MetaInv-Net: Meta Inversion Network for Sparse View CT Image Reconstruction

Haimiao Zhang, Baodong Liu, Hengyong Yu, Bin Dong

Abstract—X-ray Computed Tomography (CT) is widely used in clinical applications such as diagnosis and image-guided interventions. In this paper, we propose a new deep learning based model for CT image reconstruction with the backbone network architecture built by unrolling an iterative algorithm. However, unlike the existing strategy to include as many data-adaptive components in the unrolled dynamics model as possible, we find that it is enough to only learn the parts where traditional designs mostly rely on intuitions and experience. More specifically, we propose to learn an initializer for the conjugate gradient (CG) algorithm that involved in one of the subproblems of the backbone model. Other components, such as image priors and hyperparameters, are kept as the original design. This makes the proposed model very light-weighted. Since a hypernetwork is introduced to inference on the initialization of the CG module, it makes the proposed model a certain meta-learning model. Therefore, we shall call the proposed model the meta-inversion network (MetaInv-Net). The proposed MetaInv-Net has much less trainable parameters and superior image reconstruction performance than some state-of-the-art deep models in CT imaging. In simulated and real data experiments, MetaInv-Net performs very well and can be generalized beyond the training setting, i.e., to other scanning settings, noise levels, and noise types.

I. INTRODUCTION

COMPUTED tomography (CT) is one of the most important diagnostic imaging techniques. In clinical applications, sparse view CT is adopted to decrease the radiation dose and reduce the scanning time. This, however, inevitably leads to ill-posed inverse problems [1], [2] and gives rise to numerous new and exciting image reconstruction models and algorithms.

A. Trends of CT Models and Algorithms

Monochromatic energy CT imaging can be formulated as the following linear inverse problem

\[ Y = Pu + \eta, \]

where \( Y \) is the measured sinogram (projection data), \( P \) is an operator that models the imaging system. In the 2D continuum case, \( P \) is equivalent to the Radon transform for parallel beam imaging geometry. The parameter \( \eta \) is the measurement error or additive noise (e.g., zero mean White Gaussian noise). In incomplete data CT (e.g., sparse view CT, limited-angle CT and interior/exterior CT), the operator \( P \) is non-invertable. Thus, the above inverse problem is highly ill-posed.

Classical CT image reconstruction methods include the well-known Filtered Backprojection (FBP), algebraic reconstruction technique (ART), etc. [3]. However, these methods are relatively sensitive to noise \( \eta \). This is a manifestation of the ill-posed nature of the problem. To solve this issue, regularization based models have been widely adopted for the past few decades. Typical regularization based models take the following form [4]

\[ \min_u \frac{1}{2} \|Pu - Y\|_2^2 + \lambda R(u), \]  

where the first term is the discrepancy between the estimated \( u \) and measured data \( Y \), and \( R(u) \) is the regularization term that incorporates our prior knowledge on the image to be reconstructed. One popular class of regularization takes the form \( R(u) = g(Wu) \), with \( g \) being a sparsity promoting norm such as the \( \ell_1 \)-norm or \( \ell_0 \)-norm, and \( W \) is a linear/nonlinear transformation that maps a desired CT image \( u \) to its sparse coefficients. Successful examples include the total variation (TV) [5], nonlocal means [6], BM3D [7], WNNM [8], wavelets and wavelet frame models [9]–[11], K-SVD [12], data-driven (tight) frame [13], [14], low dimensional manifold method (LDMM) [15], etc.

Recent years, the rapid development of machine learning, especially deep learning, has lead to a paradigm shift of modeling and algorithmic design in computer vision and medical imaging [16]–[21]. Deep learning based models (or deep models for short) are able to leverage large image datasets to learn better image representations and produce better image reconstruction results than traditional methods [22]–[28]. To improve interpretability and reliability of deep models, deep learning methods and traditional iterative reconstruction methods were combined. This emerging new approach is often known as the unrolled dynamics (UD) approach, which firstly unrolls an iterative algorithm (e.g. an optimization algorithm associated to a regularization model) to form the backbone network architecture and then replaces parts of key ingredients, such as image representations, model parameters, inversion operators, proximal operators, etc., by deep neural networks. The UD approach started with the seminal work of [29] and has lead to many exciting developments in signal/image processing [30]–[35] and medical imaging [36]–[43].
In [29], the authors proposed to unroll the iterative soft-thresholding algorithm (ISTA) [44] to improve model efficiency for the sparse coding problem. Later, many other popular iterative algorithms were unrolled to generate deep models for various ill-posed inverse problems, such as the UD model from the primal-dual hybrid gradient (PDHG) [40], [45], alternating direction method of multipliers (ADMM) [36], [37], [43], [46], projected/proximal gradient descent [39], etc. Now, we review these UD methods in a collective fashion. Consider a general object functional as

$$F_\lambda(u, z; \beta) = D(u, Y) + \gamma C(u, z; \beta) + \lambda R_W(z),$$

where $D(u, Y)$ is the data discrepancy term, $C(u, z; \beta)$ is a constraint term that links the primal variable $u$, the auxiliary variable $z$ and the dual variable $\beta$, and $R_W(z)$ is the regularization term with variable $z$ and a certain sparsifying transformation $W$ which represents the image prior used by the model. Then, the aforementioned deep UD models use the following iterative algorithm (at iteration $k+1$) as their backbone network architecture

$$u^{k+1} = \arg \min_u F_\lambda(u, z^k; \beta^k),$$

$$z^{k+1} = \arg \min_z F_\lambda(u^{k+1}, z; \beta^k),$$

$$\beta^{k+1} = \beta^k + \gamma(Wu^{k+1} - z^{k+1}).$$

(2)

In these deep UD models, the $u$-subproblem is the image reconstruction layer, the $z$-subproblem is the denoising layer, and $\lambda$ and $\gamma$ are either hyperparameters or trainable parameters. Different deep UD model has its own unique design on the $u$- and $z$-layer, and makes its own choice on handcrafting and learning in every module of the model. Next, we shall provide some details on the design of a few deep UD models.

A summary is given in Table I.

In DUBLID [35] and ADMM-Net [36], $u$-layer was a handcrafted analytic inversion formula with trainable parameters. For $z$-layer, a soft-threshold operation was implemented with learnable parameters. The sparsifying transformation $W$ was learned from the dataset in an end-to-end manner.

In DnCNN [32] and ADMM-CSNet [43], the $u$-layer was solved by a handcrafted analytic formula with trainable parameters. For $z$-layer, a convolutional neural network was implemented along with the learnable sparsifying transformation $W$ as an image denoiser.

In ISTA-Net [33] and DPDNN [34], $u$-layer was implemented by a one-step gradient descent with trainable step size. For $z$-layer, convolutional neural network was implemented along with the learnable sparsifying transformation $W$ as an image denoiser.

In PD-Net [40], the authors replaced the proximal operators in both $u$-layer and $z$-layer by multi-layer convolutional neural networks. Model parameters and sparsifying transformation $W$ were learned from the dataset.

**B. Motivation**

The general strategy of designing deep UD models is to first select an appropriate iterative algorithm as the backbone network architecture and then decide on which components of the UD need to be data-aware. As arbitrary as the design of deep UD models may sound, our general rule of thumb is that it is unnecessary to learn the knowledge or principles that are certain to play an important role; instead, we should focus on learning the parts of the unrolled dynamics where traditional designs mostly rely on oversimplification, inaccurate intuitions or trails-and-errors. This is partially supported by [48], [49], where the authors showed that we do not need to rely too much on learning to achieve an asymptotic linear convergence rate of the learned ISTA.

In this paper, we propose a new deep UD model by unrolling the half-quadratic splitting (HQS) algorithm [50] to form the backbone network architecture. The $u$-subproblem (i.e., the image reconstruction layer) of HQS is a least square problem. We solve it by the conjugate gradient (CG) algorithm. We shall call the HQS algorithm with the $u$-subproblem solved by CG as HQS-CG. At iteration $k$, the common practice is to initialize the CG algorithm (with $L$ iterations) using $u^{k-1,L}$, the approximated solution $u$ from the iteration $k-1$. Ideally, if $u^{k-1,L}$ gets close to the fixed point of the HQS-CG algorithm quickly, the entire algorithm will terminate quickly and output a high-quality CT image. However, it is hard to design a good initialization for the CG step, since it can be as difficult as finding the solution of the inverse problem itself. In this work, we adopt a neural network to predict the initial value at iteration $k$ as

$$u^{k,0} = \Theta(u^{k-1,L}, \Theta^k),$$

where $\Theta^k$ collects the trainable parameters. For the $z$-subproblem (i.e., the image denoising layer), we observe that the sparsifying transformation $W$ can be fixed. In the numerical experiment, we choose $W$ as the highpass components of the piecewise linear tight wavelet frame transform.

Unlike most of the existing deep CT image reconstruction models, our proposed model is rather light-weighted. We only use a trainable CNN to infer an initialization for CG algorithm while keeping all the other components of the HQS-CG algorithm unchanged. Other than the initialization for each CG step, the hyperparameters $\lambda$ and $\gamma$ are also important and difficult to tune in practice. However, as shown in our numerical experiments that using the neural network to approximate $\lambda$ and $\gamma$ along with $u^{k,0}$ does not bring noticeable overall benefit, while it inevitably introduces more trainable parameters to the proposed model. Therefore, we shall manually fix $\lambda$ and $\gamma$ in all of our experiments without tuning them for each test case.

We would argue that the proposed deep UD model based on HQS-CG can be interpreted as a certain meta-learning model. Meta-learning is a branch of machine learning that exploits intrinsic common knowledge between different tasks to effectively solve new tasks. Meta-learning can be applied for hyperparameter initialization [51], [52], multi-task learning [53], weight pruning [54], neural architecture search [55], [56], Bayesian neural networks [57], and hyperparameter optimization [58]. A specific form of meta-learning models adopts a hypernetwork to predict hyperparameters or weights in the backbone neural network [59]. Figure 1 shows a common
TABLE I
THE LEARNED ISSUE OF DIFFERENT UD MODELS. THE EXISTENCE OF LEARNABLE PARAMETERS IS INDICATED BY YES OR NO WITHIN BRACKET.

| Models                  | Learned Issue | $u$-Layer (Param.) | $g$-Layer (Param.) | Image Flow (W) |
|-------------------------|---------------|---------------------|---------------------|----------------|
| DUBLID [40]             | No            | Yes                 | Yes                 | Yes            |
| ISTA-Net [33], DPDNN [34] | Yes          | Yes                 | Yes                 | Yes            |
| ADMM-CSNet [43]         | Yes           | Yes                 | Yes                 | Yes            |
| DnCNN [32]              | Yes           | Yes                 | Yes                 | Yes            |
| MetaInv-Net (MetaInv-Net) | Yes          | Yes                 | Yes                 | Yes            |

architecture of the hypernetwork based deep model. If we write the image reconstruction model as

$$T(Y, \theta) \rightarrow u,$$

where $\theta$ represents the model (or algorithm) parameters, $Y$ is the given measured data, and $u$ is the reconstructed image. In Hypernetwork, the model parameters $\theta$ can be predicted by another neural network $H_{\phi}(\cdot)$ rather than being learned directly from a dataset. Hence, the image reconstruction model can be rewritten as

$$T(Y, H_{\phi}(Y)) \rightarrow u,$$

with $\phi$ collects the weight of Hypernetwork and is learned by end-to-end training on a dataset. Hypernetwork helps to compress the trainable parameters and simplify the training process while maintaining its meta-learning property.

![Fig. 1. A standard hypernetwork.](image)

For our proposed neural network model, the $u$-subproblem that solved by the CG algorithm is initialized by a predicted value from a CNN with the residual structure taking the current state (i.e., the approximation of image $u^{k-1, L}$) as input. This CNN is a Hypernetwork of the full model, which is illustrated in Fig. 1. We denote our model as Meta-Inversion Network (MetaInv-Net) in the rest of the paper. Note that there is not any trainable parameters in the backbone network. All trainable parameters are in the hypernetwork.

![Fig. 2. MetaInv-Net.](image)

C. Our Contribution

The contributions of this work are summarized as follows:

- Lighter model and better image quality. We propose a new deep model, MetaInv-Net, which has much less trainable parameters but with higher reconstruction qualities than many state-of-the-art deep models for CT image reconstruction.
- Faster to training. The light-weighted MetaInv-Net can be trained with much less data while still achieves better performance. Therefore, it is more data set independent during the training phase.
- Better generalization. The proposed MetaInv-Net generalizes well beyond the training setting (i.e., to other scanning settings, noise levels, and noise types).

The remaining part of this paper is organized as follows. In Section III we present details on the design of the MetaInv-Net. Numerical experiments are presented in Section III. Finally, the remarks on conclusions and future works are presented in Section IV.

II. METHODS

In this section, we introduce full details on the proposed MetaInv-Net. We first introduce the backbone model used by MetaInv-Net, followed by a detailed description of the design of the Hypernetwork and the loss function.

A. Half Quadratic Splitting (HQS)-CG Algorithm

The backbone model is the unrolled HQS algorithm [50] solving the following optimization problem

$$\min_{u, z} \frac{1}{2} \|Pu - Y\|^2 + \lambda \|z\|_1 + \frac{1}{2} \sum_{i=1}^{M} \gamma_i \|W_i u - z_i\|^2; \quad (3)$$

where $W = (W_1, \ldots, W_M)$ is a $M$ channel operator, $z = (z_1, \ldots, z_M)$, $\lambda > 0$ and $\gamma = (\gamma_1, \ldots, \gamma_M)$ with $\gamma_i > 0, i = 1, \ldots, M$. The operator $W$ is chosen as the highpass components of the piecewise linear tight wavelet frame transform [11], [60].

Now, the variables in optimization problem (3) are splitted into two blocks that can be updated alternatively

$$u^{k+1} = \arg \min_{u} \|Pu - Y\|^2 + \sum_{i=1}^{M} \gamma_i \|W_i u - z_i\|^2,$$

$$z^{k+1} = \arg \min_{z} \lambda \|z\|_1 + \frac{1}{2} \sum_{i=1}^{M} \gamma_i \|W_i u^{k+1} - z_i\|^2; \quad (4)$$

with proper initialization $u_0$ and $z_0$. Solution to each of the two subproblem takes the form

$$u^{k+1} = \left( P^T P + \sum_{i=1}^{M} \gamma_i W_i^T W_i \right)^{-1} \left[ P^T Y + \sum_{i=1}^{M} \gamma_i W_i^T z_i \right],$$

$$z^{k+1} = T_{\lambda/\gamma}(W u^{k+1}). \quad (5)$$
The linear system in the $u$-subproblem of (5) is solved approximately by the CG algorithm. Rewrite this large scale linear system as

$$A u = B^k,$$

where $A = P^T P + \sum_{i=1}^M \gamma_i W_i^T W_i$ and $B^k = P^T y + \sum_{i=1}^M \gamma_i W_i^T z_i^k$. Commonly, no hyperparameters are necessarily being provided to the CG algorithm except the stopping criteria ($L$ iterations). For the $(k+1)$-th out-loop of HGS-CG algorithm, the initial variable $u^{k+1,0}$ for CG algorithm is usually set to 0 or the former output $u^{k,L}$.

The proximal operator $T_\lambda(\cdot)$ is the anisotropic soft-thresholding operator defined by $T_\lambda(x) = \text{sign}(x) \max\{|x| - \lambda, 0\}$. Summary of the HGS-CG method is shown in Algorithm 1.

\begin{algorithm}[H]
\caption{HGS-CG algorithm}
\begin{algorithmic}[1]
\STATE Initialization: $u^{0,L} = u_{\text{FBP}}, z^0 = W u^{0,L}$
\FOR{$k = 0 : K - 1$}
\STATE $u^{k+1,0} = u^{k,L}$ or 0
\STATE $u^{k+1,L} = \text{CG}(u^{k+1,0}, z^k)$
\STATE $z^{k+1} = T_{\lambda/\gamma}(W u^{k+1,L})$
\ENDFOR
\STATE Output: $u^{K,L}$
\end{algorithmic}
\end{algorithm}

Note that, the hyperparameters $\lambda$ and $\gamma$ in Algorithm 1 can be manually selected with extensive trials-and-errors. To make the algorithm more practically applicable, data-adaptive hyperparameter estimation methods are often adopted, which include the discrepancy principle [61], [62], cross validation, L-curve [61] and the quasi-optimality criterion [61]. Recently, deep learning approaches are able to determine pixel-wise, data-adaptive and task-dependent hyperparameters and operators in the model through end-to-end training on the given dataset [29], [36], [40], [47]. Recent development of deep learning based hyperparameter optimization includes hypernetwork based methods [54], [56], [58] and gradient descent-based approaches [63]–[67].

Other components in Algorithm 1 such as the sparsifying transform $W$ and the proximal operator $T_\lambda(\cdot)$ can also be adaptively learned from the dataset as what has been done by the earlier works listed in Table I. In this work, however, we fix the hyperparameters, $\lambda, \gamma$ and $W$, throughout our experiments. We observe that these hyperparameters are less important than the initial values for the CG algorithm in the backbone model. Our numerical experiments confirm that although using a hypernetwork to estimate all the aforementioned hyperparameters may lead to better results for some cases (but not all as shown in Figure 12), it also leads to more trainable parameters for the model and out-weights its benefit in general.

B. Meta-Inversion Network (MetaInv-Net)

The backbone architecture of MetaInv-Net is built by unrolling the HGS-CG Algorithm (Algorithm 1) with $K$ iterations forming a $K$-layer feed-forward network.

1) Image Reconstruction Module: In this paper, a CNN-based initializer is adopted to providing a better initialization variable $u^{k+1,0}$ for CG algorithm. The CG initializer with trainable parameters $\Theta^{k+1}$ and past reconstruction $u^{k,L}$ as input is built as

$$u^{k+1,0} = u^{k,L} + \text{CNN}(u^{k,L}; \Theta^{k+1}).$$

Here, the skip connection helps to avoid gradient vanishing/exploding during training. It also admits a dynamic system interpretation [68]–[70]. The detailed architecture of the CNN in (6) is composed of $S$ convolutional (Conv) layer and Parametric ReLU (PReLU) activation. Fig. 3 shows the architecture of this module.

![Fig. 3. CG initializer module.](image)

2) Denoiser Module: There is no trainable parameter in this module. We choose the operator $W$ as the tight wavelet frame transform. The proximal operator $T_{\lambda}(\cdot)$ is a soft-thresholding. The empirical result shows that this handcrafted image denoiser does not have explicitly influence on the quality of final reconstruction. In conclusion, a handcrafted denoiser has some benefits 1) the handcrafted $T_{\lambda}(\cdot)$ has no training parameters, thus reduces the total number of parameters in MetaInv-Net, 2) $W$ is dataset independent and thus should have better generalization property. Fig. 4 shows one block of the MetaInv-Net. Summary of MetaInv-Net is presented in Algorithm 2 and its architecture is presented in Fig. 5.

![Fig. 4. MetaInv-Net building block.](image)

![Fig. 5. MetaInv-Net.](image)

\begin{algorithm}[H]
\caption{MetaInv-Net}
\begin{algorithmic}[1]
\STATE Initialization: $u^{0,L} = u_{\text{FBP}}, z^0 = W u^{0,L}$
\FOR{$k = 0 : K - 1$}
\STATE $u^{k+1,0} = u^{k,L} + \text{CNN}(u^{k,L}; \Theta^{k+1})$
\STATE $u^{k+1,L} = \text{CG}(u^{k+1,0}, z^k)$
\STATE $z^{k+1} = T_{\lambda/\gamma}(W u^{k+1,L})$
\ENDFOR
\STATE Output: $u^{K,L}$
\end{algorithmic}
\end{algorithm}
3) Loss Function: Loss function in MetaInv-Net is defined by
\[ \mathcal{L}(u^{1:L}, \ldots, u^{K:L}, u^{gt}) = \sum_{k=1}^{K} (\mu_1 L_2(u^{k:L}, u^{gt}) + \mu_2 L_{asim}(u^{k:L}, u^{gt})) , \]
where \( L_2(x, y) \) represents the \( \ell_2 \)-norm of \( x - y \) and \( L_{asim} \) is the SSIM loss. Similar to the value function design in reinforcement learning, constants \( \mu_1 \) and \( \mu_2 \) can be viewed as discount factors for \( \ell_2 \) loss \( L_2 \) and SSIM loss \( L_{asim} \), respectively.

III. Numerical Experiments

In this section, we validate the performance of MetaInv-Net on sparse view CT image reconstruction beyond the training setting (e.g., to different scanning settings, noise levels, and noise types).

A. Comparison Methods

The FBP is commonly used as a benchmark in CT image reconstruction. The HQS-CG Algorithm (Algorithm 1) is an iterative reconstruction method with the stopping criterion set to \( K = 50 \). Two state-of-the-art deep UD models, i.e., PD-Net [40] and JSR-Net [41] are adopted in our comparison with MetaInv-Net. As one can see both visually and quantitatively, the proposed MetaInv-Net has comparable performance as HQS-CG, and it outperforms the compared deep learning models, especially for the cases when fewer scanning views are used. Furthermore, error maps from MetaInv-Net contain less structures than the compared approaches at different scanning settings.

B. Training Setting

We adopt five patients’ data from the “2016 NIH-AAPM-Mayo Clinic Low Dose CT Grand Challenge” as a training set. The contained 1683 slices of high-resolution images are resized to \( 512 \times 512 \) pixels and used as the ground truth. An additional one patient’s data from the same data set containing 366 slices image is used as the validation set. The scanning geometry is a fan beam X-ray source with 180 scanning views equally distributed around \( 360^\circ \) and 800 detector elements. The simulated sinogram is contaminated by Gaussian noise with a standard deviation \( (\sigma=0.05) \). To have a fair comparison, we set the hyperparameters of each deep model properly, so that they contain a similar number of training parameters. CG initializer of MetaInv-Net has depth \( S = 6 \) and channel width 8. Table II summarizes the details of training setting. Unless specifically explained, all the deep models are only trained on sparse view CT reconstruction with the fixed setting in Table II.

For simplicity, we replace the original parameter representation \( \lambda/\gamma \) by \( \lambda \). In each layer of MetaInv-Net, \( \lambda_k \) and \( \gamma_k \) are set by \( \lambda_k = \lambda_{k-1} - \delta_k \) and \( \gamma_k = \gamma_{k-1} - \delta_k \). We empirically set \( \lambda_0 = 0.005 \), \( \delta_0 = 0.0007 \), \( \gamma_0 = 0.01 \) and \( \delta_0 = 0.003 \). The discount factors \( \mu_1 \) and \( \mu_2 \) in the definition of loss function are set to 1.1 and 1, respectively.

The training is conducted on PyTorch 1.3.1 backend with an NVIDIA Titan Xp GPU with memory 10.75G. The ADAM optimizer with an initial learning rate, \( 10^{-3} \), adopted in the training phase.

C. Test Setting

We test the well-trained deep models on different scanning settings, noise levels, noise types, and data sets. We choose eight slices of Abdoment CT images as a test set, which comes from the same distribution of training set but not used during training. This test set is denoted as AAPM-8. Another test set is one patient’s Pancreas CT containing 210 slices. This data set is denoted as Pancreas-210. It has a different distribution to the training set. Quantitative results are measured by the SSIM index and PSNR.

D. Sparse View CT

1) Gaussian Noise Case: In this experiment, we test MetaInv-Net on AAPM-8 and Pancreas-210. Table III reports the quantitative results (averaged on the test set) of compared approaches. Note that these results are the generalization performance at different scanning settings, noise levels, and test sets. The contents in the bracket indicate the standard deviation. We observe that the MetaInv-Net is superior to HQS-CG when the test settings are similar to the training setting. For more challenging tasks (i.e., a smaller number of scanning views and higher noise levels), MetaInv-Net has better generalization performance than PD-Net and JSR-Net.
obviously outperforms traditional iterative algorithms by a big margin. In contrast, the proposed MetaInv-Net generalizes as well as the none learning based method HQS-CG, which demonstrates the flexibility and robustness of MetaInv-Net.

Fig. 7. Sparse-view CT image reconstruction (GT1) with Gaussian noise \( \sigma = 0.2 \) and different number of scanning views (e.g., 60, 90, 120 and 180). The gray scale window is \([0, 1]\). The rows from top to bottom correspond to numbers of sparse views 60, 90, 120, and 180.

2) Poisson Noise Case: In this experiment, the simulated sinogram is contaminated by Poisson noise. The noise level is controlled by the emitted photon intensity from the X-ray source based on the statistical property of Poisson distribution. Here, we assume the photon intensity, \( I_0 \), to be a constant. Then, Poisson noisy projection data is simulated by

\[
Y = -\log\{\text{Poisson}(I_0e^{-Pu})/I_0\}.
\]

Each method is tested on the Pancreas-210 data set. As the test set size is large, it is hard to fine-tune the HQS-CG Algorithm for each phantom. Thus, we only compare MetaInv-Net to PD-Net and JSR-Net. Fig. 11 shows the quantitative measure of SSIM/PSNR versus different Poisson noise level. When the number of scanning views is increased, the performance of...
Table V: Quantitative measure of the reconstructed image (GT2) in Fig. 9.

| Models   | Noise level | SSIM | PSNR | SSIM | PSNR |
|----------|-------------|------|------|------|------|
| FBP      | 0.8267      | 37.3130 | 0.9704 | 35.1631 | 0.7630 |
| HQS-CG   | 0.8322      | 37.3130 | 0.9704 | 35.1631 | 0.7630 |
| PD-Net   | 0.8031      | 37.4115 | 0.9704 | 35.1631 | 0.7630 |
| JSR-Net  | 0.8101      | 37.6215 | 0.9704 | 35.1631 | 0.7630 |
| MetaInv-Net | 0.9343      | 39.4073 | 0.9704 | 35.1631 | 0.7630 |

![Fig. 9. Sparse-view CT image reconstruction (GT2) with Gaussian noise \( \sigma = 0.2 \) and different number of scanning views (e.g., 60, 90, 120, and 180). Gray scale window is \([0.3, 0.7]\). The rows from top to bottom correspond to numbers of sparse views 60, 90, 120, and 180.](image)

![Fig. 10. Error map of the reconstructed image GT2. The gray scale window is \([-0.08, 0.08]\). The rows from top to bottom correspond to numbers of sparse views 60, 90, 120, and 180.](image)

MetaInv-Net becomes closer but still better than PD-Net and JSR-Net. PD-Net is even worse than FBP when the noise level is higher.

3) Computation Time: We compute the per-image time cost at the inference phase with the image size 512 \( \times \) 512 and sinogram size 180 \( \times \) 800. The time cost for HQS-CG, PD-Net, JSR-Net, and MetaInv-Net is 7.0460, 1.3301, 1.9455, and 1.0500 seconds(s), respectively.

E. More Experiments

In this section, we do the ablation study for MetaInv-Net. There are several variants of MetaInv-Net, which learned: 1) \( \lambda \) and \( \gamma \), 2) \( W \), 3) CG-Initializer (CG-Init), \( \lambda \) and \( \gamma \), 4) CG-Init and \( W \), and 5) CG-Init, \( \lambda \), \( \gamma \), and \( W \), from data. All these models are trained in the same setting as the previous section. The well-trained models are tested on the AAPM-8 with different Gaussian noise levels and a various number of scanning views.

1) Learning Different Components in MetaInv-Net: The hyperparameters \( \lambda \) and \( \gamma \) in MetaInv-Net can be predicted by fully connected neural network (FCN) with the input adopted from each layer’s state variables. For \( \lambda \) in each layer, the \( \ell_2 \)-norm of highpass components of wavelet frame coefficients \( W u \) is computed channel-wisely and used as the input of an FCN. For \( \gamma \) in each layer, the \( \ell_2 \)-norm of highpass components of \( W u \), \( W \top u \), \( W \top W z \), and \( P u \), \( Y \) are concatenated and used as the input of a FCN. Both the predicted \( \lambda \) and \( \gamma \) have equal dimension as the number of highpass channels of the wavelet frame transform. The learning of the image representation \( W \) is implemented by adopting the standard error backpropagation during the training phase.

Fig. [12] presents the curves of SSIM/PSNR versus different Poisson noise level. The test set is Pancreas-210.
Fig. 12. MetaInv-Net is compared to its variants at different Gaussian noise level.

Fig. 13. Deep models trained with different training set size.

In electrocardiographic (ECG)-gated cardiac multiphase 4D imaging, the reconstruction of each phase is a sparse view CT imaging problem. The current setting is a realistic simulation with ground truth. In this section, the proposed MetaInv-Net is tested on real data, which is scanned with the same protocol to [72]. The original helical geometry is converted to a fan-beam geometry. There are a total of 2200 scanning views equally distributed 360°. The number of detector elements is 888, and its length is 95.93 cm. The reconstructed image has 800 × 800 pixels around 24.3 × 24.3 cm². The distance of source to rotation center is 53.85 cm and detector to rotation center is 49.83 cm.

Each compared method adopts the complete measured data to reconstruct its reference (Full View in Fig. 14). The sinogram is downsampled to 400, 800, and 1600 views to simulate the sparse view CT imaging. Fig. 14 shows the reconstructed images at different sparse view settings. When the number of sparse views is smaller, MetaInv-Net produces an image with better visual quality than FBP, PD-Net, and JSR-Net. HQS-CG produces the image with much fewer artifacts than MetaInv-Net. However, the details and edges in HQS-CG reconstruction are smoother than MetaInv-Net.

2) Different Training Set Size: In this experiment, we retrain the deep models with the same training setting (except training epochs=20) at different data set sizes (e.g., 1% and 5% of the full training set). For noise level σ = 0.2, we test the deep models on AAPM-8 at different numbers of scanning views. For each training setting, the curve of SSIM/PSNR versus the number of views is established in Fig. 13. MetaInv-Net is much less demanding in data quantity than PD-Net and JSR-Net.

F. Real Data Study

In electrocardiographic (ECG)-gated cardiac multiphase 4D imaging, the reconstruction of each phase is a sparse view CT imaging problem. The current setting is a realistic simulation with ground truth. In this section, the proposed MetaInv-Net is tested on real data, which is scanned with the same protocol to [72]. The original helical geometry is converted to a fan-beam geometry. There are a total of 2200 scanning views equally distributed 360°. The number of detector elements is 888, and its length is 95.93 cm. The reconstructed image has 800 × 800 pixels around 24.3 × 24.3 cm². The distance of source to rotation center is 53.85 cm and detector to rotation center is 49.83 cm.

Each compared method adopts the complete measured data to reconstruct its reference (Full View in Fig. 14). The sinogram is downsampled to 400, 800, and 1600 views to simulate the sparse view CT imaging. Fig. 14 shows the reconstructed images at different sparse view settings. When the number of sparse views is smaller, MetaInv-Net produces an image with better visual quality than FBP, PD-Net, and JSR-Net. HQS-CG produces the image with much fewer artifacts than MetaInv-Net. However, the details and edges in HQS-CG reconstruction are smoother than MetaInv-Net.

2) Different Training Set Size: In this experiment, we retrain the deep models with the same training setting (except training epochs=20) at different data set sizes (e.g., 1% and 5% of the full training set). For noise level σ = 0.2, we test the deep models on AAPM-8 at different numbers of scanning views. For each training setting, the curve of SSIM/PSNR versus the number of views is established in Fig. 13. MetaInv-Net is much less demanding in data quantity than PD-Net and JSR-Net.

F. Real Data Study

In electrocardiographic (ECG)-gated cardiac multiphase 4D imaging, the reconstruction of each phase is a sparse view CT imaging problem. The current setting is a realistic simulation with ground truth. In this section, the proposed MetaInv-Net is tested on real data, which is scanned with the same protocol to [72]. The original helical geometry is converted to a fan-beam geometry. There are a total of 2200 scanning views equally distributed 360°. The number of detector elements is 888, and its length is 95.93 cm. The reconstructed image has 800 × 800 pixels around 24.3 × 24.3 cm². The distance of source to rotation center is 53.85 cm and detector to rotation center is 49.83 cm.

Each compared method adopts the complete measured data to reconstruct its reference (Full View in Fig. 14). The sinogram is downsampled to 400, 800, and 1600 views to simulate the sparse view CT imaging. Fig. 14 shows the reconstructed images at different sparse view settings. When the number of sparse views is smaller, MetaInv-Net produces an image with better visual quality than FBP, PD-Net, and JSR-Net. HQS-CG produces the image with much fewer artifacts than MetaInv-Net. However, the details and edges in HQS-CG reconstruction are smoother than MetaInv-Net.

2) Different Training Set Size: In this experiment, we retrain the deep models with the same training setting (except training epochs=20) at different data set sizes (e.g., 1% and 5% of the full training set). For noise level σ = 0.2, we test the deep models on AAPM-8 at different numbers of scanning views. For each training setting, the curve of SSIM/PSNR versus the number of views is established in Fig. 13. MetaInv-Net is much less demanding in data quantity than PD-Net and JSR-Net.

F. Real Data Study

In electrocardiographic (ECG)-gated cardiac multiphase 4D imaging, the reconstruction of each phase is a sparse view CT imaging problem. The current setting is a realistic simulation with ground truth. In this section, the proposed MetaInv-Net is tested on real data, which is scanned with the same protocol to [72]. The original helical geometry is converted to a fan-beam geometry. There are a total of 2200 scanning views equally distributed 360°. The number of detector elements is 888, and its length is 95.93 cm. The reconstructed image has 800 × 800 pixels around 24.3 × 24.3 cm². The distance of source to rotation center is 53.85 cm and detector to rotation center is 49.83 cm.

Each compared method adopts the complete measured data to reconstruct its reference (Full View in Fig. 14). The sinogram is downsampled to 400, 800, and 1600 views to simulate the sparse view CT imaging. Fig. 14 shows the reconstructed images at different sparse view settings. When the number of sparse views is smaller, MetaInv-Net produces an image with better visual quality than FBP, PD-Net, and JSR-Net. HQS-CG produces the image with much fewer artifacts than MetaInv-Net. However, the details and edges in HQS-CG reconstruction are smoother than MetaInv-Net.

2) Different Training Set Size: In this experiment, we retrain the deep models with the same training setting (except training epochs=20) at different data set sizes (e.g., 1% and 5% of the full training set). For noise level σ = 0.2, we test the deep models on AAPM-8 at different numbers of scanning views. For each training setting, the curve of SSIM/PSNR versus the number of views is established in Fig. 13. MetaInv-Net is much less demanding in data quantity than PD-Net and JSR-Net.

F. Real Data Study

In electrocardiographic (ECG)-gated cardiac multiphase 4D imaging, the reconstruction of each phase is a sparse view CT imaging problem. The current setting is a realistic simulation with ground truth. In this section, the proposed MetaInv-Net is tested on real data, which is scanned with the same protocol to [72]. The original helical geometry is converted to a fan-beam geometry. There are a total of 2200 scanning views equally distributed 360°. The number of detector elements is 888, and its length is 95.93 cm. The reconstructed image has 800 × 800 pixels around 24.3 × 24.3 cm². The distance of source to rotation center is 53.85 cm and detector to rotation center is 49.83 cm.

Each compared method adopts the complete measured data to reconstruct its reference (Full View in Fig. 14). The sinogram is downsampled to 400, 800, and 1600 views to simulate the sparse view CT imaging. Fig. 14 shows the reconstructed images at different sparse view settings. When the number of sparse views is smaller, MetaInv-Net produces an image with better visual quality than FBP, PD-Net, and JSR-Net. HQS-CG produces the image with much fewer artifacts than MetaInv-Net. However, the details and edges in HQS-CG reconstruction are smoother than MetaInv-Net.

2) Different Training Set Size: In this experiment, we retrain the deep models with the same training setting (except training epochs=20) at different data set sizes (e.g., 1% and 5% of the full training set). For noise level σ = 0.2, we test the deep models on AAPM-8 at different numbers of scanning views. For each training setting, the curve of SSIM/PSNR versus the number of views is established in Fig. 13. MetaInv-Net is much less demanding in data quantity than PD-Net and JSR-Net.

F. Real Data Study

In electrocardiographic (ECG)-gated cardiac multiphase 4D imaging, the reconstruction of each phase is a sparse view CT imaging problem. The current setting is a realistic simulation with ground truth. In this section, the proposed MetaInv-Net is tested on real data, which is scanned with the same protocol to [72]. The original helical geometry is converted to a fan-beam geometry. There are a total of 2200 scanning views equally distributed 360°. The number of detector elements is 888, and its length is 95.93 cm. The reconstructed image has 800 × 800 pixels around 24.3 × 24.3 cm². The distance of source to rotation center is 53.85 cm and detector to rotation center is 49.83 cm.

Each compared method adopts the complete measured data to reconstruct its reference (Full View in Fig. 14). The sinogram is downsampled to 400, 800, and 1600 views to simulate the sparse view CT imaging. Fig. 14 shows the reconstructed images at different sparse view settings. When the number of sparse views is smaller, MetaInv-Net produces an image with better visual quality than FBP, PD-Net, and JSR-Net. HQS-CG produces the image with much fewer artifacts than MetaInv-Net. However, the details and edges in HQS-CG reconstruction are smoother than MetaInv-Net.
ment along with the increasing number of available scanning views. MetaInv-Net has a little better performance than other deep models and FBP, while it is inferior to HQS-CG. Note that the comparison here is not fair due to the lack of ground truth for real data reconstruction.

Fig. 15. SSIM/PSNR v.s. number of views for sparse view CT on real data.

IV. CONCLUSION

In this work, we propose a new deep model for sparse view CT. It contains much less trainable parameters and reconstructs images with higher quantitative than many state-of-the-art deep models. Since the MetaInv-Net is light-weighted, it can be well trained with much less training data. Furthermore, MetaInv-Net generalizes well beyond the training setting, e.g., to different scanning settings (number of views), noise levels, and noise types. As part of the future works, we will explore a better strategy to select hyperparameters and design more powerful data-adaptive image priors in the MetaInv-Net. Since the modifications of the MetaInv-Net to the backbone model is minor, convergence analysis of the MetaInv-Net is also interesting and meaningful.

ACKNOWLEDGMENT

Haimiao Zhang is supported by China Postdoctoral Science Foundation under grant 2018M641056. Baodong Liu is supported in part by National Key R&D Program of China grant No. 2017YFF0107201 and CAS Interdisciplinary Innovation Team, Project No. JCTD-2019-02. Bin Dong is supported in part by National Natural Science Foundation of China (NSFC) grant No. 11831002, Beijing Natural Science Foundation (No. 180001) and Beijing Academy of Artificial Intelligence (BAAI).

REFERENCES

[1] M. E. Davison, “The ill-conditioned nature of the limited angle tomography problem,” SIAM J. Appl. Math., vol. 43, no. 2, pp. 428–448, 1983.
[2] A. K. Louis, “Incomplete data problems in X-ray computerized tomography,” Numer. Math., vol. 48, no. 3, pp. 251–262, 1986.
[3] F. Natterer, The mathematics of computerized tomography. SIAM, 2001.
[4] G. Aubert and P. Kornprobst, Mathematical problems in image processing: partial differential equations and the calculus of variations. Springer, 2006.
[5] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” Physica D, vol. 60, no. 1, pp. 259–268, 1992.
[6] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in CVPR, vol. 2. IEEE, 2005, pp. 60–65.
[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” IEEE Trans. Image Process., vol. 16, no. 8, pp. 2080–2095, 2007.
[8] S. Gu, L. Zhang, W. Zhu, and X. Feng, “Weighted nuclear norm minimization with application to image denoising,” in CVPR, 2014, pp. 2862–2869.
[9] I. Daubechies, Ten Lectures on Wavelets. SIAM, 1992.
[10] S. Mallat, A Wavelet Tour of Signal Processing, The Sparse Way, 3rd ed. Burlington, MA: Academic Press, 2009.
[11] B. Dong and Z. Shen, MRA-Based Wavelet Frames and Applications. IAS Lecture Notes Series, Hong-Kai, ed. "Mathematics in Image Processing". Vol. 19, American Mathematical Society, 2013.
[12] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” IEEE Trans. Image Process., vol. 15, no. 12, pp. 3736–3745, 2006.
[13] J.-F. Cai, H. Ji, Z. Shen, and G.-B. Ye, “Data-driven tight frame construction and image denoising,” Appl. Comput. Harmon. Anal., vol. 37, no. 1, pp. 89–105, 2014.
[14] C. Tai and E. Weinan, “Multiscale adaptive representation of signals: I. the basic framework,” J. Mach. Learn. Res., vol. 17, no. 1, pp. 4875–4912, 2016.
[15] S. Osher, Z. Shi, and W. Zhu, “Low dimensional manifold model for image processing,” SIAM J. Imaging Sci., vol. 10, no. 4, pp. 1699–1690, 2017.
[16] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge: MIT Press, 2016.
[17] G. Wang, “A perspective on deep imaging,” IEEE Access, vol. 4, pp. 8914–8924, 2016.
[18] G. Wang, M. D. M. Kalra, and C. G. Orton, “Machine learning will transform radiology significantly within the next 5 years,” Med. Phys., vol. 44, no. 6, pp. 2041–2044, 2017.
[19] M. T. McCann, K. H. Jin, and M. Unser, “Convolutional neural networks for inverse problems in imaging: A review,” IEEE Signal Process. Mag., vol. 34, no. 6, pp. 85–95, 2017.
[20] G. Wang, J. C. Ye, K. Mueller, and J. A. Fessler, “Image reconstruction is a new frontier of machine learning,” IEEE Trans. Med. Imaging, vol. 37, no. 6, pp. 1289–1296, 2018.
[21] H. Zhang and B. Dong, “A review on deep learning in medical image reconstruction,” Journal of the Operations Research Society of China, 2020.
[22] H. Chen, Y. Zhang, M. K. Kalra, F. Lin, Y. Chen, P. Liao, J. Zhou, and G. Wang, “Low-dose CT with a residual encoder-decoder convolutional neural network,” IEEE Trans. Med. Imaging, vol. 36, no. 12, pp. 2524–2535, 2017.
[23] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, “Deep convolutional neural network for inverse problems in imaging,” IEEE Trans. Image Process., vol. 26, no. 9, pp. 4509–4522, 2017.
[24] E. Kang, J. Min, and J. C. Ye, “A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction,” Med. Phys., vol. 44, no. 10, pp. e360–e375, 2017.
[25] Z. Zhang, X. Liang, X. Dong, Y. Xie, and G. Cao, “A sparse-view CT reconstruction method based on combination of densenet and deconvolution.” IEEE Trans. Med. Imaging, vol. 37, no. 6, pp. 1407–1417, 2018.
[26] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, “Image reconstruction by domain-transform manifold learning,” Nature, vol. 555, pp. 769–774, 2018.
[27] Q. Yang, P. Yan, Y. Zhang, H. Yu, Y. Shi, X. Mou, M. K. Kalra, Y. Zhang, L. Sun, and G. Wang, “Low dose CT image denoising using a generative adversarial network with wasserstein distance and perceptual loss,” IEEE Trans. Med. Imaging, 2018.
[28] J. Yoo, S. Sabir, D. Heo, K. H. Kim, A. Wahab, Y. Choi, S. Lee, E. Y. Chae, H. H. Kim, Y. M. Bae et al., “Deep learning diffuse optical tomography,” IEEE Trans. Med. Imaging, vol. 39, no. 4, pp. 877–887, 2020.
[29] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in ICML, 2010, pp. 399–406.
[30] W. Zhang, D. Liu, J. Yang, W. Han, and T. Huang, “Deep networks for image super-resolution with sparse prior,” in ICCV, December 2015.
[31] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Scholkopf, “Learning to deblur,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 7, pp. 1439–1451, 2016.
[32] K. Zhang, W. Zuo, S. Gu, and L. Zhang, “Learning deep CNN denoiser prior for image restoration,” in CVPR, 2017.
[33] J. Zhang and B. Ghanem, “ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing,” in CVPR, 2018, pp. 1828–1837.
[34] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, “Denoising prior driven deep neural network for image restoration,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 10, pp. 2305–2318, 2019.
[35] Y. Li, M. Tolgih, J. Geng, V. Monga, and Y. C. Eldar, “Efficient and interpretable deep blind image deblurring via algorithm unrolling,” IEEE Trans. Comput. Imaging, vol. 6, pp. 666–681, 2020.
