using CNNs.

This limitation results in suboptimal outcomes, particularly evident in tasks such as image reconstruction. *Secondly*, the escalating computational costs of unrolling methods align with the general trend of increased complexity in modern neural networks. This escalation not only amplifies the required number of iterations due to the algorithm’s iterative nature but also contributes to their high computational demand.

To tackle the aforementioned issues, we introduce a novel second-order unrolling network for sparse-view CT reconstruction. *In particular*, to enable the learnable regularization term to apprehend long-range interactions within the image, we propose a non-local regularization block termed **Incept-Mixer**. Drawing inspiration from the multi-layer perceptron mixer [43] and the inception architecture [42], it is created to combine the best features from both sides: capturing long-range interactions from the attention-like mechanism of MLP-Mixer and extracting local invariant features from the inception block. This block facilitates a more precise image reconstruction. *Second*, to cut down on the computational costs associated with unrolling networks, we propose to decrease the required iterations for convergence by employing second-order optimization methods such as [19, 27]. We introduce a novel unrolling framework named **QN-Mixer**. Our approach is based on the quasi-Newton method that approximate the Hessian matrix using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update [9, 12, 53]. Furthermore, we reduce memory usage by working on a projected gradient (latent gradient), preserving performance while reducing the computational cost tied to Hessian matrix approximation. This adaptation enables the construction of a deep unrolling network, showcasing superlinear convergence. Our contributions are summarized as follows:

- We introduce a novel second-order unrolling network coined **QN-Mixer** where the Hessian matrix is approximated using a latent BFGS algorithm with a deep-net learned regularization term.
- We propose **Incept-Mixer**, a neural architecture acting as a non-local regularization term. Incept-Mixer integrates deep features from inception blocks with MLP-Mixer, enhancing multi-scale information usage and capturing long-range dependencies.
- We demonstrate the effectiveness of our proposed method when applied to the sparse-view CT reconstruction problem on an extensive set of experiments and datasets. We show that our method outperforms state-of-the-art methods in terms of quantitative metrics while requiring less iterations than first-order unrolling networks.

143 2. Related Works

144 In this section, we present prior work closely related to
 145 our paper. We begin by discussing the general framework
 146 for unrolling networks in Sec. 2.1, which is based on the
 147 gradient descent algorithm. Subsequently, in Sec. 2.2 and
 148 Sec. 2.3, we delve into state-of-the-art methods in post-
 149 processing and unrolling networks, respectively.

150 2.1. Background

151 **Inverse Problem Formulation for CT.** Image reconstruction
 152 problem in CT can be mathematically formalized as the
 153 solution to a linear equation in the form of:

$$154 \quad \mathbf{y} = \mathbf{A}\mathbf{x} + \epsilon, \quad (1)$$

155 where $\mathbf{x} \in \mathbb{R}^n$ is the (unknown) object to reconstruct with
 156 $n = h \times w$, $\mathbf{y} \in \mathbb{R}^m$ is the data (i.e. sinogram), where
 157 $m = n_v \times n_d$, n_v and n_d denote the number of projec-
 158 tion views and detectors, respectively. $\mathbf{A} \in \mathbb{R}^{n \times m}$ is the
 159 forward model (i.e. discrete Radon transform [37]), and
 160 $\epsilon \in \mathbb{R}^m$ is the system noise. The goal of CT image re-
 161 construction is to recover the (unknown) object, \mathbf{x} , from
 162 the observed data \mathbf{y} . As the problem is ill-posed due to the
 163 missing data, the linear system in Eq. (1) becomes under-
 164 determined and may have infinite solutions. Hence, recon-
 165 structed images suffer from artifacts, blurring, and noise.
 166 To address this issue, iterative reconstruction algorithms are
 167 utilized to minimize a regularized objective function with a
 168 L^2 norm constraint:

$$169 \quad \hat{\mathbf{x}} = \arg \min_{\mathbf{x}} J(\mathbf{x}) = \frac{\lambda}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \mathcal{R}(\mathbf{x}), \quad (2)$$

170 where $\mathcal{R}(\mathbf{x})$ is the regularization term, balanced with the
 171 weight λ . Those ill-posed problems were initially addressed
 172 using optimization techniques, such as the truncated sin-
 173 gular value decomposition (SVD) algorithm [39], or iter-
 174 ative approaches like the algebraic reconstruction technique
 175 (ART) [4], simultaneous ART (SART) [2], conjugate gra-
 176 dient for least squares (CGLS) [20], and total generalized
 177 variation regularization (TGV) [40]. Additionally, tech-
 178 niques such as total variation [44] and Tikhonov regulariza-
 179 tion [8] can be employed to enhance reconstruction results.
 180 **Deep Unrolling Networks.** By assuming that the regular-
 181 ization term in Eq. (2) (i.e. \mathcal{R}) is differentiable and convex,
 182 a simple gradient descent scheme can be applied to solve
 183 the optimization problem:

$$184 \quad \mathbf{x}_{t+1} = \mathbf{x}_t - \alpha \nabla_{\mathbf{x}} J(\mathbf{x}_t), \\ \text{where } \nabla_{\mathbf{x}} J(\mathbf{x}_t) = \lambda \mathbf{A}^\dagger (\mathbf{A}\mathbf{x}_t - \mathbf{y}) + \nabla_{\mathbf{x}} \mathcal{R}(\mathbf{x}_t). \quad (3)$$

185 Here, α represents the step size (i.e. search step), and \mathbf{A}^\dagger is
 186 the pseudo-inverse of \mathbf{A} .

187 Previous research [15, 49] has emphasized the limita-
 188 tions of optimization algorithms, such as the manual selec-
 189 tion of the regularization term and the optimization hyper-
 190 parameters, which can negatively impact their performance,
 191 limiting their clinical application. Recent advancements
 192 in deep learning techniques have enabled automated pa-
 193 rameter selection directly from the data, as demonstrated
 194 in [6, 10, 21, 30, 35, 52]. By allowing the terms in Eq. (3)
 195 to be dependent on the iteration, the gradient descent itera-
 196 tion becomes:

$$197 \quad \mathbf{x}_{t+1} = \mathbf{x}_t - \lambda_t \mathbf{A}^\dagger (\mathbf{A}\mathbf{x}_t - \mathbf{y}) + \mathcal{G}(\mathbf{x}_t), \quad (4)$$

198 where \mathcal{G} is a learned mapping representing the gradient of
 199 the regularization term. It is worth noting that the step size
 200 α in Eq. (3) is omitted as it is redundant when considering
 201 the learned components of the regularization term. Finally,
 202 Eq. (4) is unrolled into a deep recurrent neural network in
 203 order to learn the optimization parameters.

204 2.2. Post-processing Methods

205 Recent advances in sparse-view CT reconstruction lever-
 206 age two main categories of deep learning methods: post-
 207 processing and dual-domain approaches. Post-processing
 208 methods, including RedCNN [5], FBPCConvNet [17], and
 209 DDNet [54], treat sparse-view reconstruction as a denoising
 210 step using FBP reconstructions as input. While effective in
 211 addressing artifacts and reducing noise, they often struggle
 212 with recovering global information from extremely sparse
 213 data. To overcome this limitation, dual-domain methods in-
 214 tegrate sinograms into neural networks for an interpolation
 215 task, recovering missing data [14, 22]. Dual-domain meth-
 216 ods, surpassing post-processing ones, combine information
 217 from both domains. DuDoNet [26], an initial dual-domain
 218 method, connects image and sinogram domains through
 219 a Radon inversion layer. Recent Transformer-based dual-
 220 domain methods, such as DuDoTrans [45] and DDPTrans-
 221 former [24], aim to capture long-range dependencies in the
 222 sinogram domain, demonstrating superior performance to
 223 CNN-based methods.

224 2.3. Advancements in Deep Unrolling Networks

225 Unrolling networks constitute a line of work inspired by
 226 popular optimization algorithms used to solve Eq. (2).
 227 Leveraging the iterative nature of optimization algorithms,
 228 as presented in Eq. (4), unrolling networks aim to directly
 229 learn optimization parameters from data. These meth-
 230 ods have found success in various inverse problems, in-
 231 cluding sparse-view CT [6, 18, 48, 50, 52], limited-angle
 232 CT [7, 10, 47], low-dose CT [1, 15], and compressed sens-
 233 ing MRI [11, 41].

234 **First-order.** One pioneering unrolling network, Learned
 235 Primal-Dual reconstruction [1], replaces traditional proximal
 236 operators with CNNs. In contrast, LEARN [6] and

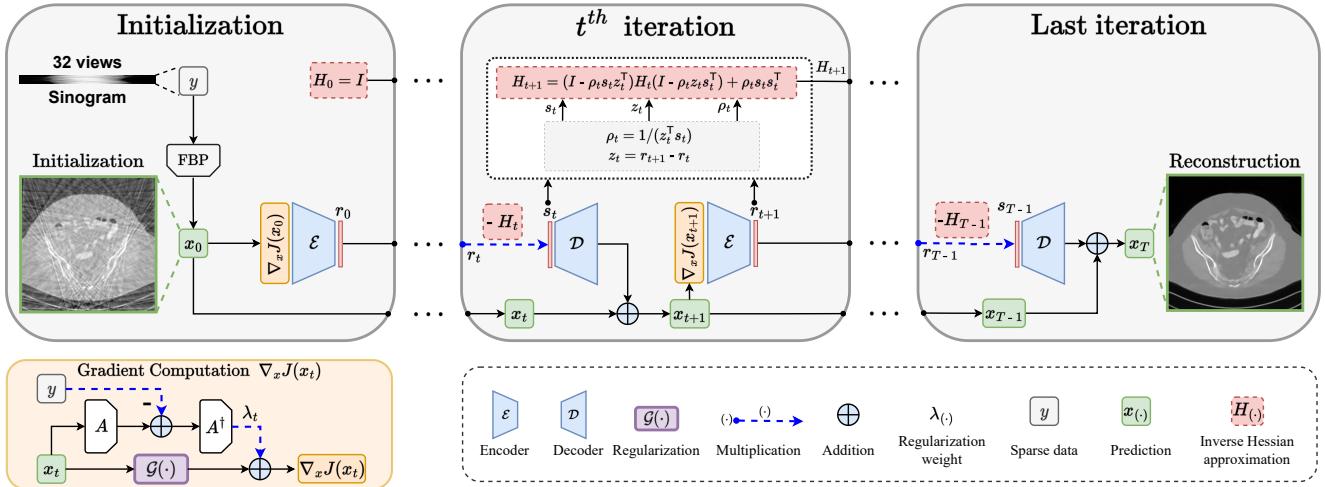


Figure 2. Overall structure of the proposed QN-Mixer for sparse-view CT reconstruction, unrolled from Algorithm 2. The method leverages the advantages of the quasi-Newton method for faster convergence while incorporating a latent BFGS update.

237 LEARN++ [52] directly unroll the optimization algorithm
 238 from Eq. (4) into a deep recurrent neural network. More
 239 recently, Transformers [3, 28] have been introduced into
 240 unrolling networks, such as RegFormer [50] and HUMUS-
 241 Net [11]. While achieving commendable performance,
 242 these methods require more computational resources than
 243 traditional CNN-based unrolling networks and incur a sig-
 244 nificant memory footprint due to linear scaling with the
 245 number of unrolling iterations.

246 **Second-order.** To address this, a new category of unrolling
 247 optimization methods has emerged [13], leveraging second-
 248 order techniques like the quasi-Newton method [9, 12, 19].
 249 These methods converge faster, reducing computational de-
 250 demands, but struggle with increased memory usage due to
 251 Hessian matrix approximation and their application is lim-
 252 ited to small-scale problems [27, 53]. In contrast our
 253 method propose a memory-efficient approach by oper-
 254 ating within the latent space of gradient information (i.e.
 255 $\nabla_x J(x)$ in Eq. (3)).

Algorithm 1: Quasi-Newton for sparse-view CT

Data: y (sparse sinogram)
 Manual choice of the regularization term \mathcal{R} ;
 $H_0 \leftarrow I^{n \times n}$;
 $x_0 \leftarrow A^\dagger y$;
for $t \in \{0, \dots, T-1\}$ **do**

$$\begin{aligned} s_t &\leftarrow -H_t \nabla_x J(x_t) \\ x_{t+1} &\leftarrow x_t + s_t \\ z_t &\leftarrow \nabla_x J(x_{t+1}) - \nabla_x J(x_t) \\ \rho_t &\leftarrow 1/(z_t^\top s_t) \\ H_{t+1} &\leftarrow (I - \rho_t s_t z_t^\top) H_t (I - \rho_t z_t s_t^\top) + \rho_t s_t s_t^\top \end{aligned}$$

3. Methodology

QN-Mixer is a novel second-order unrolling network inspired by the quasi-Newton (Sec. 3.1) method. It approximates the inverse Hessian matrix with a latent BFGS algorithm and includes a non-local regularization term, Incept-Mixer, designed to capture non-local relationships (Sec. 3.2). To cope with the significant computational burden associated with the full approximation of the inverse Hessian matrix, we use a latent BFGS algorithm (Sec. 3.3). An overview of the proposed method is depicted in Fig. 2, and the complete algorithm is presented in Sec. 3.4.

3.1. Quasi-Newton method

The quasi-Newton method can be applied to solve Eq. (2) and the iterative optimization solution is expressed as:

$$x_{t+1} = x_t - \alpha_t H_t \nabla_x J(x_t), \quad (5)$$

where $H_t \in \mathbb{R}^{n \times n}$ represents the inverse Hessian matrix approximation at iteration t , and α_t is the step size. The BFGS method updates the Hessian matrix approximation in each iteration. This matrix is crucial for understanding the curvature of the objective function around the current point, guiding us to take more efficient steps and avoiding unnecessary zigzagging. In the classical BFGS approach, the line search adheres to Wolfe conditions [9, 12]. A step size of $\alpha_t = 1$ is attempted first, ensuring eventual acceptance for superlinear convergence [19]. In our approach, we adopt a fixed step size of $\alpha_t = 1$. The algorithm is illustrated in Algorithm 1.

3.2. Regularization term: Incept-Mixer

Recent research on unrolling networks has often focused on selecting the representation of the regularization term grad-

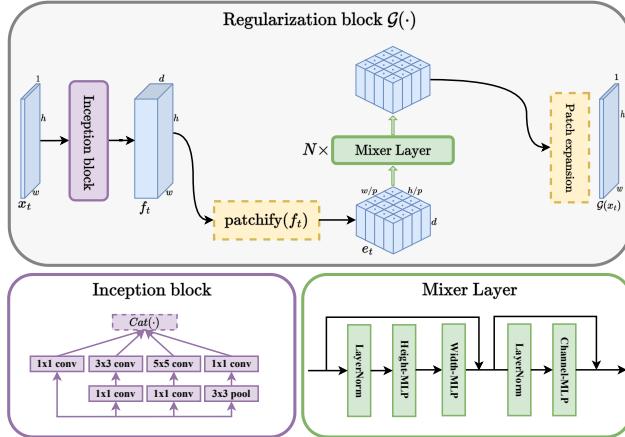


Figure 3. Architecture of our regularization block. It is referred to as “**Incept-Mixer**” and denoted as \mathcal{G} in Eq. (4)

ent (i.e. \mathcal{G} in Eq. (4)), ranging from conv-nets [6, 41, 52] to more recent attention-based nets [11, 50]. In alignment with this trend, we introduce a non-local regularization block named **Incept-Mixer** and depicted in, Fig. 3. This block is crafted by drawing inspiration from both the multi-layer perceptron mixer [43] and the inception architecture [42], leveraging the strengths of each: capturing long-range interactions through the attention-like mechanism of MLP-Mixer and extracting local invariant features from the inception block. This design choice is evident in the ablation study (see Tab. 6) where Incept-Mixer outperforms both alternatives.

Starting from an image $x_t \in \mathbb{R}^{h \times w \times c}$ at iteration t , we pass it through an Inception block to create a feature map $f_t \in \mathbb{R}^{h \times w \times d}$, where d is the depth of features. Subsequently, f_t undergoes patchification using a CNN with a kernel size and stride of p , representing the patch size. This process yields patch embeddings, $e_t = \text{patchify}(f_t) \in \mathbb{R}^{\frac{h}{p} \times \frac{w}{p} \times d}$. These embeddings are then processed through a **Mixer Layer** with token and channel MLPs, layer normalization, and skip connections for inter-layer information flow, following [43]:

$$\begin{aligned} \text{MixerLayer}(e_t) &= \text{Mix}(\text{MLP}_{\text{channel}}, \\ &\quad \text{Mix}([\text{MLP}_{\text{height}}, \text{MLP}_{\text{width}}], e_t), \end{aligned} \quad (6)$$

where $\text{Mix}(\text{Layer}, e_t) = \text{Layer}(\text{LN}(e_t)) + e_t$, with LN as layer normalization. $\text{MLP}_{\text{height}}$, $\text{MLP}_{\text{width}}$ are applied to height and width features, respectively, and $\text{MLP}_{\text{channel}}$ to rows and shared. Finally, after N such mixer layers, the regularized sample is transformed back to an image through a patch expansion step to obtain $G(x_t)$. Consequently, the iterative optimization solution is as follows:

$$\begin{aligned} x_{t+1} &= x_t - H_t \nabla_x J(x_t), \\ \text{where } \nabla_x J(x_t) &= \lambda_t A^\dagger (Ax_t - y) + \mathcal{G}(x_t). \end{aligned} \quad (7)$$

Here, $\mathcal{G}(x_t)$ denotes the Incept-Mixer model, representing the learned gradient of the regularization term.

3.3. Latent BFGS update

We propose a memory-efficient latent BFGS update. Drawing inspiration from LDMs [38], At step t , given the gradient value $\nabla_x J(x_t) \in \mathbb{R}^{h \times w \times c}$, the encoder \mathcal{E} encodes it into a latent representation $r_t = \mathcal{E}(\nabla_x J(x_t)) \in \mathbb{R}^{l_h \cdot l_w}$. Importantly, the encoder downsamples the gradient by a factor $f_\mathcal{E} = \frac{h}{h_l} = \frac{w}{w_l}$. Throughout the paper, we explore different downsampling factors (see Tab. 5) $f_\mathcal{E} = 2^k$, where $k \in \mathbb{N}$ is the number of downsampling stacks. Encoding the gradient reduces the optimization variable size of BFGS (i.e. $H_t \in \mathbb{R}^{(l_h \cdot l_w) \times (l_h \cdot l_w)}$), thereby decreasing the computational cost associated with high memory demand. The direction is then computed in the latent space $s_t = -H_t r_t$, and finally, the decoder \mathcal{D} reconstructs the update from the latent direction, giving $\mathcal{D}(s_t) = \mathcal{D}(-H_t \mathcal{E}(\nabla_x J(x_t))) \in \mathbb{R}^{h \times w \times c}$. It is noteworthy that \mathcal{E} and \mathcal{D} are shared across the algorithm iterations, as shown in Fig. 2.

3.4. Proposed algorithm of QN-Mixer

Algorithm 2: QN-Mixer (latent BFGS update)

Data: y (sparse sinogram)
 $H_0 \leftarrow I^{(l_h \cdot l_w) \times (l_h \cdot l_w)}$;
 $x_0 \leftarrow A^\dagger y$;
 $r_0 \leftarrow \mathcal{E}(\nabla_x J(x_0))$;
for $t \in \{0, \dots, T-1\}$ **do**

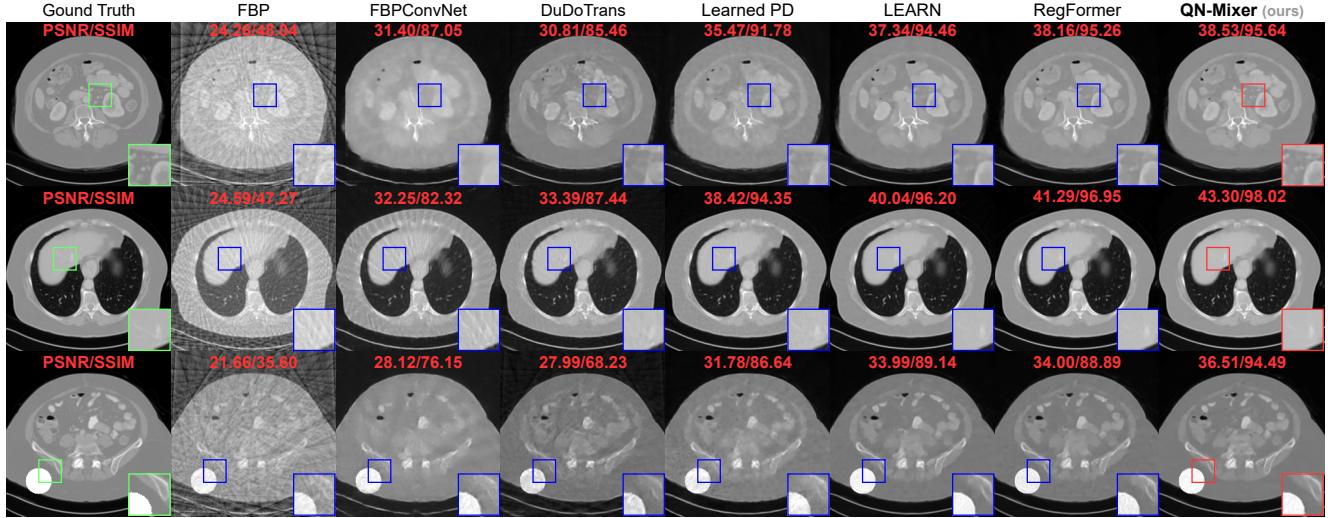
$$\begin{cases} s_t \leftarrow -H_t r_t \\ x_{t+1} \leftarrow x_t + \mathcal{D}(s_t) \\ r_{t+1} \leftarrow \mathcal{E}(\nabla_x J(x_{t+1})) \\ z_t \leftarrow r_{t+1} - r_t \\ \rho_t \leftarrow 1/(z_t^\top s_t) \\ H_{t+1} \leftarrow (I - \rho_t s_t z_t^\top) H_t (I - \rho_t z_t s_t^\top) + \rho_t s_t s_t^\top \end{cases}$$

Our method, builds on the BFGS update [9, 12] rank-one approximation for the inverse Hessian. This approximation serves as a preconditioning matrix, guiding the descent direction. In contrast to [13], which directly learns the inverse Hessian approximation from data, our approach incorporates the mathematical equations of the BFGS algorithm for more accurate approximations. The full QN-Mixer algorithm is illustrated in Algorithm 2.

4. Experiments

In this section, we initially present our experimental settings, followed by a comparison of our approach with other state-of-the-art CT reconstruction methods. Finally, we delve into the contribution analysis of each component in our model.

Method	No noise ($N_0 = 0$)						Low noise ($N_1 = 10^6$)						High noise ($N_2 = 5 \times 10^5$)					
	$n_v = 32$		$n_v = 64$		$n_v = 128$		$n_v = 32$		$n_v = 64$		$n_v = 128$		$n_v = 32$		$n_v = 64$		$n_v = 128$	
	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑	PSNR ↑ SSIM ↑
FBP	22.65	40.49	27.29	57.94	33.04	79.50	22.09	32.73	26.51	49.56	31.69	71.09	19.05	15.56	22.71	25.74	26.52	40.87
FBPConvNet [17]	30.32	85.11	35.42	90.15	39.71	94.64	30.20	84.46	35.09	89.72	39.06	94.08	29.91	82.52	34.13	87.85	36.89	91.28
DuDoTrans [45]	30.48	84.70	35.37	91.87	40.62	96.41	30.34	83.72	35.36	91.42	39.75	95.49	30.09	81.83	34.09	88.67	37.08	93.44
Learned PD [1]	35.88	92.09	41.03	96.28	43.33	97.31	35.78	92.21	39.03	94.79	41.65	96.44	33.80	89.23	37.34	93.23	39.17	94.69
LEARN [6]	37.58	94.65	42.26	97.25	43.11	97.57	36.95	93.63	39.91	95.82	42.17	97.11	34.38	90.51	37.15	93.53	39.38	95.18
RegFormer [50]	38.71	95.42	43.56	97.76	47.95	98.98	37.21	94.73	41.65	96.92	44.38	98.02	35.93	92.78	38.53	94.84	40.52	96.19
QN-Mixer (ours)	39.51	96.11	45.57	98.48	50.09	99.32	37.50	94.92	42.46	97.70	44.27	98.11	35.91	92.49	38.73	94.92	40.51	96.27

Table 1. Quantitative evaluation on AAPM of state-of-the-art methods (PSNR in dB and SSIM in %). **Bold**: Best, under: second best.Figure 4. **Visual comparison on AAPM.** From top to bottom: the results under the following conditions: first ($n_v = 32, N_1$), second ($n_v = 64, N_1$), third ($n_v = 32, N_0$). The last row presents out-of-distribution (OOD) results with a randomly overlaid circle on a test image. The display window is set to $[-1000, 800]$ HU.

4.1. Experimental Setup

Datasets. We evaluate our method on two widely used datasets: the “2016 NIH-AAPM-Mayo Clinic Low-Dose CT Grand Challenge” dataset (AAPM) [32] and the DeepLesion dataset [51]. The AAPM dataset comprises 2378 full-dose CT images from 10 patients, while DeepLesion is the largest publicly accessible multi-lesion real-world CT dataset, including 4427 unique patients.

Implementation details. For AAPM, we select 1920 training images from 8 patients, 244 validation images from 1 patient, and 214 testing images from the last patient. For DeepLesion, we select a subset of 2000 training images and 300 testing images randomly from the official splits. All images are resized to 256×256 pixels. To simulate the forward and backprojection operators, we use the Operator Discretization Library (ODL) [36] with a 2D fan-beam geometry (512 detector pixels, source-to-axis distance of 600 mm, axis-to-detector distance of 290 mm). Sparse-view CT images are generated with $n_v \in \{32, 64, 128\}$ projection views, uniformly sampled from a full set of 512 views covering $[0, 2\pi]$. To mimic real-world CT images, we intro-

duce mixed noise to the sinograms, combining 5% Gaussian noise and Poisson noise with an intensity of 1×10^6 .

Training details. For each set of n_v views, we train our model for 50 epochs using 4 Nvidia Tesla V100 (32GB RAM). We employ the AdamW optimizer [29] with a learning rate of 1×10^{-4} , weight decay 1×10^{-2} , and utilize the mean squared error loss with a batch size of 1. Additionally, we incorporate a learning rate decay factor of 0.1 after 40 epochs. Unrolling iterations for QN-Mixer are set to $T = 14$. Incept-Mixer uses a patch size of $p = 4$, $d = 96$ embedding dimension, and $N = 2$ mixer layers. The inverse Hessian size is $64^2 \times 64^2$ with $k = 2$ downsampling blocks. Following [50], A^\dagger is implemented using the FBP algorithm for the pseudo-inverse of A .

Evaluation metrics. Consistent with standard evaluation practices [1, 45, 50], we use the structural similarity index measure (SSIM) with parameters set to: level = 5, a Gaussian kernel of size 11, and a standard deviation of 1.5 as our primary performance indicator. Additionally, we complement our evaluation with the peak signal-to-noise ratio (PSNR).

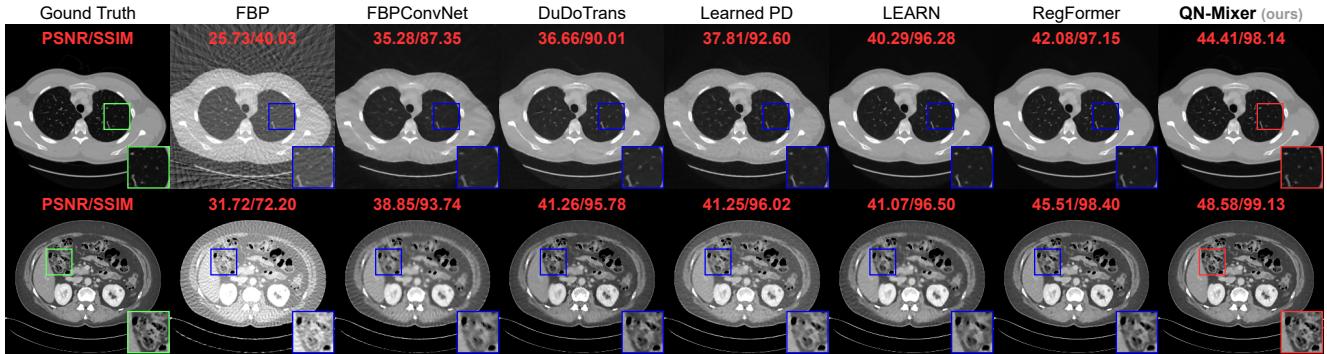


Figure 5. **Visual comparison on DeepLesion** of state-of-the-art methods. Rows display results under different conditions: ($n_v = 64, N_1$) and ($n_v = 128, N_1$). Display windows are set to $[-1000, 800]$ HU for the first row and $[-200, 300]$ HU for the second row.

Method	$n_v = 32$		$n_v = 64$		$n_v = 128$	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
FBP	21.55	31.65	26.07	47.17	31.49	69.63
FBPConvNet [17]	30.74	80.41	34.64	87.36	38.69	92.94
DuDoTrans [45]	32.11	79.86	36.02	88.14	40.47	93.81
Learned PD [1]	34.02	88.44	37.56	92.46	40.79	95.32
LEARN [6]	35.76	92.12	39.83	95.66	41.34	96.21
RegFormer [50]	37.38	93.89	41.70	96.78	46.10	98.39
QN-Mixer (ours)	39.39	95.67	43.75	97.73	48.62	98.64

Table 2. **Quantitative evaluation on DeepLesion** for state-of-the-art methods (PSNR in dB and SSIM in %). With Poisson noise level of $N_1 = 10^6$. **Bold**: Best, under: second best.

State-of-the-art baselines. We compare QN-Mixer to multiple state-of-the-art competitors: (1) *post-processing* based denoising methods, i.e., FBPConvNet [17], and DuDoTrans [45]; (2) *first-order unrolling* reconstruction networks, i.e., Learned Primal-Dual [1], LEARN [6], and RegFormer [50]. Note that we replace the pseudo-inverse operator used by LEARN with the FBP algorithm, as it has been demonstrated to be more effective according to [50]. To ensure a fair comparison, we utilize the code-base released by the authors when possible or meticulously implement the methods based on the details provided in their papers. All approaches undergo training and testing on the same datasets, as elaborated in implementation details.

4.2. Comparison with state-of-the-art methods

Quantitative comparison. We compared our model with state-of-the-art baselines on two public datasets. For AAPM, models were trained and tested across three projection views ($n_v \in \{32, 64, 128\}$) and three noise levels, namely no noise $N_0 = 0$, low noise $N_1 = 10^6$, and high noise $N_2 = 5 \times 10^5$ (see Tab. 1). For DeepLesion, models were trained and tested on the same three projection views and a noise level of $N_1 = 10^6$ (see Tab. 2). Visual results are provided in Fig. 4 (AAPM) and Fig. 5 (DeepLesion). Impressively, our method achieves state-of-the-art results on DeepLesion across all projection views. It outperforms the second-best baseline, RegFormer, with an average im-

Method	$n_v = 32$		$n_v = 64$		$n_v = 128$	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
FBP	21.38	33.36	26.08	50.29	31.43	73.06
FBPConvNet [17]	28.05	<u>75.96</u>	32.50	82.90	35.45	88.14
DuDoTrans [45]	28.11	68.17	32.71	83.26	36.41	90.36
Learned PD [1]	31.96	87.10	36.40	92.57	37.63	93.17
LEARN [6]	34.48	<u>90.15</u>	36.89	<u>91.85</u>	<u>38.32</u>	<u>94.67</u>
RegFormer [50]	<u>34.49</u>	89.98	<u>36.95</u>	91.48	38.02	92.44
QN-Mixer (ours)	36.84	94.84	42.11	97.78	45.69	98.82

Table 3. **Quantitative evaluation on out-of-distribution (OOD) AAPM test dataset** of state-of-the-art methods (PSNR in dB and SSIM in %). **Bold**: Best, under: second best.

provements of +2.23 dB in PSNR and +1.02% in SSIM. On AAPM without noise, we achieve state-of-the-art results across all projection views and improve the second best by an average +1.65 dB and +0.58%. In the presence of low noise, QN-Mixer achieves state-of-the-art results performance in all cases except $n_v = 128$ with -0.11 dB and shows an average improvements of +0.33 dB and +0.35% over RegFormer. With high noise, our method performs nearly on par in $n_v = 32$ (-0.02 dB and -0.29%), achieves state-of-the-art in $n_v = 64$ (+0.2 dB and +0.08%), and competes closely in $n_v = 128$ (-0.01 dB and +0.08%). As noise increases, we attribute the decline in improvement to the compressed gradient information in the latent BFGS, influenced by sinogram changes, and the utilization of the FBP algorithm instead of the pseudo-inverse.

Performance comparison on OOD textures. In medical imaging, where training data predominantly consists of normal patient images, it is imperative to develop methods that generalize to scans with lesions or anomalies. To address this challenge, we conducted an experiment to evaluate model performance on out-of-distribution (OOD) textures. The frozen models were tested with CT images featuring a randomly positioned white circle with no noise added to the sinograms, as illustrated in the third row of Fig. 4. In Tab. 3, QN-Mixer attains state-of-the-art results across all n_v views. First-order unrolling networks such as LEARN and RegFormer exhibit significant PSNR degrada-

419
420
421
422
423
424
425
426
427
428
429
430
431
432
433

446
447
448
449
450
451
452
453
454
455
456
457
458

tion of -3.1 dB and -4.22 dB, respectively, for $n_v = 32$, while our method demonstrates a milder degradation of -2.67 dB.

Visual comparison. As it can be seen on Fig. 4 and Fig. 5, FBPCovNet and DuDoTrans exhibit noticeable blurry images with severe artifacts when $n_v = 32$. While Learned PD and LEARN show satisfactory performance, they struggle with intricate details, like in the liver and spine. In contrast, RegFormer produces high-quality images but faces challenges in generalizing to OOD data. QN-Mixer excels in producing high-quality images with fine details, even under challenging conditions such as $n_v = 32$ views and OOD data.

Method	#Iters	Epoch time (s)	#Params (M)	Memory (GB)
FBPCovNet [17]	-	68	31.1	1.30
DuDoTrans [45]	-	92	15.0	1.38
Learned PD [1]	10	82	0.25	0.81
LEARN [6]	30	780	4.50	1.85
RegFormer [50]	18	700	5.00	10.19
QN-Mixer (ours)	14	594	8.50	7.83

Table 4. **Comparison of computational efficiency.** Training epoch time is reported in seconds, #Params in M and memory costs for state-of-the-art methods on AAPM with $n_v = 32$ views.

459
460
461
462
463
464
465
466
467

Efficiency comparison. The results in Tab. 4 show that QN-Mixer is more computationally efficient than RegFormer, with a $1.3\times$ reduction in memory usage. Furthermore, our training time demonstrates a significant enhancement, realizing a speed improvement of 106 seconds per epoch compared to first-order unrolling methods like LEARN and RegFormer. Additionally, our method requires only 14 iterations, in contrast to the 30 and 18 iterations needed by LEARN and RegFormer, respectively.

Hessian size	PSNR \uparrow	SSIM \uparrow
$8^2 \times 8^2$	35.69	93.71
$16^2 \times 16^2$	38.11	95.31
$32^2 \times 32^2$	39.37	96.01
$64^2 \times 64^2$	39.51	96.11

Table 5. **Ablation on the inverse Hessian approximation size.**

4.3. Ablation Study

468
469
470
471
472
473
474
475
476
477
478

In this section, we leverage the AAPM dataset with $n_v = 32$ views by default, and no noise is introduced to the sinogram. **Inverse Hessian approximation size.** The results in Tab. 5 emphasize the significant impact of the inverse Hessian approximation size on our performance. When too small, a notable degradation is observed (e.g., $8^2 \times 8^2$), while larger sizes result in performance improvements as the approximation approaches the full inverse Hessian. However, exceeding $64^2 \times 64^2$ was unfeasible in our experiments due to increasing memory costs and memory constraints.

Method	PSNR \uparrow	SSIM \uparrow
QN+Inception	31.65	85.28
QN+MLP-Mixer	36.89	93.87
QN-Mixer+ A^\dagger	38.94	95.85
Incept-Mixer + first-order	37.45	94.25
QN-Mixer (ours)	39.51	96.11

Table 6. **Ablation for the regularization term.** A^\dagger denotes the pseudo-inverse.

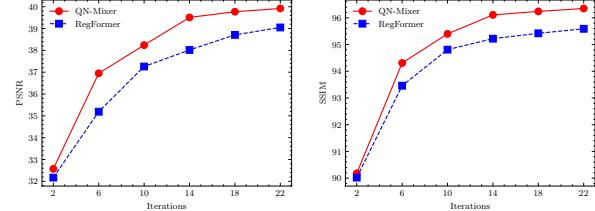


Figure 6. **Ablation on the number of unrolling iterations.** We compare our QN-Mixer against RegFormer. Left: PSNR (dB); Right: SSIM (%)

Number of unrolling iterations. In Fig. 6, we visually depict the influence of the number of unrolling iterations on the performance of QN-Mixer and RegFormer. Notably, the performance of both methods shows improvement with an increase in the number of iterations. When subjected to an equal number of iterations, our method consistently surpasses RegFormer in performance. Remarkably, we achieve comparable results to RegFormer even with only 10 iterations, demonstrating the efficiency of our approach.

Regularization term. We evaluate the impact of the regularization term in our framework. Our Incept-Mixer is compared against various learned alternatives, including the Inception block [42] and MLP-Mixer block [43]. Additionally, we explore the use of the pseudo-inverse A^\dagger instead of the FBP. The results presented in Tab. 6 show that employing the pseudo-inverse results in a less pronounced degradation (-0.57 dB and -0.28%), enhancing the interpretability of QN-Mixer. Finally, we test our Incept-Mixer in the first-order framework, highlighting the significance of the second-order latent BFGS approximation with a significant improvement ($+2.06$ dB and $+1.86\%$).

5. Conclusion

In this paper, we investigate the application of deep second-order unrolling networks for tackling imaging inverse problems. To this end, we introduce QN-Mixer, a quasi-Newton inspired algorithm where a latent BFGS method approximates the inverse Hessian, and our Incept-Mixer serves as the non-local learnable regularization term. Extensive experiments confirm the successful sparse-view CT reconstruction by our model, showcasing superior performance with fewer iterations than state-of-the-art methods. In summary, this research offers a fresh perspective that can be applied to any iterative reconstruction algorithm. A limitation of our work is the memory requirements associated with quasi-Newton algorithm. We introduced a memory-efficient alternative by projecting the gradient to a lower dimension, successfully addressing the CT reconstruction problem. However, its applicability to other inverse problems may be limited. In future work, we aim to extend our approach to handle larger Hessian sizes, broadening its application to a range of problems.

520 **References**

- [1] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE TMI*, 37:1322–1332, 2018. 2, 3, 6, 7, 8, 521 4
- [2] Andersen Arie and Kak Avinash. Simultaneous algebraic 522 reconstruction technique (SART): a superior implementation 523 of the art algorithm. *Ultrasonic imaging*, 6:81–94, 1984. 3
- [3] Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit 524 Jakob, Jones Llion, Gomez Aidan N, Kaiser Lukasz, and 525 Polosukhin Illia. Attention is all you need. In *NeurIPS*, 2017. 526 4
- [4] Kak Avinash and Slaney Malcolm. *Principles of Computerized 527 Tomographic Imaging*. Society for Industrial and Applied Mathematics, 2001. 3
- [5] Hu Chen, Yi Zhang, Mannudeep K. Kalra, Feng Lin, Yang 528 Chen, Peixi Liao, Jiliu Zhou, and Ge Wang. Low-dose CT 529 with a residual encoder-decoder convolutional neural network. 530 *IEEE TMI*, 36:2524–2535, 2017. 2, 3
- [6] Hu Chen, Yi Zhang, Yunjin Chen, Junfeng Zhang, Weihua 531 Zhang, Huaiqiang Sun, Yang Lv, Peixi Liao, Jiliu Zhou, and 532 Ge Wang. LEARN: Learned experts’ assessment-based 533 reconstruction network for sparse-data CT. *IEEE TMI*, 37: 534 1333–1347, 2018. 2, 3, 5, 6, 7, 8, 4
- [7] Weilin Cheng, Yu Wang, Hongwei Li, and Yuping Duan. 535 Learned full-sampling reconstruction from incomplete data. 536 *IEEE TCI*, pages 945–957, 2020. 2, 3
- [8] Peng Chengbin, Rodi William L., and M. Nafi Toksoz. A 537 *Tikhonov Regularization Method for Image Reconstruction*, 538 pages 153–164. Springer US, 1993. 3
- [9] William C. Davidon. Variable metric method for minimization. 539 *SIAM Journal on Optimization*, 1:1–17, 1991. 2, 4, 5, 540 1
- [10] Hu Dianlin, Zhang Yikun, Liu Jin, Luo Shouhua, and Chen 541 Yang. DIOR: Deep iterative optimization-based residual- 542 learning for limited-angle CT reconstruction. *IEEE TMI*, 543 pages 1778–1790, 2022. 2, 3
- [11] Zalan Fabian, Berk Tinaz, and Mahdi Soltanolkotabi. 544 HUMUS-Net: Hybrid unrolled multi-scale network architecture 545 for accelerated MRI reconstruction. In *NeurIPS*, 2022. 546 2, 3, 4, 5
- [12] Roger Fletcher. *Practical Methods of Optimization*. John 547 Wiley & Sons, New York, NY, USA, 1987. 2, 4, 5, 1
- [13] Erik Gartner, Luke Metz, Mykhaylo Andriluka, C. Daniel 548 Freeman, and Cristian Sminchisescu. Transformer-based 549 Learned Optimization. In *CVPR*, pages 11970–11979, 2023. 550 2, 4, 5
- [14] Muhammad Usman Ghani and W. Clem Karl. Deep 551 learning-based sinogram completion for low-dose CT. In 552 *2018 IEEE 13th Image, Video, and Multidimensional Signal 553 Processing Workshop*, pages 1–5, 2018. 2, 3
- [15] Gupta Harshit, Jin Kyong Hwan, Nguyen Ha Q., McCann Michael T., and Unser Michael. CNN-based projected 554 gradient descent for consistent CT image reconstruction. 555 *IEEE TMI*, 37:1440–1453, 2018. 2, 3
- [16] Dianlin Hu, Jin Liu, Tianling Lv, Qianlong Zhao, Yikun 556 Zhang, Guotao Quan, Juan Feng, Yang Chen, and Limin 557
- Luo. Hybrid-domain neural network processing for sparse- 558 view CT reconstruction. *IEEE TRPMS*, 5:88–98, 2020. 2
- [17] Kyong Hwan Jin, Michael T. McCann, Emmanuel Froustey, 559 and Michael Unser. Deep convolutional neural network for 560 inverse problems in imaging. *IEEE TIP*, 26:4509–4522, 561 2017. 2, 3, 6, 7, 8, 4
- [18] Xiang Jinxi, Dong Yonggui, and Yang Yunjie. FISTA-Net: 562 Learning a fast iterative shrinkage thresholding network for 563 inverse problems in imaging. *IEEE TMI*, 40:1329–1339, 564 2021. 2, 3
- [19] Nocedal Jorge and Wright Stephen J. Quasi-Newton meth- 565 ods. *Numerical optimization*, 75:135–163, 2006. 2, 4, 1
- [20] Satoshi Kawata and Orhan Nalcioglu. Constrained iterative 566 reconstruction by the conjugate gradient method. *IEEE TMI*, 567 4:65–71, 1985. 3
- [21] Erich Kobler, Alexander Effland, Karl Kunisch, and Thomas 568 Pock. Total deep variation for linear inverse problems. In 569 *CVPR*, pages 7546–7555, 2020. 3
- [22] Hoyeon Lee, Jongha Lee, Hyeongseok Kim, Byungchul 570 Cho, and Seungryong Cho. Deep-neural-network-based 571 sinogram synthesis for sparse-view CT image reconstruc- 572 tion. *IEEE TRPMS*, 3:109–119, 2018. 2, 3
- [23] Minjae Lee, Hyemi Kim, and Hee-Joung Kim. Sparse-view 573 CT reconstruction based on multi-level wavelet convolution 574 neural network. *Physica Medica*, 80:352–362, 2020. 2
- [24] Runrui Li, Qing Li, Hexi Wang, Saize Li, Juanjuan Zhao, 575 Yan Qiang, and Long Wang. DDPTransformer: Dual- 576 domain with parallel transformer network for sparse view CT 577 image reconstruction. *IEEE TCI*, pages 1–15, 2022. 2, 3
- [25] Zilong Li, Chenglong Ma, Jie Chen, Junping Zhang, and 578 Hongming Shan. Learning to distill global representation 579 for sparse-view ct. In *ICCV*, pages 21196–21207, 2023. 2
- [26] Wei-An Lin, Cheng Liao, Haofu Peng, Xiaohang Sun, Jing- 580 dan Zhang, Jiebo Luo, Rama Chellappa, and Zhou S. Kevin. 581 DuDoNet: Dual domain network for CT metal artifact reduc- 582 tion. In *CVPR*, pages 10512–10521, 2019. 2, 3
- [27] Chengchang Liu and Luo Luo. Quasi-newton methods for 583 saddle point problems. In *NeurIPS*, 2022. 2, 4
- [28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng 584 Zhang, and Stephen Lin Baining Guo. Swin transformer: 585 Hierarchical vision transformer using shifted windows. In 586 *ICCV*, pages 9992–10002, 2021. 4
- [29] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay 587 Regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6
- [30] Sebastian Lunz, Ozan Öktem, and Carola-Bibiane 588 Schönlieb. Adversarial regularizers in inverse problems. In 589 *NeurIPS*, 2018. 3
- [31] Cormack Allan Macleod. Representation of a function by its 590 line integrals, with some radiological applications. *Journal 591 of Applied Physics*, 34:2722–2727, 1963. 2
- [32] C. McCollough. TU-FG-207A-04: Overview of the low dose 592 CT grand challenge. *Medical Physics*, 43:3759–3760, 2016. 593 6, 7
- [33] Luke Metz, C. Daniel Freeman, James Harrison, Niru Mah- 594 eswaranathan, and Jascha Sohl-Dickstein. Practical tradeoffs 595 between memory, compute, and performance in learned op- 596 timizers. *arXiv preprint arXiv:2203.11860*, 2022. 2

- 633 [34] Donald L. Miller and David Schauer. The alara principle in
634 medical imaging. *Philosophy*, 44:595–600, 1983. 2
- 635 [35] Subhadip Mukherjee, Marcello Carioni, Ozan Öktem, and
636 Carola-Bibiane Schönlieb. End-to-end reconstruction meets
637 data-driven regularization for inverse problems. In *NeurIPS*,
638 2021. 3
- 639 [36] Ozan Öktem, Jonas Adler, Holger Kohr, and The ODL Team.
640 Operator discretization library (ODL), 2014. 6
- 641 [37] Johann Radon. über die bestimmung von funktionen
642 durch ihre integralwerte längs gewisser mannigfaltigkeiten.
643 *Berichte über die Verhandlungen der Königlich-Sächsischen*
644 *Akademie der Wissenschaften zu Leipzig*, 69:262–277, 1917.
645 3
- 646 [38] Rombach Robin, Blattmann Andreas, Lorenz Dominik,
647 Esser Patrick, and Ommer Björn. High-resolution image
648 synthesis with latent diffusion models. In *CVPR*, pages
649 10684–10695, 2022. 5
- 650 [39] Liu Rui, He Lu, Luo Yan, and Yu Hengyong. Singular value
651 decomposition-based 2D image reconstruction for computed
652 tomography. *Journal of X-ray science and technology*, 25:
653 113–134, 2017. 3
- 654 [40] Niu Shaohua, Gao Yan, Bian Zhaoying, Huang Jing, Chen
655 Wufan, Yu Hengyong, Liang Zhengrong, and Ma Jianhua.
656 Sparse-view x-ray CT reconstruction via total generalized
657 variation regularization. *PMB*, 59:2997–3017, 2014. 3
- 658 [41] Anuroop Sriram, Jure Zbontar, Tullie Murrell, Aaron De-
659 fazio, C. Lawrence Zitnick, Nafissa Yakubova, Florian
660 Knoll, and Patricia Johnson. End-to-End variational net-
661 works for accelerated MRI reconstruction. In *MICCAI*,
662 pages 64–73, 2020. 2, 3, 5
- 663 [42] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet,
664 Scott Reed, Dragomir Anguelov, Dumitru Erhan, and Vin-
665 cent Vanhoucke Andrew Rabinovich. Going deeper with
666 convolutions. In *CVPR*, pages 1–9, 2015. 2, 5, 8
- 667 [43] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas
668 Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung,
669 Andreas Peter Steiner, Daniel Keysers, Jakob Uszkoreit,
670 Mario Lucic, and Alexey Dosovitskiy. MLP-mixer: An all-
671 MLP architecture for vision. In *NeurIPS*, 2021. 2, 5, 8
- 672 [44] Gomi Tsutomu and Koibuchi Yukio. Use of a Total Varia-
673 tion minimization iterative reconstruction algorithm to eval-
674 uate reduced projections during digital breast tomosynthesis.
675 *BioMed Research International*, 2018:1–14, 2018. 3
- 676 [45] Ce Wang, Kun Shang, Haimiao Zhang, Qian Li, and S. Kevin
677 Zhou. DuDoTrans: Dual-domain transformer for sparse-
678 view CT reconstruction. In *Machine Learning for Medical*
679 *Image Reconstruction*, pages 84–94. Springer International
680 Publishing, 2022. 2, 3, 6, 7, 8, 4
- 681 [46] Ge Wang, Hengyong Yu, and Bruno De Man. An outlook on
682 X-ray CT research and development. *Medical Physics*, 35:
683 1051–1064, 2008. 2
- 684 [47] Jiaxi Wang, Li Zeng, Chengxiang Wang, and Yumeng Guo.
685 ADMM-based deep reconstruction for limited-angle CT.
686 *PMB*, 64, 2019. 2, 3
- 687 [48] Wu Weiwen, Hu Dianlin, Niu Chuang, Yu Hengyong, Vard-
688 hanabhuti Varut, and Wang Ge. DRONE: Dual-domain
689 residual-based optimization network for sparse-view CT re-
690 construction. *IEEE TMI*, 40:3002–3014, 2021. 2, 3
- 691 [49] Dufan Wu, Kyungsang Kim, Georges El Fakhri, and
692 Quanzheng Li. Iterative low-dose ct reconstruction with pri-
693 ors trained by artificial neural network. *IEEE TMI*, 36:2479–
694 2486, 2017. 3
- 695 [50] Wenjun Xia, Ziyuan Yang, Qizheng Zhou, Zexin Lu,
696 Zhongxian Wang, and Yi Zhang. Transformer-based itera-
697 tive reconstruction model for sparse-view CT reconstruction.
698 In *MICCAI*, 2022. 2, 3, 4, 5, 6, 7, 8
- 699 [51] Ke Yan, Xiaosong Wang, Le Lu, and Ronald M. Summers.
700 DeepLesion: Automated mining of large-scale lesion an-
701 notations and universal lesion detection with deep learning.
702 *Journal of Medical Imaging*, 5:036501, 2018. 6
- 703 [52] Zhang Yi, Chen Hu, Xia Wenjun, Chen Yang, Liu Baodong,
704 Liu Yan, Sun Huaiqiang, and Zhou Jiliu. LEARN++: Re-
705 current dual-domain reconstruction network for compressed
706 sensing CT. *IEEE TRPMS*, 7:132–142, 2023. 2, 3, 4, 5
- 707 [53] Tsai Yu-Jung, Bousse Alexandre, Ehrhardt Matthias J.,
708 Stearns Charles W., Ahn Sangtae, Hutton Brian F., Arridge
709 Simon, and Thielemans Kris. Fast quasi-newton algorithms
710 for penalized reconstruction in emission tomography and
711 further improvements via preconditioning. *IEEE TMI*, 37:
712 1000–1010, 2018. 2, 4
- 713 [54] Zhicheng Zhang, Xiaokun Liang, Xu Dong, Yaoqin Xie,
714 and Guohua Cao. A sparse-view CT reconstruction method
715 based on combination of DenseNet and deconvolution. *IEEE*
716 *TMI*, 37:1407–1417, 2018. 2, 3

QN-Mixer: A Quasi-Newton MLP-Mixer Model for Sparse-View CT Reconstruction

Supplementary Material

717	1. Inverse Hessian approximation	1
718	1.1. Approximation via optimization	1
719	1.2. Validation of adherence to BFGS	2
720	2. More Ablation Study and Analysis	3
721	2.1. Ablation on Incept-Mixer	3
722	2.2. More visualization results	3
723	2.3. Iterative results visualization	4
724	2.4. More OOD results and visualization	4
725	2.5. Reconstruction error visualization	5
726	2.6. Noise Power Spectrum analysis	5
727	3. Reproducibility	5
728	3.1. QN-Mixer pseudo-code	5
729	3.2. External libraries used	7
730	3.3. QN-Mixer's parameters initialization	7
731	3.4. Incept-Mixer's architecture	7
732	3.5. AAPM dataset splits	7
733	3.6. Robustness eval. protocol for OOD scenarios	8
734	4. Limitations	8
735	5. Notations	8

1. Inverse Hessian approximation

1.1. Approximation via optimization

The fundamental idea is to iteratively build a recursive approximation by utilizing curvature information along the trajectory. It is crucial to emphasize that a quadratic approximation offers a direction that can be leveraged within the iterative update scheme. This direction is defined by the equation:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \alpha_t \mathbf{d}_t. \quad (8)$$

In order to determine the direction \mathbf{d}_t , we can employ a quadratic approximation of the objective function. This approximation can be expressed as:

$$J(\mathbf{x}_t + \mathbf{d}) \approx m_t(\mathbf{d}) = J(\mathbf{x}_t) + \nabla J(\mathbf{x}_t)^\top \mathbf{d} + \frac{1}{2} \mathbf{d}^\top \mathbf{B}_t \mathbf{d}, \quad (9)$$

where $J(\mathbf{x}_t)$ represents the objective function evaluated at the current point \mathbf{x}_t , $\nabla J(\mathbf{x}_t)$ denotes the gradient of the objective function at \mathbf{x}_t , $\mathbf{B}_t \in \mathbb{R}^{n \times n}$ corresponds to the approximation of the Hessian matrix. By minimizing the right-hand side of the quadratic approximation in Eq. (9), we can determine the optimal direction \mathbf{d}_t . Taking the

derivative of $m_t(\mathbf{d})$ with respect to \mathbf{d} and setting it to zero, we obtain:

$$\frac{\nabla m_t(\mathbf{d})}{\nabla \mathbf{d}} = \mathbf{d}_t \mathbf{B}_t + \nabla J(\mathbf{x}_t) \xrightarrow{\nabla m_t(\mathbf{d})=0} \mathbf{d}_t = -\mathbf{B}_t^{-1} \nabla J(\mathbf{x}_t), \quad (7)$$

by substituting this result in Eq. (8) we obtain:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha_t \mathbf{B}_t^{-1} \nabla J(\mathbf{x}_t). \quad (8)$$

The objective is to ensure that the curvature along the trajectory is consistent. In other words, at the last two iterations, m_{t+1} should match the gradient $\nabla J(\mathbf{x}_t)$ in the following way:

$$\nabla m_{t+1} \Big|_{\mathbf{d}=0} = \nabla J(\mathbf{x}_{t+1}), \quad \nabla m_{t+1} \Big|_{\mathbf{d}=-\alpha_t \mathbf{d}_t} = \nabla J(\mathbf{x}_t). \quad (9)$$

This condition ensures that the quadratic approximation captures the correct curvature information along the trajectory, allowing for accurate optimization and convergence of the algorithm. By evaluating $\nabla m_{t+1}(\cdot)$ at the point $-\alpha_t \mathbf{d}_t$ we obtain:

$$\alpha_t \mathbf{B}_{t+1} \mathbf{d}_t = \nabla J(\mathbf{x}_{t+1}) - \nabla J(\mathbf{x}_t). \quad (10)$$

From Eq. (8) we get the secant equation:

$$\mathbf{B}_{t+1} (\underbrace{\mathbf{x}_{t+1} - \mathbf{x}_t}_{\mathbf{s}_t}) = \underbrace{\nabla J(\mathbf{x}_{t+1}) - \nabla J(\mathbf{x}_t)}_{\mathbf{z}_t} \rightarrow \mathbf{B}_{t+1} \mathbf{s}_t = \mathbf{z}_t. \quad (11)$$

To avoid explicitly computing the inverse matrix \mathbf{B}_t^{-1} , we can introduce an approximation $\mathbf{H}_t = \mathbf{B}_t^{-1}$ and optimize it as follows:

$$\begin{aligned} \mathbf{H}_{t+1} &= \arg \min_{\mathbf{H}} \|\mathbf{H} - \mathbf{H}_t\|_W^2 \quad \triangleright \mathbf{H}_{t+1} \text{ close to } \mathbf{H}_t \\ \text{s.t.: } \mathbf{H} &= \mathbf{H}^\top \quad \triangleright \text{symmetry} \\ \mathbf{H} \mathbf{z}_t &= \mathbf{s}_t \quad \triangleright \text{secant equation} \end{aligned} \quad (10)$$

Here $\|\cdot\|_W^2$ denotes the weighted Frobenius norm. This optimization problem aims to find an updated approximation \mathbf{H}_{t+1} that is close to \mathbf{H}_t , while satisfying the constraints that \mathbf{H}_{t+1} is symmetric and satisfies the secant equation $\mathbf{H}_{t+1} \mathbf{z}_t = \mathbf{s}_t$. BFGS [9, 12, 19] uses $\mathbf{W} = \int_0^1 \nabla^2 J(\mathbf{x}_t + t\alpha_t \mathbf{d}_t) dt$, to solve this optimization problem and obtain the iterative update of \mathbf{H} :

$$\mathbf{H}_{t+1} = (\mathbf{I} - \rho_t \mathbf{s}_t \mathbf{z}_t^\top) \mathbf{H}_t (\mathbf{I} - \rho_t \mathbf{z}_t \mathbf{s}_t^\top) + \rho_t \mathbf{s}_t \mathbf{s}_t^\top, \quad (11)$$

where $\rho_t = \frac{1}{\mathbf{z}_t^\top \mathbf{s}_t}$, $\mathbf{s}_t = \mathbf{x}_{t+1} - \mathbf{x}_t$, and $\mathbf{z}_t = \nabla J(\mathbf{x}_{t+1}) - \nabla J(\mathbf{x}_t)$. This update equation allows us to iteratively refine the approximation \mathbf{H}_t based on the current gradient information and the changes in the solution.

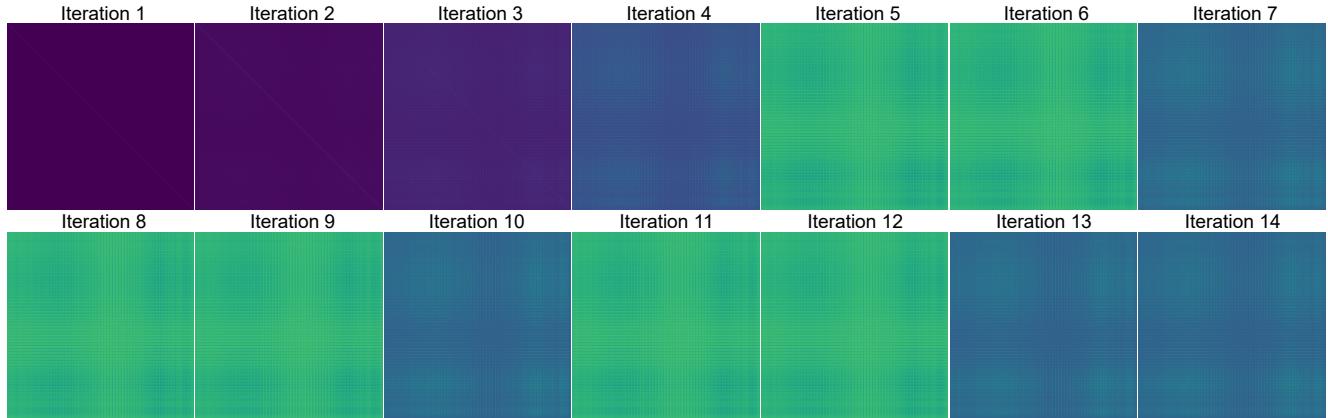


Figure 7. **Visualization of the inverse Hessian approximation across iterations.** Observe the subtle changes between each iteration, attributed to the influence of the objective function used to estimate H_t (see Eq. (10)).

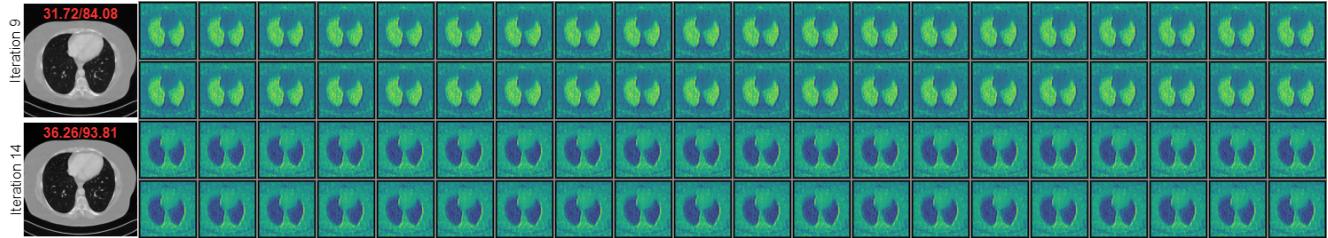


Figure 8. **Inverse Hessian approximation rows visualization.** We present the first 40 rows for the 9th and 14th inverse Hessian approximations on the first and second lines, respectively. Each row is of size 64^2 , reshaped into a 64×64 image. The corresponding image reconstructions are shown on the left, along with PSNR (dB) and SSIM (%) values at the top.

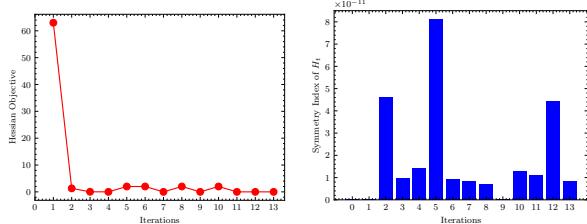


Figure 9. **Inverse Hessian matrix approximation algorithm.** Verification of requirements over the iterations. Left: Objective function value in Eq. (10); Right: Symmetry index of the inverse Hessian approximation refer to Eq. (12).

790

1.2. Validation of adherence to BFGS

791 We validate the adherence of our method to the BFGS re-
 792 quirements. To achieve this, we present the constraint val-
 793 ues of the optimization algorithm given in Eq. (10) using
 794 a test set image from AAPM, as illustrated in Fig. 9. The
 795 symmetry index is defined as follows:

$$796 \quad SI = \frac{1}{n \cdot (n-1)} \sum_{i=1}^n \sum_{j=1, j \neq i}^n |A_{ij} - A_{ji}|. \quad (12)$$

797 Our results demonstrate the effectiveness of our approach
 798 in satisfying the essential conditions required by the BFGS
 799 algorithm. Notably, the symmetry index is consistently
 800 close to zero, indicating the symmetry of the matrix H_t

at each iteration, which is the first constraint of the BFGS method. Furthermore, with regard to the objective function value, it is evident that it is close to zero, except for the initial approximation. This deviation can be attributed to the use of the identity matrix as the starting point.

801
802
803
804
805

Inverse Hessian matrix approximation visualizations. Figure 7 depicts H_t at different iterations. These visualizations confirm the required symmetry of the matrix in each iteration. Additionally, the matrix H_t is close to the identity matrix at the second iteration, becoming more structured in the third iteration. This behavior aligns with expectations, as the matrix H_t is initialized as the identity matrix and updated based on gradient information and solution changes.

806
807
808
809
810
811
812
813

Visualization of reshaped rows. To further understand the inverse Hessian matrix approximation structure, we depict the reshaped (64×64) first 40 rows of H_t at iterations 9 and 14 in Fig. 8. These rows store gradient attention information used for updating the solution, consistent with the matrix H_t being updated based on gradient information and solution changes. In future work, we plan to explore the impact of H_t on the optimization process and its influence on reconstruction performance.

814
815
816
817
818
819
820
821
822

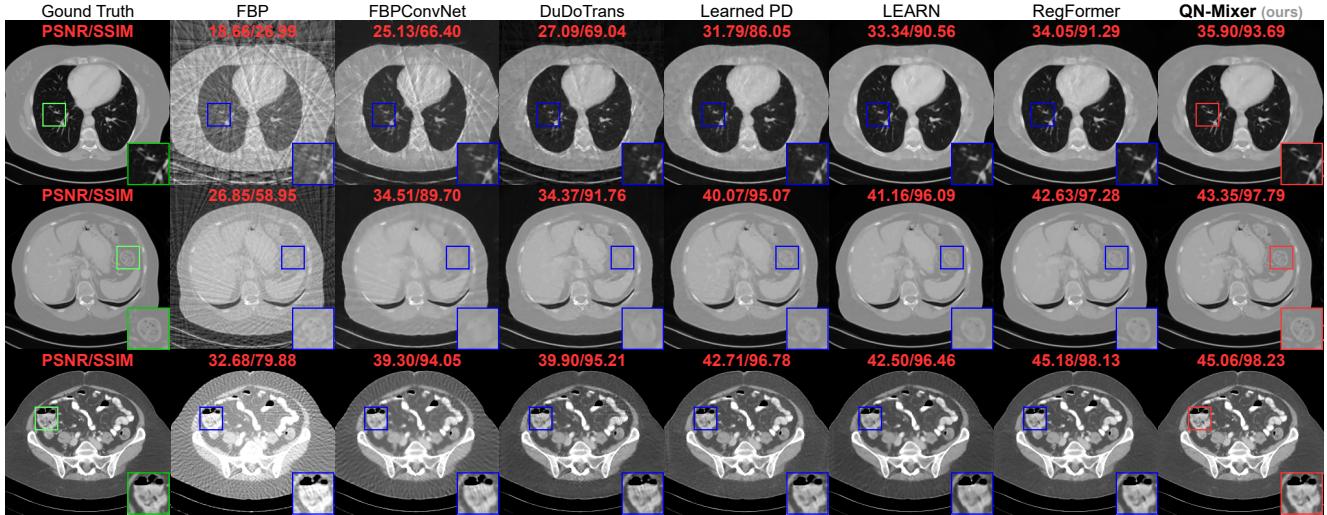


Figure 10. **Visual comparison on AAPM.** From top to bottom: the results under the following conditions: first ($n_v = 32, N_1$), second ($n_v = 64, N_1$), third ($n_v = 128, N_1$). The display window is set to $[-1000, 800]$ HU for the first two rows and to $[-200, 300]$ HU for the last row.

2. More Ablation Study and Analysis

2.1. Ablation on Incept-Mixer

We further investigate the impact of the hyperparameters of Incept-Mixer on the reconstruction performance. We vary the patch size p and the number of stacked Mixer layers N and report the results in Tab. 7a, and Tab. 7b respectively.

Impact of the path size. We observe that increasing the patch size p from 2 to 4 improves the performance (+1.28 dB and +1.04%) while further increasing the patch size from 4 to 8 decreases the performance (-1.32 dB and -0.79%).

We attribute this observed pattern to the trade-off between local and global features in the reconstruction process. When the patch size is small, such as $p = 2$, the model focuses on capturing fine-grained local details, which can enhance reconstruction accuracy. As the patch size increases to $p = 4$, the network gains a broader perspective by considering larger regions, leading to an improvement in performance. However, when the patch size becomes too large, for example, $p = 8$, the model might start incorporating more global context at the expense of losing finer details. This can result in a decrease in performance as the model becomes less sensitive to localized patterns.

Impact of the number of stacked Mixer layer. We observe that increasing the number of stack N from 1 to 2 improves the performance (+1.86 dB and +1.31%), while further increasing the patch size from 2 to 3 decreases the performance (-1.03 dB and -0.60%) and from 3 to 4 de-

	p	N	PSNR \uparrow	SSIM \uparrow
(a)	1	1	37.64	94.79
	2	2	39.51	96.11
	4	3	38.47	95.51
	8	4	38.17	95.40
(b)				

Table 7. **Ablation of Incept-Mixer.** (a) p is the patch size; (b) N is the number of stacked Mixer layers. The best performance is attained using $p = 4$ and $N = 2$.

creases the performance (-0.30 dB and -0.11%).

Similarly, when varying the number of stacked Mixer layers N , we observe a trend where an increase in N initially contributes to improved performance, as the model can capture more complex features and relationships. However, as N continues to grow, the network may encounter diminishing returns, and the benefits of additional layers diminish, potentially leading to overfitting or increased computational overhead.

Robustness to hyperparameters. Hence, there exists an optimal trade-off between the patch size p and the number of stacked Mixer layers N , but the model performs similarly for a wide range of values. In our experiments, we use $p = 4$ and $N = 2$ for all the datasets, which highlights the robustness of our method to these hyperparameters.

2.2. More visualization results

Fig. 10 displays supplementary visualizations of our approach on AAPM. Our method consistently produces high-

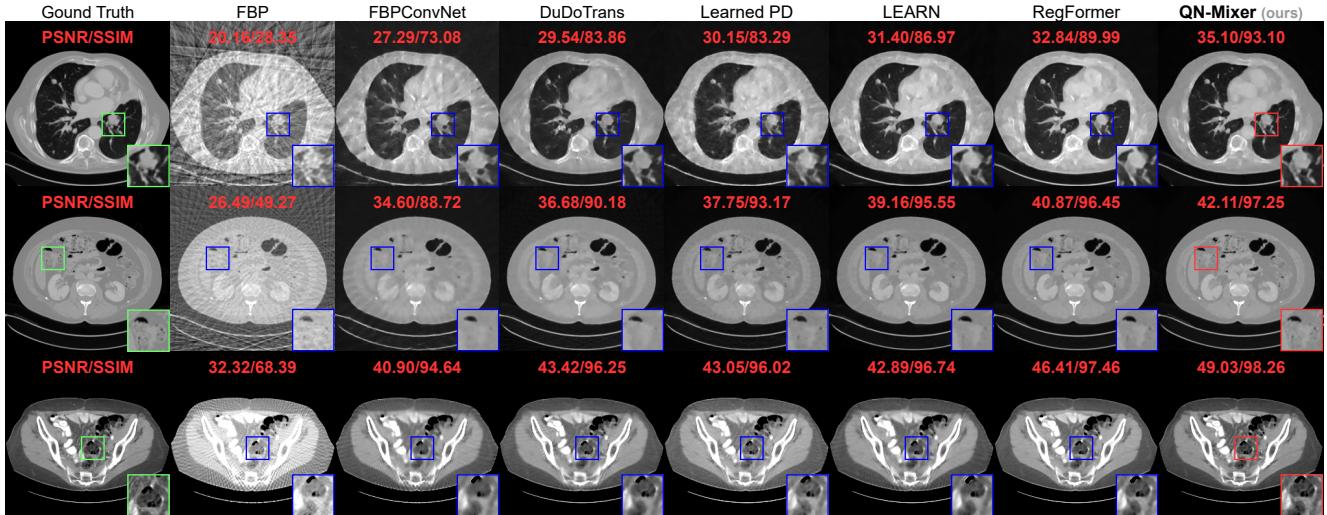


Figure 11. **Visual comparison on DeepLesion.** From top to bottom: the results under the following conditions: first ($n_v = 32, N_1$), second ($n_v = 64, N_1$), third ($n_v = 128, N_1$). The display window is set to $[-1000, 800]$ HU for the first two rows and to $[-200, 300]$ HU for the last row.

quality reconstructions across all views. Notably, among state-of-the-art techniques, QN-Mixer excels in reconstructing fine-grained details. For instance, it accurately captures small vessels in the first row, delicate soft tissue structures in the second row, and sharp boundaries in the third row.

In Fig. 11, we showcase additional visualizations of our method applied to DeepLesion. QN-Mixer demonstrates superior performance across all views, yielding high-quality reconstructions. This is particularly evident in the challenging scenario of 32 views, where our method outperforms others in capturing fine-grained details, such as small vessels and lesions. Importantly, these results are achieved with fewer iterations compared to alternative unrolling networks like RegFormer.

2.3. Iterative results visualization

In order to demonstrate the effectiveness of QN-Mixer, we present a series of intermediate reconstruction results in Fig. 12. These results illustrate the progression of the reconstruction process at different iterations of our method. By examining the reconstructed outputs at each iteration, our goal is to offer insights into the evolution of image quality. Notably, we observe that the improvement in quality, as quantified by the PSNR and SSIM values of each iteration, does not consistently increase with each iteration (see Iteration 10 in Fig. 12). We suspect that the observed unexpected behavior may arise from the variation of the objective function (i.e. Eq. (2)) around the point t in the unrolled network, which is dependent on a learnable gradient regularization term Incept-Mixer.

Method	$n_v = 32$		$n_v = 64$		$n_v = 128$	
	SSIM \uparrow	PSNR \uparrow	SSIM \uparrow	PSNR \uparrow	SSIM \uparrow	PSNR \uparrow
FBP	72.88	18.97	83.42	22.13	91.75	24.85
FBPCovNet [17]	63.91	20.94	73.02	24.12	80.60	25.74
DuDoTrans [45]	60.51	19.09	79.75	25.00	85.65	27.23
Learned PD [1]	67.99	21.92	83.79	25.51	85.42	25.86
LEARN [6]	79.70	24.46	84.44	26.74	88.16	26.20
RegFormer [50]	72.45	23.69	77.33	25.46	84.99	28.22
QN-Mixer (ours)	86.17	25.95	94.56	30.95	97.04	33.98

Table 8. **Quantitative results of the reconstruction of the cropped OOD circle.** **Bold:** Best, under: second best.

2.4. More OOD results and visualization

In our initial analysis, we assess the robustness of methods to out-of-distribution (OOD) data using complete patient images, computing SSIM and PSNR metrics for the entire image. While achieving the best performance across all views, we observe a degradation in performance for all methods when focusing on the circle region, as expected due to its complexity.

To address this, we extend our evaluation to specific regions, those containing the white circles. This targeted approach isolates the reconstruction performance exclusively to the circle region. Our experiments involve a randomly selected 5 samples from the AAPM test set depicted in Fig. 13. The overall performance across the complete set of 214 patient images is summarized in Tab. 8.

Our method significantly outperforms the second-best across all views in both SSIM and PSNR. For the most challenging case of 32 views, we surpass the second best by +6.47% and +1.49 dB. With 64 views, our performance exceeds the second best by +10.12% and +4.21 dB. In the

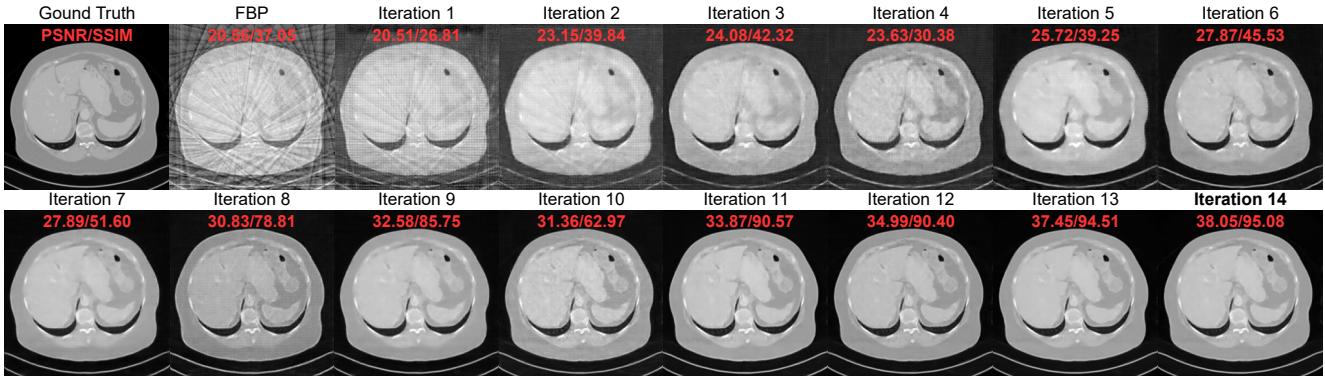


Figure 12. **QN-Mixer’s intermediate reconstructions** using AAPM with $(n_v = 32, N_1)$. Display window is set to $[-1000, 800]$ HU.

918 easiest case of 128 views, we outperform the second best by
 919 +5.29% and +5.76 dB. As anticipated, all methods exhibit
 920 degraded performance when focusing on the circle region,
 921 and the gap between our method and the second-best widens
 922 compared to the complete image. Moreover, the numerical
 923 results in Tab. 8 align with the visualizations in Fig. 13,
 924 reinforcing the robustness of our method in reconstructing
 925 abnormal data.

926 2.5. Reconstruction error visualization

927 We present the reconstruction error of QN-Mixer in com-
 928 parison to LEARN and RegFormer in Fig. 14. The images
 929 are organized from left to right based on the SSIM value. As
 930 illustrated, our method consistently produces high-quality
 931 reconstructions. In the most challenging scenario ($n_v = 32$)
 932 with no added noise, and for the least favorable image, our
 933 method achieves a reconstruction with an SSIM of 93.53%,
 934 maintaining notably satisfactory performance compared to
 935 RegFormer with an SSIM of 91.30%. For the best recon-
 936 struction across all methods, our method achieves an SSIM
 937 of 97.72%, while RegFormer achieves an SSIM of 97.48%.
 938 These results demonstrate the robustness of our method
 939 when dealing with challenging scenarios.

940 2.6. Noise Power Spectrum analysis

941 We conducted a comprehensive examination of the noise
 942 characteristics in our reconstructed images through noise
 943 power spectrum (NPS) analysis. NPS serves as a metric,
 944 quantifying the magnitude and spatial correlation of noise
 945 properties, or textures, within an image. It is derived from
 946 the Fourier transform of the spatial autocorrelation function
 947 of a zero-mean noise image.

948 NPS analysis was performed on a configuration of Re-
 949 gions of Interest (ROIs) as depicted in Fig. 16. This process
 950 was applied to all 214 images from the AAPM test set and
 951 for three different views (32, 64, and 128). The average 1D
 952 curves were generated by radially averaging the 2D NPS
 953 maps, and the results are presented in Fig. 15.

954 The area under the NPS curve is equal to the square of
 955 the noise magnitude. Importantly, the ordering of methods
 956 based on noise magnitude corresponds to the ranking ob-
 957 served in our quantitative experiments for PSNR and SSIM
 958 in the main text. For example, FBP, which exhibits the low-
 959 est noise magnitude, also performs the poorest in terms of
 960 PSNR and SSIM. Conversely, our method, with the high-
 961 est noise magnitude, stands out as the top performer in
 962 both PSNR and SSIM metrics. Furthermore, the mean and
 963 peak frequencies serve as key indicators of noise texture or
 964 “noise grain size”, where higher frequencies denote finer
 965 texture. Remarkably, our method showcases superior mean
 966 and peak frequencies compared to other methods, suggest-
 967 ing a finer noise texture or smaller grain size.

968 This alignment with good clinical practice standards re-
 969 inforces the robust performance of our method in capturing
 970 and preserving image details, as supported by both quanti-
 971 tative metrics and noise analysis.

972 3. Reproducibility

973 All our experiments are fully reproducible. While the
 974 complete algorithm is already provided in the main pa-
 975 per (see Algorithm 2), we additionally present a PyTorch
 976 pseudo-code for enhanced reproducibility in Sec. 3.1. We
 977 furnish comprehensive references to all external libraries
 978 used in Sec. 3.2. Detailed information regarding the ini-
 979 tialization of our model can be found in Sec. 3.3. The pre-
 980 cise parameters of our regularizer, Incept-Mixer architec-
 981 ture, are available in Sec. 3.4. We outline the exact data
 982 splits utilized across the paper for the AAPM dataset in
 983 Sec. 3.5. Lastly, to facilitate the reproduction of our out-of-
 984 distribution (OOD) protocol, we provide the pseudo-code in
 985 Sec. 3.6.

986 3.1. QN-Mixer pseudo-code

987 Our QN-Mixer algorithm is introduced in Algorithm 2, and
 988 for improved reproducibility, we present a PyTorch pseudo-
 989 code in Algorithm 3. The fundamental concept underly-

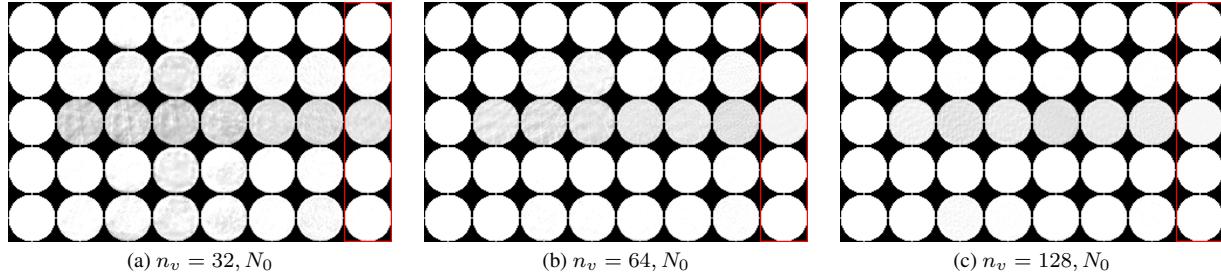


Figure 13. **Visualization of 5 samples of the OOD circle texture reconstruction.** From each figure and from left to right, we show the ground truth, FBP, FBConvNet, DuDoTrans, Learned PD, LEARN, RegFormer and QN-Mixer.

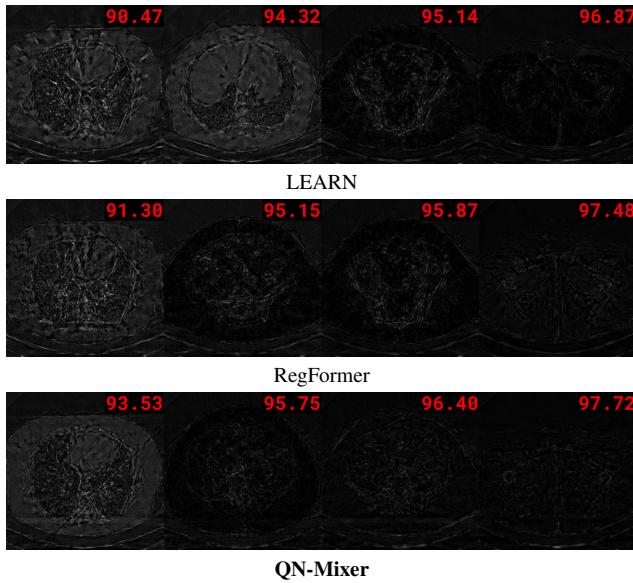


Figure 14. **Reconstruction errors** with LEARN, RegFormer, and QN-Mixer using $(n_v = 32, N_0)$. Images are ordered left to right by SSIM, with the first column showing the worst reconstruction among 214 patient images. The second and third columns represent the 1/3 and 2/3 percentiles, respectively, and the last column corresponds to the best reconstruction with the highest SSIM.

ing unrolling networks lies in having a modular gradient function, denoted as $\nabla J(\mathbf{x}_t)$, which can be easily adapted to incorporate various regularization terms. Subsequently, the core element is the unrolling iteration block responsible for updating both the solution \mathbf{x}_t and the inverse Hessian approximation \mathbf{H}_t . The update of the inverse Hessian approximation is executed through the latent BFGS algorithm. Notably, each iteration call takes the physics operator as input, tasked with computing the forward and pseudo-inverse operators for the CT reconstruction problem, along with the gradient encoder and direction decoder, which are shared across all iterations. For a more in-depth understanding, refer to Algorithm 3. Note the employment of `torch.no_grad()` to inhibit the computation of gradients for the inverse Hessian approximation. Since there is

no necessity to compute gradients for this variable, given that it is updated through the latent BFGS algorithm.

Within these two modules, second-order quasi-Newton methods can be seamlessly incorporated by simply modifying the latent BFGS algorithm or the regularization term, offering flexibility to the user.

Algorithm 3: Minimal QN-Mixer pseudo-code

```

1  class GradientFunction(nn.Module):
2      def __init__(self, regularizer):
3          self.regularizer = regularizer
4          self.lambda_ = nn.Parameter(torch.zeros(1))
5
6      def forward(self, physics, y, x):
7          y_t = physics.forward_operator(x)
8          # Compute the regularization term
9          reg_x = self.regularizer(x)
10         # Compute the data fidelity term
11         y_dft = y_t - y
12         # Compute the backprojection
13         x_dft = physics.backward_operator(y_dft)
14         g = self.lambda_* x_dft + reg_x
15
16     return g
17
18 class QN_Iteration(nn.Module):
19     def __init__(self, gradient_function):
20         self.gradient = gradient_function
21
22     def latent_bfgs(self, h, s_t, z_t):
23         I = torch.eye(len(s_t))
24         rho_t = 1. / torch.dot(z_t, s_t)
25         u_t = I - torch.outer(s_t, z_t) * rho_t
26         d_t = I - torch.outer(z_t, s_t) * rho_t
27         return (torch.matmul(u_t, torch.matmul(h, d_t))
28                 + (torch.outer(s_t, s_t) * rho_t))
29
30     def forward(self, physics, encoder, decoder,
31                y, x, h, r, is_last):
32         # Compute latent direction s_t
33         s_t = -torch.matmul(h, r)
34         d = decoder(s_t)
35
36         # Update the reconstruction
37         x = x + d
38         # Return x if it is the last iteration
39         if is_last:
40             return x, h, r
41         else:
42             r_p = encoder(self.gradient(physics, y, x))
43             z_t = r_p - r
44             with torch.no_grad():
45                 h = self.latent_bfgs(h, s_t, z_t)
46             return x, h, r_p

```

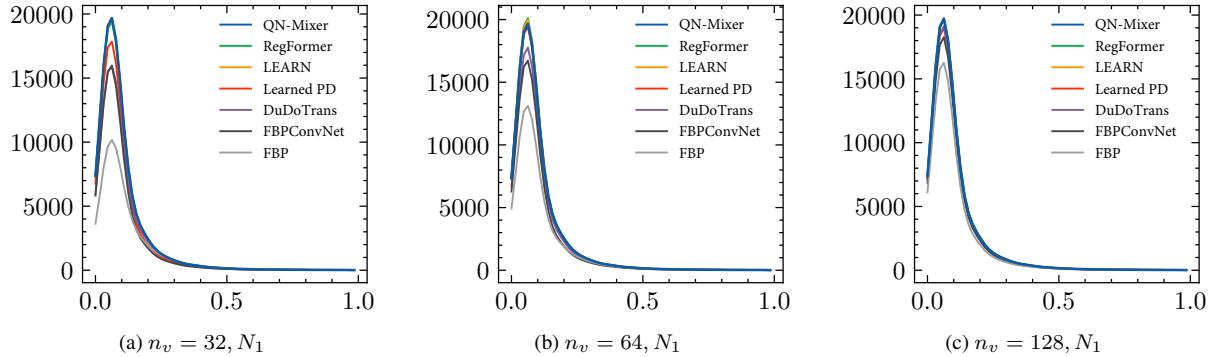


Figure 15. **Noise Power Spectrum (NPS) Analysis** in comparison to state-of-the-art methods. The x-axis represents normalized frequency in cycles per pixel (px^{-1}), and the y-axis represents noise power spectrum ($\text{HU}^2 \text{px}^2$). Display windows are configured as $[-1000, 800]$ HU. Mean and peak frequencies are intricately linked to noise texture, with finer textures correlating to higher mean and peak frequencies in the NPS. Our method exhibits the highest peak frequencies, indicating that our reconstructed images feature the most refined noise texture among all compared methods.

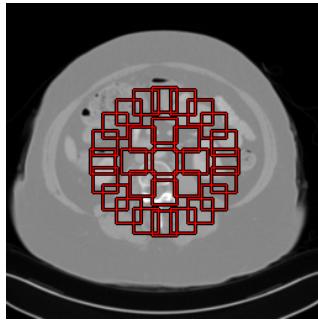


Figure 16. **ROIs for NPS Analysis:** Red squares denote 20×20 pixel ROIs distributed evenly across two circular regions. The first circle (radius 25) holds 8 ROIs, and the second circle (radius 50) has 20 ROIs. Both circles, centered at the image center, include a total of 29 ROIs per image. This standard positioning in the CT community underscores the clinical diagnostic importance of the image center.

1011

3.2. External libraries used

1012
1013

We utilized the following external libraries to implement our framework and conduct our experiments:

1014

- Operator Discretization Library (ODL):

1015

<https://github.com/odlgroup/odl>

1016

- High-Performance GPU Tomography Toolbox (ASTRA):

1017

<https://www.astra-toolbox.com/>

1018

- Medical Imaging Python Library (Pydicom):

1019

<https://pydicom.github.io/>

1020

3.3. QN-Mixer’s parameters initialization

1021
1022
1023
1024
1025

To enhance reproducibility, we provide the parameters initialization of QN-Mixer. *First*, for the gradient function, we initialize the CNNs of Incept-Mixer using the Xavier uniform initialization. The multi-layer perceptron of the MLP-Mixer is initialized with values drawn from a truncated normal distribution with a standard deviation of 0.02.

The λ_t values are initialized to zero, and the inverse Hessian approximation H_0 is initialized with the identity matrix I . *Second*, for the latent BFGS, both the encoder and decoder CNNs are initialized with the Xavier uniform initialization.

3.4. Incept-Mixer’s architecture

For enhanced reproducibility, we present the architecture of Incept-Mixer in Tab. 10. The Incept-Mixer architecture consists of a sequence of Inception blocks, followed by Mixer blocks. Each Mixer block comprises a channel-mixing MLP and a spatial-mixing MLP. The MLPs are constructed with a fully-connected layer, a GELU activation function, and another fully-connected layer. Ultimately, the regularization value is projected to the same dimension as the input image through a patch expansion layer, which is composed of a fully-connected layer and a CNN layer.

Patient ID	L067	L109	L143	L192	L286	L291	L096	L506	L333	L310
#slices	224	128	234	240	210	343	330	211	244	214
Training	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
Validation	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Testing	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

Table 9. **AAPM dataset split specification.** The validation set comprises images from patient L333, and testing utilizes images from patient L310. The images from the remaining patients have been designated for training purposes.

3.5. AAPM dataset splits

In our experiments, we use the AAPM 2016 Clinic Low Dose CT Grand Challenge public dataset [32], which holds substantial recognition as it was formally established and authorized by the esteemed Mayo Clinic. To ensure the integrity of our evaluation process, we followed the precedent set by [6, 50] and created the training set using data

1049 from eight patients, while reserving a separate patient for
 1050 the testing and validation sets. This approach guarantees
 1051 that no identity information is leaked during test time. Our
 1052 specification is presented in Tab. 9.

1053 3.6. Robustness eval. protocol for OOD scenarios

Algorithm 4: add_circle_ood pseudo-code.

```

1 def add_circle_ood(img, value=1):
2     h, w = img.shape[::-1][2]
3     radius = np.random.randint(5, 20)
4     c_x = np.random.randint(radius, w-radius)
5     c_y = np.random.randint(radius, h-radius)
6     center = (c_x, c_y)
7
8     Y, X = np.ogrid[:h, :w]
9     dist_x = (X - center[0])**2
10    dist_y = (Y - center[1])**2
11    dist_from_center = np.sqrt(dist_x + dist_y)
12    mask = dist_from_center <= radius
13    img[0, mask] = value
14    return img

```

1054 In the main paper, we propose a novel protocol specifically
 1055 crafted for evaluating the effectiveness of reconstruction
 1056 methods when handling abnormal data lying outside the
 1057 distribution of the training dataset. Assessing the
 1058 model’s capability to reconstruct abnormal data holds
 1059 significant relevance in clinical applications, where patient

Stage	Layers	#Param (k)	Output size
Input	-	-	1 × 256 × 256
InceptionBlock-1	convblock1: conv1-1: K1C16S1P0 prelu1-1 conv2-1: K1C16S1P0 prelu2-1 convblock2, prelu2-2: K3C32S1P1 prelu2-2 conv3-1: K1C16S1P0 prelu3-1 conv3-2: K5C32S1P2 prelu3-2 maxpool4-1: K3S1P1 convblock4: conv4-1: K1C16S1P0 prelu4-1	17.6	96 × 256 × 256
PatchEmbed-2	conv2-1: K4C96S4P0 rearrange2-1: bchw → bhwc layernorm3-1: D96 rearrange3-1: bhwc → bcwh linear3-1: D64O256 heightmlp3-1: gelu3-1 linear3-2: D256O64 rearrange3-2: bcwh → bchw linear3-3: D64O256 widthmlp3-1: gelu3-2 linear3-4: D256O64 rearrange3-3: bchw → bhwc layernorm3-2: D96 linear3-5: D96O384 channelmlp3-1: gelu3-3 linear3-6: D384O96	145.5	96 × 64 × 64
MixerLayer-3 × (N = 2)	linear4-1: D96O1536 layernorm4-1: D96 conv4-1: K1C1S1P0	140.8 × 2	96 × 64 × 64
PatchExpand-4	linear4-1: D96O1536 layernorm4-1: D96 conv4-1: K1C1S1P0	147.7	1 × 256 × 256

Table 10. **Incept-Mixer architecture.** K-C-S-P represents the kernel, channel, stride, and padding configuration of CNNs, while D-O indicates the input and output dimensions of linear layers.

1060 data may exhibit deviations from the characteristics present
 1061 in the training data. We strongly advocate for future re-
 1062 search endeavors to embrace and employ this protocol as
 1063 a standard for evaluating the robustness of reconstruction
 1064 methods. To facilitate the seamless integration of this proto-
 1065 col, we furnish the function’s pseudo-code in Algorithm 4.

4. Limitations

1066 Our approach inherits similar limitations from prior meth-
 1067 ods [6, 50]. First, our method entails a prolonged optimiza-
 1068 tion time, stemming from the utilization of unrolling recon-
 1069 struction networks [6, 50], in contrast to post-processing-
 1070 based denoising methods [17, 45]. While our method repre-
 1071 sents the fastest unrolling network, there is still a need to ad-
 1072 dress the existing gap. Integrating Limited-memory BFGS
 1073 into our QN-Mixer framework is an interesting research di-
 1074 rection for accelerating training. Second, while we have as-
 1075 sessed our method using the well known AAPM low-dose
 1076 and DeepLesion datasets and compared it with several state-
 1077 of-the-art methods, the evaluation is conducted on images
 1078 representing specific anatomical regions (thoracic and ab-
 1079 dominal images). The generalizability of our method to a
 1080 broader range of datasets, which may exhibit diverse char-
 1081 acteristics or variations, remains unclear. Third, the acquisi-
 1082 tion of paired data has always been an important concern in
 1083 clinic. Combining our approach with unsupervised training
 1084 framework to overcome this limitation can be an exciting
 1085 research direction. Finally, the incorporation of actual pa-
 1086 tient data into our training datasets raises valid privacy con-
 1087 cerns. Although the datasets we utilized underwent thor-
 1088ough anonymization and are publicly accessible, exploring
 1089 a solution that can effectively operate with synthetic data
 1090 emerges as an intriguing avenue to address this challenge.

5. Notations

1091 We offer a reference lookup table, available in Table 11,
 1092 containing notations and their corresponding shapes as dis-
 1093 cussed in this paper.

Notation	Shape	Value(s)	Description
n_v	\mathbb{N}^*	$\{32, 64, 128\}$	The number of projection views
n_d	\mathbb{N}^*	512	The number of projection detectors
h	\mathbb{N}^*	256	Height of the image
w	\mathbb{N}^*	256	Width of the image
c	\mathbb{N}^*	1	Channels of the image
l_h	\mathbb{N}^*	64	Latent height
l_w	\mathbb{N}^*	64	Latent width
$m = n_v \times n_d$	\mathbb{N}^*	$n_v \times 512$	Data (sinogram) size
$n = h \times w$	\mathbb{N}^*	256×256	Image size
$(l_h \cdot l_w) \times (l_h \cdot l_w)$	\mathbb{R}	$(64 \cdot 64) \times (64 \cdot 64)$	Size of the latent BFGS optimization variable i.e. \mathbf{H}
T	\mathbb{N}^*	14	Number of iterations of our method
t	\mathbb{N}	-	Iteration of the loop in the algorithm
\mathbf{y}	$n_v \times n_d$	-	Sparse sinogram
\mathbf{A}	$\mathbb{R}^{n \times m}$	-	The forward model (i.e. discrete Radon transform)
\mathbf{A}^\dagger	$\mathbb{R}^{m \times n}$	-	The pseudo-inverse of \mathbf{A}
\mathbf{x}_0	$\mathbb{R}^{h \times w \times c}$	$\mathbf{A}^\dagger \mathbf{y}$	Initial reconstruction
λ_t	\mathbb{R}	-	Regularization weight at step t
α_t	\mathbb{R}	-	Step size (i.e. search step)
\mathbf{x}_t	$\mathbb{R}^{h \times w \times c}$	-	Reconstructed image at iteration t
$\nabla_{\mathbf{x}} J(\mathbf{x}_t)$	$\mathbb{R}^{h \times w \times c}$	-	Gradient value at iteration t
\mathbf{H}_t	$\mathbb{R}^{(l_h \cdot l_w) \times (l_h \cdot l_w)}$	-	Approximation of the inverse Hessian matrix at iteration t
$\mathbf{I}^{n \times n}$	$\mathbb{N}^{n \times n}$	-	Identity matrix of size $n \times n$
\mathbf{f}_t	$\mathbb{R}^{h \times w \times d}$	-	Feature map after the Inception block at iteration t
\mathbf{e}_t	$\mathbb{R}^{\frac{h}{p} \times \frac{w}{p} \times d}$	-	MLP-Mixer embeddings
d	\mathbb{R}	96	Depth of features
p	\mathbb{N}^*	4	Stride and kernel size in the patchification Conv 2D net
N	\mathbb{N}^*	2	Number of stacked Mixer layers
$\mathcal{G}(\cdot)$	-	-	Learned gradient of the regularization term (i.e. the Incept-Mixer model)
$\mathcal{G}(\mathbf{x}_t)$	$\mathbb{R}^{h \times w \times c}$	-	Regularization term at step t
$\mathcal{E}(\cdot)$	-	-	The gradient encoder
$\mathcal{D}(\cdot)$	-	-	The direction decoder
k	\mathbb{N}^*	$\{2, 3, 4, 5\}$	Number of Downsampling stacks in the encoder
$f_\mathcal{E} = 2^k$	\mathbb{N}^*	$\{4, 8, 16, 32\}$	Downsampling factor of the gradient in the encoder
$w_l = w / f_\mathcal{E}$	\mathbb{N}^*	$\{64, 32, 16, 8\}$	Number of columns of the down-sampled gradient
$h_l = h / f_\mathcal{E}$	\mathbb{N}^*	$\{64, 32, 16, 8\}$	Number of rows of the down-sampled gradient
$\mathbf{r}_t = \mathcal{E}(\nabla_{\mathbf{x}} J(\mathbf{x}_t))$	$\mathbb{R}^{l_h \cdot l_w}$	-	Latent representation of the gradient
$s_t = -\mathbf{H}_t \mathbf{r}_t$	$\mathbb{R}^{l_h \cdot l_w}$	-	Direction in the latent space
$\rho_t = (\mathbf{z}_t^\top \mathbf{s}_t)^{-1}$	$\mathbb{R}^{l_h \cdot l_w}$	-	BFGS divider variable
N_0	-	-	Zero noise added to the sinogram
N_1	-	-	5% Gaussian noise, 1×10^6 intensity Poisson noise
N_2	-	-	5% Gaussian noise, 5×10^5 intensity Poisson noise

Table 11. **Lookup table of notations and hyperparameters** used in the paper.