Deep Learning with Adaptive Hyper-parameters for Low-Dose CT Image Reconstruction

Qiaqiao Ding, Yuesong Nan, Hao Gao, and Hui Ji* 

Abstract—Low-dose CT (LDCT) imaging is preferred in many applications to reduce the object’s exposure to X-ray radiation. In recent years, one promising approach to image reconstruction in LDCT is the so-called optimization-unrolling-based deep learning approach, which replaces pre-defined image prior by learnable adaptive prior in some model-based iterative image reconstruction scheme (MBIR). While it is known that setting appropriate hyper-parameters in MBIR is challenging yet important to the reconstruction quality, it does not receive enough attention in the development of deep learning methods. This paper proposed a deep learning method for LDCT reconstruction that unrolls a half-quadratic splitting scheme. The proposed method not only introduces learnable image prior built on framelet filter bank, but also learns a network that automatically adjusts the hyper-parameters to fit noise level and the data for processing. As a result, only one universal model needs to be trained in our method to process the data taken under different dose levels. Experimental evaluation on clinical patient dataset showed that the proposed method outperformed both conventional and deep-learning-based solutions by a large margin.

Index Terms—X-ray CT, Image reconstruction, Low Dose CT, Deep Neural Networks

I. INTRODUCTION

X-ray Computed Tomography (CT) is to provide high-resolution three-dimensional (3D) images of internal anatomical structures using X-ray scanning and computational tomographic imaging techniques. As excessive exposure to radiation from X-ray CT scanner may increase the risk of damage to issue, many techniques have been developed to use less radiation than conventional CT scan, including decreasing the number of projection views [1] and lowering the X-ray tube current [2]. The latter is the so-called low-dose CT (LDCT). As a result, the signal-to-noise ratio (SNR) of the measurements collected is much lower than that of conventional CT, which has a negative impact on the quality of the reconstructed image.

Given measurements from LDCT with low SNR, the images reconstructed by conventional filtered back-projection (FBP) method are often of poor quality with noticeable artifacts. In such a setting, the so-called model-based iterative reconstruction (MBIR) methods [3], [4] are more widely used for image reconstruction. In MBIR, image reconstruction is usually formulated as an optimization problem regularized by certain functional for suppressing noise magnification during reconstruction. Such a regularization term is derived from some pre-defined prior on the image. In the past, many different priors have been proposed for MBIR-based reconstruction, e.g., total variation (TV) [5]–[7], wavelet-based sparsity prior [8], [9], nonlocal sparsity prior [10], and low-rank-based patch prior [11]–[14]. The resulting optimization problems from these MBIR-based methods usually require some iterative scheme to find an approximate solution. Instead of using some man-crafted prior for regularization, there are also some works that learn image prior for regularization, e.g., dictionary learning [15] and sparsifying transform learning [16]. In recent years, there has been a rapid progress on the development of deep learning methods for image reconstruction in LDCT. Most existing methods, e.g., [17]–[19], are based on the so-called optimization unrolling, which replaces the step involving the image prior by a learnable function built on a deep neural network (NN). The performance of these optimization-unrolling-based deep learning methods is very promising in LDCT image reconstruction.

For MBIR, there are usually several hyper-parameters involved in the iterative scheme. It is known that in addition to the choice of image prior, the setting of these hyper-parameters also plays an important role in the reconstruction; see [15], [20], [21]. In other words, how to automate the selection of appropriate hyper-parameters is an important yet challenging task in MBIR. The optimal values of these hyper-parameters depend on many factors [20]–[25], including:

• SNR of measurement data;
• Content of target image for reconstruction;
• Global convergence property and fast convergence rate.

Based on these factors, there has been an enduring effort on developing automatic selection strategies of hyper-parameters for the classic non-learning-based regularization methods, e.g., [16], [20], [22]–[25].

Similar to classic MBIR-based approaches, there are also several hyper-parameters involved in existing optimization-unrolling-based deep learning methods, which are either manually tuned-up [17], [26] or treated as one part of network weights to be learned during the training [18], [19], [27]. As a result, the setting of these hyper-parameter is optimized only for one specific noise level of measurement data. When processing the data with different noise levels, they need to train different networks w.r.t. different noise levels. Such a practice can be inconvenient and sometimes difficult in practical usage when noise level of data is unknown. Furthermore, these hyper-parameters are fixed as constants for different

*Corresponding author.

Q. Ding* (e-mail: matding@nus.edu.sg), Y. Nan* (e-mail: matny@nus.edu.sg), and H. Ji* (e-mail: matjh@nus.edu.sg) are with Department of Mathematics, National University of Singapore. 119076, Singapore.

H. Gao is with Department of Radiation Oncology, University of Kansas Medical Center, Kansas City, KS 66160, USA.
measurements which correspond to the images with different contents. Such a constant treatment on hyper-parameters in existing deep learning methods [17], [28] certainly is not optimal. Motivated by the practical value of processing data with unknown noise level and the need for better image quality in LDCT imaging, this paper aims at developing an adaptive and automated hyper-parameter selection mechanism in deep-learning-based image reconstruction methods, as well as other refinements for further performance improvement.

In this paper, we proposed an image reconstruction method in LDCT whose image prior and hyper-parameter selection are collaboratively learned by two NNs. The proposed method is based on the unrolling of the so-called half-quadratic splitting method [29]. Specifically, the proposed NN is composed by \( K \) stages, which corresponds to \( K \) outer iterations in the optimization method. Each stage contains two blocks:

- **Inversion block** reconstructs an image using both the measurement and the estimate from the previous stage, whose hyper-parameters are predicted by a multi-layer perception neural network (MLP).
- **De-noising block** removes the artifacts of the estimate passed from the inverse block by a convolutional neural network (CNN).

The de-noising block, same as many existing approaches, is implemented using a CNN. However, the design of inversion block is less attentive yet crucial. In a nutshell, we are focusing on the construction of a powerful inversion block which enables the adaption to both noise level and image contents. The main difference is highlighted as follows:

- Hyper-parameters involved inner-loop is predicted by a learnable NN for the adaptivity of both dose levels and image contents.
- Fine-grained high-pass filter bank is adopted instead of often-seen gradient operator or learnable filter bank for better sub-band reconstruction and image content adaptation.

Extensive experimental studies showed these designs not only bring performance advantage in LDCT image reconstruction, in compared with existing methods by a large margin, but also allows one to train a single NN model to process the data with unknown dose level.

A. Related work

Due to space limitation, we only discuss most related methods for LDCT image reconstruction, e.g., deep-learning-based methods. One type of such methods uses deep NN to post-process the images reconstructed from some existing methods. By treating the artifacts as noise in the reconstructed images, these methods train an NN-based image denoiser to remove the artifacts from images for better quality. The denoising NN is trained by using the pairs of the image reconstructed from LDCT and its counterpart from conventional CT. Different NN architectures have been proposed for such a denoiser, e.g., CNN [30], residual encoder CNN [31], residual network [32], [33], U-net [32], [34], GAN [35], [36], multi-resolution deep convolutional framelets neural network [37]. Instead of directly denoising images, Kang et al. [38], [39] proposed to denoise wavelet transform coefficients of the images using an NN-based denoiser. Overall, in such a post-processing approach, deep learning does not get involved in the reconstruction and data consistence is omitted.

In recent years, a more popular deep-learning-based approach is built on optimization unrolling, which introduces NNs in the iterations of the MBIR by replacing image-prior-based operations with the function modeled using learnable NN [40]. In other words, the pre-defined image prior is replaced by a learnable prior using NN. There are two ways to train the NN. One is plugging a pre-trained NN into the iterations of some MBIR; see e.g. [17], [28]. The other is training the end-to-end NNs together with the MBIR; see e.g. [18], [19], [27], [41]–[45].

In addition to the training scheme, another main difference among different optimization-unrolling-based deep learning methods lies in what numerical solver is chosen for unrolling. Based on the alternating direction method of multipliers (ADMM), the ADMM-net is proposed in [27] for image reconstruction in compressed-sensing-based Magnetic Resonance Imaging (MRI). For consistent CT image reconstruction, Gupta et al. [17] proposed to unroll the proximal gradient methods with CNN-based learnable prior. For LDCT image reconstruction, the ADMM is used in [18] and the primal-dual hybrid gradient (PDHG) method is used in [19] for unrolling with learnable image prior.

Similar to most iterative schemes of MBIR, how to set appropriate hyper-parameters is very crucial to the quality of reconstructed images in these optimization-unrolling-based deep learning methods; see e.g. [17], [28]. Different from conventional methods, there are few systematic studies that address the problem of hyper-parameter setting in deep learning methods for image reconstruction.

In existing deep learning methods, the hyper-parameters are either manually tuned-up [17], [28] or treated as a part of learnable NN parameters [18], [19]. Both treatments are suboptimal. The former can only take a few trials on the hyper-parameters, since training the NN is a very time-consuming process. The later usually cannot find optimal values for hyper-parameters either, as these hyper-parameters are treated the same as millions of other NN weights in the optimization. In addition, for measurement with different dose levels, these deep learning methods need to train different models to fit a specific noise level for optimal performance. There will be a performance hit if only one model is trained for processing the data with different dose levels.

II. MEASUREMENT MODEL AND PROBLEM FORMULATION

In CT imaging with a mono-energetic source, the projection measurements from CT scan follow the Poisson distribution [46], which can be expressed as:

\[
y_i \sim \text{Poisson}\{I_i \exp(-|A z_i|)\} + \mathcal{N}(0, \sigma_i^2),
\]

where \( y = \{y_i\}_{i=1}^{N_p} \) represents the vector of measured projections. \( I_i \) represents the incident X-ray intensity incorporating X-ray source illumination and the detector efficiency. \( A = [a_{i,j}]_{i,j} \) is the \( N_q \times N_p \) system matrix, i.e. projection matrix
of CT imaging, and \( x = [x_i]_{i=1}^{N_x} \) denotes the attenuation map. The quantity \( |Ax|_i = \sum_{j=1}^{N_y} a_{ij} x_j \) refers to the line integral of the attenuation map \( x \) along the \( i \)-th X-ray. \( N \) refers to normal distribution, and the quantity \( \sigma_k^2 \) denotes the variance of the background electronic noise which is considered to be stable for a commercial CT scanner and has been converted to photon units [47].

It is noted that the noise level varies for different target images. Given a target image \( x \), its noise level is controlled by \( I_i \), i.e., the measure data is corrupted with noise which becomes larger when dose level \( I_i \) decreases. However, even with a fixed dose level \( I_i \), for different patients or different parts of human-body, their corresponding noise levels are different too.

To reconstruct the attenuation map \( x \), one can first run the correction and the logarithm transform on the noisy measurements \( \mathbf{y} \) to generate the so-called sinogram \( \mathbf{y} = [y_i]_{i=1}^{N_y} \), whose relation to \( Ax \) is often expressed as:

\[
\mathbf{y} = Ax + \epsilon, \tag{2}
\]

where \( \epsilon \sim N(0, \sigma^2) \) denotes the noise. The linear system (2) is ill-posed, and certain regularization needs to be imposed to resolve the solution ambiguity and suppress noise magnification. Let \( p(x; \lambda) \) denotes the prior distribution function of \( x \) with distribution parameter \( \lambda \). The maximum a posterior (MAP) estimation of \( x \) is then the minimum of the cost function given by

\[
\min_x \frac{1}{2} \|Ax - y\|^2_2 - \log p(x; \lambda), \tag{3}
\]

where the first is data fidelity term and the second is regularization term on \( x \) derived from its prior distribution. The model (3) is often called penalized weighted least-squares (PWLS) image reconstruction model [3], [48], [49], which is a widely used one in CT image reconstruction.

Many regularization terms \( -\log p(x; \lambda) \) have been proposed for LDCT image reconstruction, e.g. TV [6], nonlocal TV [10] and framelet [8]. These regularization functionals are usually not directly defined in image domain, but in the domain of image gradients or their generalizations. For example, both the TV and wavelet-transform-based regularizations take the form of \( \lambda \|\nabla x\|_1 \), where \( \nabla \) is the gradient operator \( \nabla \) (TV) or wavelet transform \( W \) (wavelet). The prior distribution parameter \( \lambda \) is also called the regularization parameter, which needs to be set in advance.

There are many numerical scheme for solving (3) with regularization term \( \lambda \rho(\nabla x) = -\log(p(x; \lambda)) \), e.g. ADMM and PDHG. In this paper, our work is based on the half-quadratic splitting method [29], which solve the problem (3) by introducing an auxiliary variable \( z \) with following two steps:

**Inversion:**
\[
x^k = \arg \min_x \frac{1}{\sigma^2} \|Ax - y\|^2_2 + \frac{1}{\sigma_1^2} \|\nabla x - z^k\|^2_2, \tag{4a}
\]

**Denoising:**
\[
z^{k+1} = \arg \min_z \lambda \rho(z) + \frac{1}{\sigma_1^2} \|\nabla x^k - z\|^2_2, \tag{4b}
\]

where \( \{\rho_k\}_k \) is the parameter sequences of the algorithm and \( k \) is the iteration step. The noise variance \( \sigma \), the hyper-parameters \( \{\rho_k\}_k \), and the regularization parameter \( \lambda \), will make a noticeable impact on the convergence behavior and the quality of the result.

There are two blocks in the iterative scheme (4). The first is an inversion block (4a), which reconstructs an image from the measurement \( y \) and the estimate from the previous iteration. The second is a denoising block (4b), which refines the estimate from the inverse block using the the prior-based regularization. The performance of the scheme (4) depends on the answers to the following two questions:

1. What prior \( p(x; \lambda) \) fits the target image \( x \) well?
2. What values of the hyper-parameters are optimal for estimating \( x \)?

Most existing optimization-unrolling-based methods focus on the replacement of the denoising block (4b) using a NN-based denoiser to address the first problem. However, the second problem has not attracted much attention in spite of the importance of the hyper-parameters [17]. The current treatment on the setting of hyper-parameters is done by either manual selection or learning them as one part of NN weights.

This paper aims at designing a deep learning method for LDCT reconstruction that not only uses a CNN-based denoiser to replace (4b), but also introduces a MAP-based predictor in the inversion block (4a) to automate the adaptive setting of hyper-parameters to the SNR of measurement data and the target image \( x \). The proposed deep learning method not only provides very competitive performance for LDCT image reconstruction, but also is universal such that a single trained NN model can be used for the reconstruction of CT measurements with varying dose levels.

III. Method

In this section, we give a detailed discussion on the proposed deep learning method for LDCT reconstruction.

A. Detailed algorithm of the iterative scheme (4)

Recall that each iteration of the scheme (4) has two blocks: inversion block and denoising block. For the inversion block, the problem has an analytical solution given by

\[
x^k = (A^\top A + \frac{\sigma_1^2}{\sigma_2^2} \Gamma^\top \Gamma)^{-1} (A^\top y + \frac{\sigma_2^2}{\sigma_1^2} \Gamma^\top z^k). \tag{5}
\]

In conventional methods such as TV method, the operator \( \Gamma \) is image gradient operator \( \nabla \) that convolutes the image \( x \) by two filters: \([1, -1] \ [1, -1]^\top \) and utilizes reflective padding to keep the dimension consistency [50]. Motivated by the performance improvement of framelet transform over \( \nabla \) in many image recovery tasks, we propose to use the filter bank of spline framelet transform [51]. In our implementation, the filter bank of linear B-spline framelet transform is adopted, which contains totally \( L(= 8) \) 2D high-pass filters

\[
\{f_i\}_{i=1}^L := \{h_{k_1} h_{k_2}^\top : 0 \leq k_1, k_2 \leq 2 \ \& \ \{h_0 h_0^\top\}, \tag{5}
\]

composed by the tensor product of three 1D filters:

\[
h_0 = [1, 0, 0, 1]^\top, h_1 = [-1, 0, 1, 1]^\top, h_2 = [\sqrt{2}/4, 0, -\sqrt{2}/4]^\top. \tag{6}
\]
It can be seen that such a filter bank is composed of more fine-grained 2D filters on image gradients with various difference orders and along different directions. Then, the iterative scheme (4) can be re-formulated as

\[
x^k = \arg \min_x \| Ax - y \|_2^2 + \sum_{i=1}^L \beta_i^k \| f_i \circ x - z_i^k \|_2^2, \tag{7}
\]

\[
z_i^{k+1} = \arg \min_{z_i} \| f_i \otimes x^k - z_i |\|_2^2 + \alpha \cdot \rho(z_i), \ 1 \leq i \leq L, \tag{8}
\]

where \( \beta_i^k = \sigma_i^2, \) \( \alpha_i^k = \frac{1}{\lambda_i \sigma_i^2} \) and \( \circ \) is the convolution operation [52]. The inversion block (7) is equal to solve the linear system,

\[
x^k = (A^+ A + \sum_{i=1}^L \beta_i^k F_i^T F_i)^{-1} (A^+ y + \sum_{i=1}^L \beta_i^k F_i^T z_i^k), \tag{9}
\]

where \( F_i \) denote the matrix form [52] of the convolution operator with the filter \( f_i \).

In short, the inversion block (7) reconstructs \( x \) from measurement data \( y \) and the estimate of \{\( z_i \)\}_{i=1}^L in high-pass channels using the least squares estimator, and the denoising block (8) refines the estimate of \{\( z_i \)\}_{i=1}^L in high-frequency channels by treating the artifacts as noise and suppressing it via imposed regularization.

\section*{B. Inversion block with an MLP-based adaptive predictor for hyper-parameters}

Recall that in the analytical solution of the inversion block (9), there is a sequence of hyper-parameters \{\( \beta_i^k \)\}_{i=1}^L. The setting of such a sequence makes significant impact on the intermediate output, which in turn makes noticeable impact on the quality of the final result. In this section, we present a NN-based solution to predict \{\( \beta_i^k \)\}_{i=1}^L for optimal performance.

Indeed, the step (7) can be interpreted as an MAP estimator with the observed variable \( y \). Recall that Bayes’ rule \( p(x|y, z) \propto p(y|x, z)p(x|z) \), where \( p(y|x, z) \) is the likelihood and \( p(x|z) \) is the prior. In step (7), the likelihood is then \( p(y|x, z) = p(y|x) = N(y | Ax, \sigma^2 I) \) and the prior of latent variable \( x \) follows the conditional distribution based on known variable \( z \), i.e., \( p(x|z) = \prod_{i=1}^\infty N(F_i x | z_i, \sigma^2_i I) \). Then, we have the following relationships,

\[
\begin{aligned}
\{y - Ax = n \sim N(0, \sigma^2)\}, \\
z_i^k - F_i x = e_i^k \sim N(0, (\sigma_i^k)^2), \ 1 \leq i \leq L.
\end{aligned} \tag{10a}
\]

In such a simplified case, one can have an explicit solution to the hyper-parameters: \( \beta_i^k := (\sigma_i^k)^2 - 2 \sigma^2 \).

In above formula, the ground truth \( x \) is unknown, which hinders the estimation of hyper-parameters sequence \{\( \beta_i^k \)\}. Our observation is that for unrolling-based framework, the estimated intermediate reconstruction results \( x^k \) can serve as surrogates for unknown ground truth \( x \) in stage \( k \). Mathematically, such approximation can be done via

\[
\begin{aligned}
\{n^k = y - Ax^{k-1} + A \delta^{k-1}\}, \\
e_i^k = z_i^k - F_i x^{k-1} + F_i \delta^{k-1}, \ 1 \leq i \leq L.
\end{aligned} \tag{11a}
\]

with error \( \delta^{k-1} = x^{k-1} - x \). By taking the \( \ell_2 \) norm, the parameters \{\( \sigma^2, (\sigma_i^k)^2 \)\} can be estimated from the following relationship,

\[
\begin{aligned}
\sigma^2 &= \frac{\sigma^2}{N} h_0(\|y - Ax^{k-1}\|_2^2) \tag{12a} \\
(\sigma_i^k)^2 &= \frac{\sigma^2}{N} h_i(\|z_i^k - F_i x^{k-1}\|_2^2), \ 1 \leq i \leq L. \tag{12b}
\end{aligned}
\]

where \{\( h_i \)\}_{i=0}^L denotes the correction functions that correct the exact estimation of \( \sigma^2 \) and \( (\sigma_i^k)^2 \) with \( \ell_2 \) norm of the residues due to the presence of unknown error \( \delta^{k-1} \). Thus, the parameters can be estimated via

\[
\beta_i^k = \frac{\sigma^2}{(\sigma_i^k)^2} N_i h_i(\|z_i^k - F_i x^{k-1}\|_2^2), \ 1 \leq i \leq L \tag{13}
\]

Instead of direct modeling \{\( h_i \)\}_{i=0}^L, we choose a non-linear function, MLP, to model such relationship for simplicity, i.e. we automate the estimation of the hyper-parameter sequence \{\( \beta_i^k \)\} using a learnable NN. By setting \( r_i^k = y - Ax^{k-1} \) and \( r_i^k = z_i^k - F_i x^{k-1} \), the NN, denoted by \( P_{\text{map}}(\cdot; \theta^k_P) \) with parameters \( \theta^k_P \), takes the known value \( \{\|r_i^k\|_2\}_{i=0}^L \) as the input and outputs the estimation of \{\( \beta_i^k \)\}:

\[
P_{\text{map}}(\cdot; \theta^k_P) : \{\|r_i^k\|_2\}_{i=0}^L \in \mathbb{R}^{L+1} \rightarrow \{\beta_i^k\}_{i=1}^L \in \mathbb{R}^L. \tag{14}
\]

An MLP is implemented for modeling the mapping \( P^k \), which contains 3 layers. Each layer is composed by one fully connected layer followed by one ReLU function. See Fig. 1 for the diagram of the proposed hyper-parameter predictor \( P^k \).

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{mlp_diagram.png}
\caption{Diagram of the MLP for predicting hyper-parameter sequence \( \beta^k = \{\beta_i^k\}_{i=1}^L \).}
\end{figure}

Once the sequence \{\( \beta_i^k \)\}_{i=1}^L is predicted, we can update the estimate using the analytical solution (9). We denote such a solution by

\[
\mathcal{I}_{\text{inv}}^k : \{y, \{\beta_i^k\}_{i=1}^L, \{z_i^k\}_{i=1}^L\} \rightarrow x^k. \tag{15}
\]

In summary, the inversion block can be expressed as

\[
\mathcal{S}^k(\cdot; \theta^k_P) : \{y, \{z_i\}_{i=1}^L\} \rightarrow x^k, \tag{16}
\]

where \( \theta^k_P \) is the learnable weights.

Note that one can numerically solve (9) by conjugate gradient. To give the backward gradient for the backpropagation algorithm, let \( \ell : \mathbb{R}^{N_F} \rightarrow \mathbb{R} \) represent the cost function with respect to \( x^k \), then the derivatives of \( \ell(x^k) \) with respect to \( \{\beta_i^k\}_{i=1}^L, \{z_i^k\}_{i=1}^L \) can be correspondingly computed by:

\[
\begin{aligned}
\frac{\partial \ell(x^k)}{\partial x^k} &= \beta_i^k F_i (A^T A + \sum_{i=1}^L \beta_i^k F_i^T F_i)^{-1} \frac{\partial \ell(x^{k})}{\partial x^k}, \tag{17a} \\
\frac{\partial \ell(x^k)}{\partial \beta_i^k} &= (F_i^T z_i^k - F_i^T F_i x^k)^T (A^T A + \sum_{i=1}^L \beta_i^k F_i^T F_i)^{-1} \frac{\partial \ell(x^{k})}{\partial x^k}. \tag{17b}
\end{aligned}
\]
C. Denoising block with CNN-based adaptive prior

For the denoising block, following many existing methods, we also adopt a CNN to learn the function \( \Theta \) such that the estimate passed from the inversion block can be refined using learnable prior that is adaptive to the target image. It is noted that the measurements of an image in different high-pass channels are highly correlated. Independently running denoising in these high-pass channels is not a good practice as the inherent correlation is lost in such a process. Thus, the proposed CNN take the estimate \( x^k \) as the input and output a denoised version \( \tilde{x}^k \), which is then fed to \( L \) high-pass channels to have an estimate \( z_i^{k+1} \):

\[
z_i^{k+1} = \{ f_i \circ \tilde{x}^k \}_{i=1}^{L}, \quad 1 \leq i \leq L.
\]

Such a procedure reserves the correlations of an image among different high-pass channels.

Furthermore, instead of only taking the previous estimate \( x^k \) as the input, the CNN for denoising block takes all previous estimates \( \{ x^0, x^1, \ldots, x^k \} \) as the input by concatenation along channel dimension, which can alleviate the so-called vanishing gradient in training \([53]\). Another benefit of doing so is that the fusion of these previous estimates provide more information for the refinement, as these are different estimates of the truth with different types of artifacts. The final version of the mapping of the CNN-based denoising process can be expressed as

\[
D_{cnn}^k(\cdot, \theta_D^k) : \{ x^0, x^1, \ldots, x^k \} \rightarrow \tilde{x}^k,
\]

where \( \theta_D^k \) denotes the parameters of denoising NN, \( D_{cnn}^k \). The output of the whole denoising block is then

\[
z_i^{k+1} = f_i \circ D_{cnn}^k(\{ x^0, x^1, \ldots, x^k \}, \theta_D^k).
\]

For each stage of the iteration, we use 17-layer standard CNN with the structure Conv→BN→ReLU, except the first block and the last block. The BN layer is omitted for the first and last block. For all the Conv layers in the CNN, the kernel size is set as \( 3 \times 3 \). The channel size is set to 64. See Fig. 2 for the diagram of the CNN-based denoising block. In summary, the denoising block can be expressed as

\[
\mathcal{D}^k(\cdot, \theta_D^k) : \{ x^0, x^1, \ldots, x^k \} \rightarrow z^{k+1},
\]

where \( \theta_D^k \) is the set of learnable weights w.r.t. the CNN-based denoiser.

D. The overall architecture of the proposed method

The proposed deep learning method for LDCT contains totally \( K + 1 \) stages, denoted by \( \{ S_k \}_{k=0}^{K} \). Let \( \mathcal{D}^k \) and \( \mathcal{G}^k \) denote the inversion block and denoising block defined by (16) and (21). Then, each stage corresponds to one iteration in MBIR as follows.

\[
S^0(\cdot; \Theta^0) : \ y \xrightarrow{\mathcal{G}^0} x^0,
\]

\[
S^k(\cdot; \Theta^k) : \ y, \{ x^0 \}_{k=0}^{K} \xrightarrow{\mathcal{G}^k_{k}} \{ x^0 \}_{k=0}^{K-1} \xrightarrow{\mathcal{G}^k_{k+1}} x^k,
\]

for \( k = 1, 2, \ldots, K \), where \( \Theta^k := \{ \theta_D^{k-1}, \theta_D^k \} \) denotes the weights at the \( k \)-th stage, including the weights of \( D_{cnn}^{k-1} \), and \( \mathcal{P} \). In stage \( S^0 \), we initialize \( z^0 = 0 \) and \( \beta_0^0 = 0.005 \) for \( 1 \leq i \leq L \). It can be seen that when the measurement is fed to the proposed NN, it generates a sequence through the \( K + 1 \) stages:

\[
\{ x^0, x^1, \ldots, x^K \}.
\]

The final output of the whole NN is defined as \( x^* := x^K \). See Fig. 3 for the outline of the proposed NN, termed as AHP-Net, for LDCT reconstruction.

For the training of the proposed AHP-Net, we consider the training dataset \( \{ x_j, y_j \}_{j=1}^J \) with \( J \) training samples, where each \( (x_j, y_j) \) denotes one pair of normal dose image and low dose sinogram data. The loss function is defined as

\[
\mathcal{L}(\Theta) = \frac{1}{J} \sum_{j=1}^{J} \left( \| x_j^K - x_j \|_2^2 + \sum_{k=1}^{K-1} \mu_k \| x_j^k - x_j \|_2^2 \right),
\]

where \( \Theta := \{ \Theta^k \}_{k=0}^{K} \) is the whole set of NN parameters, and \( x^k = (S^k(\cdot; \Theta^k) \circ S^{k-1}(\cdot; \Theta^{k-1}) \circ \cdots \circ S^0(\cdot; \Theta^0)) (y_j) \) denotes the output of the AHP-Net at the \( k \)-th stage w.r.t. the input \( y_j \). The first term in (24) is for encouraging the output of the NN close to the truth. The second term is for ensuring the intermediate results not too far from the truth. The balancing parameters \( \{ \mu_k \}_{k=1}^{K} \) are set to \( \frac{1}{30} \) throughout all experiments. Our optimization was done via minimizing the loss function (24) using the Adam method. Once we finished the training of the model, an estimation of NN parameters \( \bar{\Theta} \) is obtained. For a low dose input data \( y \), the image can be reconstructed by fed it into the NN with parameters \( \bar{\Theta} \):

\[
x^* := \left( S^K(\cdot; \bar{\Theta}) \circ S^{K-1}(\cdot; \bar{\Theta}^{K-1}) \circ \cdots \circ S^0(\cdot; \bar{\Theta}^0) \right) (y).
\]

E. Implementation

The proposed AHP-Net is implemented with \( K = 3 \) stages. For the inversion block in each stage, the iteration number of CG is set to 5 and the hyperparameter \( \beta^k \) is initialized with the \( \ell_1 \) norm of high-pass filter \( f_k \). For the proposed NN method, training is performed with PyTorch [54] framework on a NVIDIA Titan GPU. As for parameters in the Adam optimizer, we set momentum parameter 0.9, mini batch size to be 4, and the learning rate to be \( 10^{-4} \). The model was trained for 50 epochs. The convolution weight of \( D_{cnn}^k \) for denoising were initialized with orthogonal matrices and the biases were initialized with zeros. All weights in \( \mathcal{P} \) were initialized to 1 and the biases to 0.

IV. EXPERIMENTS

In this section, we conduct a comprehensive experimental evaluation of the proposed AHP-Net, and compare it to other representative methods in LDCT image reconstruction.
Stage: $S_1$

\[ z^0 \xrightarrow{\beta^0} Z_{\text{inv}} \xrightarrow{D_{\text{cnn}}(\cdot, \theta^0_P)} x^0 \xrightarrow{\mathcal{F}_{\text{mr}}(\cdot, \theta^0_P)} z^1 \xrightarrow{\beta^1} Z_{\text{inv}} \xrightarrow{D_{\text{cnn}}(\cdot, \theta^1_P)} x^1 \]

Stage: $S_K$

\[ x^0, x^2 \ldots x^{K-1} \xrightarrow{\beta^K} Z_{\text{inv}} \xrightarrow{D_{\text{cnn}}(\cdot, \theta^K_D)} z^K \xrightarrow{\mathcal{F}_{\text{mr}}(\cdot, \theta^K_P)} x^K \]

Fig. 3: Diagram of the proposed AHP-Net for LDCT image reconstruction.

A. Dataset for evaluation

To evaluate the performance of the proposed method under different dose levels, we simulated low dose projection data from their normal-dose counterparts. The normal dose dataset included 6400 normal-dose prostate CT images of $256 \times 256$ pixels per image from 100 anonymized scans, where 80%, and 20% of the data is set for training and testing respectively. The LDCT projection data was simulated by adding Poisson noise onto the normal-dose projection data as Section II. The simulated geometry for projection includes 80 detector bins for each projection views evenly spanning a $360^\circ$ circular orbit, 512 detector bins for each projection with 1 mm pixel size, 100 cm source to detector distance, and 50.0 cm source to isocenter distance.

For the training dataset, 80 patients’ simulated low-dose projection and corresponding normal dose images were selected as the pairs of measurement and truth image. For every patient’s normal dose data, we simulated the low-dose measurement with uniform dose level. Then, the sinograms of different patients were obtained by taking logarithm on projection data $\mathbf{y}$. The remaining 20 scans were selected as testing dataset. For testing dataset, low dose measurement was simulated by the same way as training dataset. Throughout all experiments, 5120 pairs were included in training dataset, and 1280 pairs in testing dataset.

B. Training scheme

Two types of training schemes were conducted in this paper.

- **Different models for different dose levels.** In this training scheme, for each dose level, all deep learning methods trained one specific model using the measurements with the same dose level, and also tested it on the measurements with the same dose level. Totally 4 dose levels were evaluated: $I_i = 1 \times 10^0, 5 \times 10^4, 1 \times 10^4, 5 \times 10^3$.

- **One universal model for different dose levels.** In this training scheme, all deep learning methods trained a single model using the measurements with varying dose levels, and also tested on the measurements with varying dose levels. In this case, each measurement in training dataset was generated with its dose level randomly selected from $\{1 \times 10^0, 7.5 \times 10^4, 5 \times 10^4, 2.5 \times 10^4, 1 \times 10^4, 7.5 \times 10^3, 5 \times 10^3\}$, and each measurement in validation/testing dataset was generated by with its dose level $I_i = 1 \times 10^0, 5 \times 10^4, 1 \times 10^4, 5 \times 10^3$ respectively.

C. Comparison Method

The performance of the proposed methods was evaluated in comparison with classic FBP method with Ram-Lak filter, TV method, KSVD [55], BM3D [56], FBPConvNet [34], MoDL [44], Neumann Network [45], Projected Gradient Descent (PGD) [17] and Learned Primal-Dual (Learned-PD) [19].

- The TV method is a well-established classic method, which is often solved via the ADMM solver:

  \[
  x^{k+1} = \arg \min_x \frac{1}{2} \| A x - y \|^2 + \frac{\mu}{2} \| \nabla x - z^k + \frac{p^k}{\mu} \|^2, \\
  z^{k+1} = \arg \min_z \| z \|_1 + \frac{\mu}{2} \| z - (\nabla x^{k+1} + \frac{p^k}{\mu}) \|^2, \\
  p^{k+1} = p^k + \mu (\nabla x^{k+1} - z^{k+1}).
  \]

  The parameters $\lambda, \mu$ in the TV method are manually tuned up for optimal performance under different dose levels, which are given in Supplementary II.

- FBPConvNet [34] is one representative deep-learning-based method for CT reconstruction that use the deep NN as a post-process technique, where an NN with U-net based CNN is trained to directly denoise the image reconstructed by the FBP method.

- MoDL [44] is a deep learning method proposed for MRI reconstruction, which also can be used for LDCT reconstruction. MoDL unrolls the following iterative scheme:

  \[
  x^k = (A^T A + \lambda I)^{-1} (A^T y + \lambda x^{k-1}), \\
  x^{k+1} = D_{\text{cnn}}(x^k; \theta^k),
  \]

  where $D_{\text{cnn}}$ is a CNN-based denoiser and $\lambda$ is learned as an NN weight.

- Neumann-Net [45] is proposed for linear inverse problem in generic imaging, whose iterative scheme is as follows:

  \[
  x^K = \sum_{k=0}^{K-1} x^k + \eta (I - \eta A^T A) x^{K-1} - \eta D_{\text{cnn}}(x^{K-1}; \theta^{K-1}),
  \]

  where $D_{\text{cnn}}$ is a CNN-based mapping and $x^0 = A^T y$.

- PGD [17] is proposed for CT reconstruction with the following iterative scheme:

  \[
  x^{k+1} = (1 - \alpha_k) x^k + \alpha_k D_{\text{cnn}}(x^k - \gamma (A^T A x^k - A^T y)),
  \]
where $\mathcal{D}_{\text{cnn}}$ is a CNN-based mapping and $\alpha_k$ and $\gamma$ are parameters.

- Learned-PD [19] is the unrolling-based method proposed for CT reconstruction. By setting the initial value as $x^0$ and introducing the dual variable $h^0$, the iterative scheme based on PDHG algorithm is derived:

$$h^k = \mathcal{D}_{\text{cnn}}^{\theta_d}(h^{k-1} + \sigma A x^{k-1}, y),$$
$$x^k = \mathcal{D}_{\text{cnn}}^{\theta_p}(x^{k-1} - \tau A^T h^{k-1}),$$
$$x^k = x^{k-1} + \theta(x^k - h^{k-1}),$$

where $\theta_d$ and $\theta_p$ are the parameters of the dual proximal and the primal proximal respectively, $\sigma$ and $\tau$ are the step lengths, and $\theta$ is the overrelaxation parameter.

\section{Results}

In this section the proposed AHP-Net is evaluated on the simulated prostate CT data.

\subsection{Quantitative comparison on dataset}

Three metrics, peak signal to noise ratio (PSNR), root mean square error (RMSE) and structural similarity index measure (SSIM) [57], are used for quantitative evaluation of image quality. Recall that PSNR is defined as

$$\text{PSNR}(x, x^*) = -10 \log_{10} \left( \frac{\|x - x^*\|_2^2}{\max_i |x_i|^2} \right),$$

where $x^*$ denotes reconstructed image and $x$ denotes ground truth, i.e. normal dose image.

See Table I for the quantitative comparison of the results from different methods, where different models were trained for processing the data w.r.t. different dose levels. It can be seen that the performance of Neumann-Net is noticeably worse than the other. One possible reason might be that Neumann-Net need more iterations rather that only 3 stages. Three deep learning methods, FBPConvNet, MoDL and PGD, improved the reconstruction results. The proposed AHP-Net and Learned-PD are two best performers among all methods. In the case of relatively high dose level, $I_i = 1 \times 10^5, 5 \times 10^3$, the performances of these two methods are very close. In the case of relatively low dose level, $I_i = 1 \times 10^4, 5 \times 10^3$, the proposed AHP-Net outperformed Learned-PD by a noticeably margin, $1.7-2.5$ dB advantage in PSNR.

In short, the proposed AHP-Net showed its advantage over existing representative non-learning methods and deep learning methods, specially when the dose level is low. Also, the proposed AHP has its advantage in terms of practical usage, as it allows to train a single model to processing the data w.r.t. varying dose levels with very competitive performance. The outcome of quantitative comparison clearly indicates the importance of the setting of hyper-parameters to optimization-unrolling-based deep learning methods and the effectiveness of our solution to it.

\subsection{Visual comparison of some examples}

Due to space limitation, only the images reconstructed by different methods from one sample measurement are shown for visual comparison. The dose level of the measurement is $I_i = 1 \times 10^4$. The displayed window is set to $[-150, 150]$ HU for all figures. More results w.r.t. other dose levels can be found in Supplementary III for visual comparison.

Fig. 4 demonstrates the selected normal dose CT (NDCT) image and the zoomed region of interesting (ROI). For the image reconstructed by the non-learning methods and the NN models trained by deep learning methods, specifically for the dose level $I_i = 5 \times 10^3$, Fig. 5 shows the images reconstructed by different methods, and their zoomed-in images of boxes in Fig. 4 are displayed in Fig. 6. As shown by the cyan arrows, in comparison to he zoomed-in NDCT image, the results of FBPConvNet, MoDL, Neumann-Net, PGD and Learned-PD were more blurred than that of the proposed method. See Table III for quantitative comparison of the results shown in Fig. 5.

For the image reconstructed by the non-learning methods and the universal NN model trained by deep learning methods for varying dose levels, specifically for the dose level
different models are trained for different dose levels in deep learning methods.

<table>
<thead>
<tr>
<th>Dose level</th>
<th>Index</th>
<th>FBP</th>
<th>TV</th>
<th>KSVD</th>
<th>BM3D</th>
<th>FBPConvNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x 10^5</td>
<td>PSNR</td>
<td>37.15 ± 2.04</td>
<td>39.76 ± 2.18</td>
<td>36.55 ± 3.43</td>
<td>37.27 ± 2.15</td>
<td>38.14 ± 2.02</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>19.06 ± 1.10</td>
<td>14.18 ± 1.38</td>
<td>22.41 ± 2.09</td>
<td>20.07 ± 1.35</td>
<td>17.05 ± 1.48</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9330 ± 0.01</td>
<td>0.9685 ± 0.01</td>
<td>0.9341 ± 0.02</td>
<td>0.9297 ± 0.02</td>
<td>0.9788 ± 0.01</td>
</tr>
<tr>
<td>5 x 10^4</td>
<td>PSNR</td>
<td>32.24 ± 1.55</td>
<td>14.91 ± 1.57</td>
<td>22.09 ± 3.32</td>
<td>21.16 ± 2.08</td>
<td>17.13 ± 1.82</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>23.82 ± 2.22</td>
<td>23.40 ± 2.40</td>
<td>31.01 ± 2.16</td>
<td>31.46 ± 2.32</td>
<td>35.42 ± 2.12</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.8963 ± 0.02</td>
<td>0.9506 ± 0.01</td>
<td>0.9109 ± 0.02</td>
<td>0.9165 ± 0.02</td>
<td>0.9751 ± 0.01</td>
</tr>
<tr>
<td>1 x 10^4</td>
<td>PSNR</td>
<td>42.60 ± 5.74</td>
<td>25.27 ± 4.21</td>
<td>38.16 ± 5.03</td>
<td>34.91 ± 4.65</td>
<td>23.35 ± 2.42</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>65.63 ± 10.38</td>
<td>53.54 ± 5.86</td>
<td>54.06 ± 9.67</td>
<td>44.17 ± 8.74</td>
<td>29.51 ± 3.19</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.6883 ± 0.06</td>
<td>0.9191 ± 0.05</td>
<td>0.7304 ± 0.07</td>
<td>0.8046 ± 0.06</td>
<td>0.9616 ± 0.01</td>
</tr>
<tr>
<td>5 x 10^3</td>
<td>PSNR</td>
<td>26.17 ± 2.52</td>
<td>32.44 ± 2.95</td>
<td>27.87 ± 2.47</td>
<td>29.42 ± 2.07</td>
<td>33.39 ± 2.01</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>65.63 ± 10.38</td>
<td>53.54 ± 5.86</td>
<td>54.06 ± 9.67</td>
<td>44.17 ± 8.74</td>
<td>29.51 ± 3.19</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.5401 ± 0.07</td>
<td>0.7971 ± 0.08</td>
<td>0.6205 ± 0.09</td>
<td>0.7267 ± 0.08</td>
<td>0.9453 ± 0.01</td>
</tr>
</tbody>
</table>

TABLE II: Quantitative comparison (Mean±STD) of the results reconstructed by different methods on testing dataset, where one single model is trained for different dose levels in deep learning methods.

<table>
<thead>
<tr>
<th>Dose level</th>
<th>Index</th>
<th>NeumannNet</th>
<th>MoDL</th>
<th>PGD</th>
<th>Learned-PD</th>
<th>AHP-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x 10^5</td>
<td>PSNR</td>
<td>35.17 ± 2.14</td>
<td>38.87 ± 3.19</td>
<td>38.27 ± 2.03</td>
<td>41.16 ± 2.12</td>
<td>41.12 ± 2.69</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>24.20 ± 3.87</td>
<td>16.91 ± 4.68</td>
<td>16.76 ± 1.12</td>
<td>12.08 ± 1.39</td>
<td>12.41 ± 3.31</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9442 ± 0.02</td>
<td>0.9851 ± 0.02</td>
<td>0.9590 ± 0.01</td>
<td>0.9863 ± 0.00</td>
<td>0.9875 ± 0.01</td>
</tr>
<tr>
<td>5 x 10^4</td>
<td>PSNR</td>
<td>34.91 ± 2.11</td>
<td>38.12 ± 2.82</td>
<td>38.14 ± 2.07</td>
<td>39.35 ± 2.08</td>
<td>36.56 ± 3.06</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>25.19 ± 1.65</td>
<td>17.77 ± 5.65</td>
<td>17.03 ± 1.37</td>
<td>15.02 ± 1.37</td>
<td>14.98 ± 5.06</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.9354 ± 0.02</td>
<td>0.9783 ± 0.01</td>
<td>0.9531 ± 0.01</td>
<td>0.9808 ± 0.01</td>
<td>0.9819 ± 0.01</td>
</tr>
<tr>
<td>1 x 10^4</td>
<td>PSNR</td>
<td>33.27 ± 2.08</td>
<td>36.31 ± 2.64</td>
<td>35.53 ± 2.24</td>
<td>35.69 ± 2.10</td>
<td>38.20 ± 2.50</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>29.98 ± 3.77</td>
<td>21.54 ± 5.41</td>
<td>23.10 ± 2.84</td>
<td>22.64 ± 2.34</td>
<td>17.26 ± 4.60</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.8736 ± 0.03</td>
<td>0.9660 ± 0.01</td>
<td>0.9602 ± 0.01</td>
<td>0.9637 ± 0.01</td>
<td>0.9738 ± 0.01</td>
</tr>
<tr>
<td>5 x 10^3</td>
<td>PSNR</td>
<td>32.16 ± 2.07</td>
<td>35.02 ± 2.83</td>
<td>34.91 ± 2.15</td>
<td>35.16 ± 2.14</td>
<td>36.93 ± 2.42</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>34.07 ± 4.05</td>
<td>25.34 ± 13.45</td>
<td>24.73 ± 2.96</td>
<td>24.10 ± 2.74</td>
<td>19.84 ± 3.95</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.8017 ± 0.04</td>
<td>0.9615 ± 0.02</td>
<td>0.9527 ± 0.01</td>
<td>0.9594 ± 0.01</td>
<td>0.9673 ± 0.01</td>
</tr>
</tbody>
</table>

For the hyper-parameters set in MBIR, Fig. 11 shows the predicted hyper-parameters at different dose levels by the universal models trained for varying dose levels. It can be seen that the predicted hyper-parameter adapts well with inputs of different noise levels, where a lower-dose-level input requires larger hyper-parameter to incorporate more prior knowledge, and vice versa.

To investigate the optimal iterations K, we performed studies by training NN with various stages; see Fig. 12 for the quantitative results. It can be found that performance gains little after the 3-th stage. Besides, adding more iterations into the network will introduce more parameters and aggravate the computational burden of the network. Thus, we chose K = 3 in our experiments.

C. Ablation study

The ablation study conducted in this section focuses on how the following two parts impact the performance of image reconstruction: (1) the introduction of an MLP for adaptive prediction of the hyper-parameters and (2) the usages of 8 high-pass filters from B-spline framelet. Through the whole ablation study, all different versions of the AHP-Net are...
Fig. 5: Reconstruction results at dose level $I_i = 5 \times 10^3$ by the models trained under same dose level.

![Fig. 6: Zoom-in results of Fig. 5.](image)

trained by the same procedure on the same training/validation dataset and tested on the same testing dataset. Different dose levels are trained with different models. See Table IV for the results from the different versions.

- **no filter vs spline framelet filter bank.** To verify the effectiveness of the spline framelet filter bank of the proposed AHP-Net, the same NN but there is no filter was adopted. We set $z^k = x^k$, which the method is the same as MoDL. See the comparison of the column titled “No Filter” vs “AHP-Net” in Table IV. The results under different dose level showed the advantage of 8 filters over no filter, about 1.5–2.2 dB in PSNR, which indicates the joint benefits of framelet filter bank and adaptive hyper-parameters.

- **$\nabla$ vs spline framelet filter bank.** The ablation study was performed on AHP-Net and the same NN but whose 8 framelet filters are replaced by 2 filters of $\nabla$ ([1, -1], [1, -1]$^\top$) seen in TV regularization. See the comparison of the column titled as “Using $\nabla$” vs “AHP-Net” in Table IV. The results under different dose level showed the advantage of 8 filters over gradient operator $\nabla$, about 1.1–1.9 dB in PSNR. It is noted that in addition to performance gain by using the framelet filter bank, the performance benefit brought by MLP-based predictor is also fully exploited when using $\nabla$.

- **Learnable filters vs Spline framelet filter bank.** It would be interesting to see whether a learnable filter outperformed our spline frame filter; see the column titled as “Learnable filters” in Table IV. In the case of $I_i = 1 \times 10^5$, the performance gain of learning an MLP for adaptive prediction of hyper-parameters is 0.5dB in PSNR. In the case of lower dose levels, $I_i = 5 \times 10^4, 1 \times 10^4, 5 \times 10^3$, there is a significant improvement of the proposed method over “Learnable filters”, about 1.3–1.7dB in PSNR. One possible cause is that network training needs to solve a highly non-convex optimization problem with millions of network parameters. The filter bank plays an important role in the performance of the NN. However, when they are treated as one part of network parameters, they do not receive sufficient attention in the training.

- **Learning constant hyper-parameters as network weights vs Using an MLP for adaptive prediction of hyper-parameters.** The results from the proposed AHP-Net are compared to that from the same NN but whose hyper-parameters $\{\beta_k^j\}$ are treated as network parameters learned on training dataset; see the column titled as “Learnable HP” in Table IV. In the case of higher dose levels, $I_i = 1 \times 10^5, 5 \times 10^4$, the performance gain of learning an MLP for adaptive prediction of hyper-parameters is about 0.3–0.5 dB in PSNR. In the case of lower dose levels, $I_i = 1 \times 10^4, 5 \times 10^3$, there is a significant improvement of MLP-based adaptive prediction over learnable constant hyper-parameters, about 1.2–1.4 dB in PSNR.

Fig. 9 visualized a comparison result from different versions of the AHP-Net with dose level $I_i = 5 \times 10^4$, and Fig. 10 shows their zoomed-in regions of the boxes in Fig. 4. It can be seen that these figures correspond well with the results shown in Table IV; see more comparison in supplementary file.

D. Cross-validation

In this section, a $k$-fold cross-validation is conducted on FBPConvNet and the proposed AHP-Net to give a comprehen-
Fig. 7: Reconstruction results at dose level $I_i = 1 \times 10^4$ by the universal models trained for varying dose levels.

Fig. 8: Zoom-in results of Fig. 7.

Fig. 9: Reconstruction results at dose level $I_i = 5 \times 10^4$ by the models trained under same dose level. (a) No filter; (b) Using $\nabla$; (c) Learnable filters; (d) Learnable HP; (e) AHP-net.

Fig. 10: Zoom-in results of Fig. 9 corresponding to the red boxes in Fig. 4. (a) No Filter; (b) Using $\nabla$; (c) Learnable filters; (d) Learnable HP; (e) AHP-net.

Fig. 11: Predicted hyper-parameter at different dose levels by the universal models trained for varying dose levels.

Fig. 12: Quantitative studies of training AHP-Net with various stages $K$, in terms of PSNR and SSIM.

sive measure of its model’s performance throughout the whole dataset. More specifically, 100 scans are equally partitioned into 5 folders. Then, for each folder, the images associated with the scans from this fold are used for testing, and all others are used for training. One single model is trained for different noise levels. See Table V for the list of average quantitative metric values of the results from each test dataset. The mean of PSNR for the results from the proposed AHP-Net is about $2.4 - 3.5$ dB higher than that of FBPConvNet. However, the difference of the STD of PSNR is no more than $0.25$ dB.
TABLE IV: Quantitative comparison (Mean±STD) of the results reconstructed by the different versions of the AHP-Net for ablation study

<table>
<thead>
<tr>
<th>Dose level</th>
<th>Index</th>
<th>No Filter</th>
<th>Using ▽</th>
<th>Learnable filters</th>
<th>Learnable HP</th>
<th>AHP-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x 10^3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSNR</td>
<td>38.94 ± 0.44</td>
<td>39.26 ± 0.86</td>
<td>39.62 ± 1.50</td>
<td>39.86 ± 2.54</td>
<td>40.12 ± 2.60</td>
<td>41.12 ± 2.69</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9941 ± 0.01</td>
<td>0.9668 ± 0.01</td>
<td>0.9586 ± 0.01</td>
<td>0.9686 ± 0.02</td>
<td>0.9705 ± 0.02</td>
<td>0.9739 ± 0.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dose level</th>
<th>Index</th>
<th>No Filter</th>
<th>Using ▽</th>
<th>Learnable filters</th>
<th>Learnable HP</th>
<th>AHP-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 x 10^3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSNR</td>
<td>38.99 ± 0.70</td>
<td>39.48 ± 1.50</td>
<td>39.76 ± 2.00</td>
<td>40.04 ± 2.11</td>
<td>39.59 ± 2.08</td>
<td></td>
</tr>
<tr>
<td>RMSE</td>
<td>17.80 ± 0.79</td>
<td>17.25 ± 1.76</td>
<td>17.13 ± 4.43</td>
<td>16.14 ± 7.24</td>
<td>14.98 ± 7.06</td>
<td></td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9796 ± 0.01</td>
<td>0.9751 ± 0.02</td>
<td>0.9812 ± 0.01</td>
<td>0.9618 ± 0.02</td>
<td>0.9621 ± 0.01</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dose level</th>
<th>Index</th>
<th>No Filter</th>
<th>Using ▽</th>
<th>Learnable filters</th>
<th>Learnable HP</th>
<th>AHP-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 x 10^3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSNR</td>
<td>36.32 ± 0.62</td>
<td>36.74 ± 1.50</td>
<td>36.86 ± 2.49</td>
<td>36.78 ± 2.42</td>
<td>35.72 ± 2.45</td>
<td>35.19 ± 2.50</td>
</tr>
<tr>
<td>RMSE</td>
<td>21.65 ± 0.51</td>
<td>19.98 ± 4.51</td>
<td>20.97 ± 4.09</td>
<td>20.14 ± 5.09</td>
<td>17.26 ± 4.60</td>
<td></td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9662 ± 0.01</td>
<td>0.9601 ± 0.01</td>
<td>0.9743 ± 0.02</td>
<td>0.9755 ± 0.02</td>
<td>0.9758 ± 0.01</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dose level</th>
<th>Index</th>
<th>No Filter</th>
<th>Using ▽</th>
<th>Learnable filters</th>
<th>Learnable HP</th>
<th>AHP-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 x 10^3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSNR</td>
<td>35.07 ± 0.81</td>
<td>35.46 ± 3.41</td>
<td>35.41 ± 2.45</td>
<td>35.65 ± 2.49</td>
<td>36.35 ± 2.48</td>
<td>35.90 ± 2.46</td>
</tr>
<tr>
<td>RMSE</td>
<td>25.23 ± 1.45</td>
<td>26.21 ± 19.05</td>
<td>23.88 ± 4.45</td>
<td>23.08 ± 5.19</td>
<td>36.93 ± 2.42</td>
<td></td>
</tr>
<tr>
<td>SSIM</td>
<td>0.9665 ± 0.02</td>
<td>0.9513 ± 0.05</td>
<td>0.9624 ± 0.01</td>
<td>0.9729 ± 0.03</td>
<td>0.9673 ± 0.01</td>
<td></td>
</tr>
</tbody>
</table>

VI. DISCUSSION AND CONCLUSION

Similarly to many optimization-unnorlased-based deep learning methods, this paper also proposed a deep learning method for LDCT image reconstruction, the AHP-Net, that unrals the half-quadratic splitting scheme for MBIR. Each stage of the AHP-Net contains one inversion block and a denoising block, where the denoising block is built-on a CNN. The main difference of the proposed AHP-Net from other deep learning solutions lies the design of the inversion block.

In the proposed inversion block, we replaced the oftenued gradient operator ▽ by the filter banks with 8 high-pass filters from linear spline framelet transform, motivated by its success in l1-norm relating regularization in image recovery. Moreover, we proposed to pay special attention to the hyperparameters involved in the inversion block, and presented a MLP-based NN to predict hyper-parameters that adaptable to both dose level and image content.

The experiments showed the advantage of the proposed method over classic non-learning methods and some representavenous deep learning based methods for LDCT reconstruction. Also, another advantage is that the proposed method can only train a single model with competitive performance to process measurement data with varying dose levels.

As different dose levels implicitly refers to different SNRs. The proposed methods indeed allows to train a universal model with competitive performance to process the data with unknown SNRs, which can be valuable to practical applications in medical CT imaging. In practical medical CT scans, the dose level is not fixed and the SNR of the data is also different for different people under the same dose level, e.g. adult vs child. The adaptivity of the proposed method to different SNRs of data can be welcomed in practice.

There still exists some issues in the proposed method. One is that instead of reconstructing the 3D volume as a whole, the proposed method separately reconstructs image slices. The main reason we take such a reconstruction scheme is that the number of network parameters of the NN adopted in this paper will increase exponentially when constructing the whole 3D volume. The resulting memory usage in GPU will be too excessive to make the computation feasible under current available commodity GPUs.

In future work, we will study how to design compact deep NN that can directly reconstruct 3D volume in LDCT image reconstruction. Also, we will investigate the applications of the proposed method in some other medical image reconstruction problems, e.g. sparse-view CT reconstruction and image reconstruction from sparse samples in MRI. Further, we will study on the CT image reconstruction with spatially-variant noise distribution, where adaptive voxel-size hyper-parameter needs to be introduced.

REFERENCES


