2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)
Allerton Park and Retreat Center
Monticello, IL, USA, October 2-5, 2018

Fast and convergent iterative image recovery using trained convolutional neural networks

(Invited Paper)

Il Yong Chun, Hongki Lim, Zhengyu Huang, and Jeffrey A. Fessler
Department of Electrical Engineering and Computer Science, The University of Michigan
Ann Arbor, MI 48019-2122 USA
Email: iychun@umich.edu, hongki@umich.edu, zyhuang@umich.edu, fessler@umich.edu

***Abstract*—Deep image mapping networks have been recently applied to solving some inverse problems in imaging due to their good mapping capabilities. However, the greater mapping capability can increase the chance of causing some artificial features when test images differ from training images. Combining image mapping networks with an iterative image recovery that naturally considers imaging system physics is an alternative approach to solving inverse problems. This alternative approach can avoid artificial features, by (back-)projecting the output signals of image mapping networks while considering the imaging system physics. By generalizing the state-of-the-art iterative image recovery algorithm using learned regularizer with iteration-wise image mapping networks, this paper proposes a new recurrent convolutional neural network, *Momentum-Net*. In addition, this paper investigates the theoretical convergence behavior of Momentum-Net. Numerical experiments show that, for *a)* denoising low signal-to-noise-ratio images, and *b)* sparse-view X-ray computed tomography, the proposed Momentum-Net achieves significantly more accurate and faster image recovery, compared to the state-of-the-art data-driven regularizer or the unsupervised autoencoding regularizer.**

I. Introduction

Convolutional analysis operator learning (CAOL) can train an autoencoding convolutional neural network (CNN) in an unsupervised learning fashion to more accurately solve inverse problems [1]. CAOL has several theoretical and practical benefits from the perspectives of both training and testing. First, CAOL can benefit from "big data", i.e., training data consisting of many (high-dimensional) signals [2]. Second, the autoencoding CNNs trained via CAOL can achieve compressed sensing recovery guarantees [2]. Third, CAOL is useful for training deep layered CNNs within unsupervised learning frameworks. Finally, for sparse-view X-ray computed tomography (CT), trained autoencoding CNNs via CAOL significantly improves image reconstruction accuracy over the existing state-of-the-art edge-preserving (EP) regularizer [1]. Combined with *Block Proximal Gradient method using a Majorizer* (BPG-M) [1], [3] and sharp majorizers, one can achieve accelerated CAOL and fast signal recovery in solving inverse problems using the trained autoencoding CNNs while guaranteeing algorithmic convergence [2]. Nonetheless, the corresponding iterative algorithm needs several hundreds of iterations to converge, detracting from its practical use.

BCD-Net, a recurrent CNN designed to achieve fast iterative image recovery, consists of image mapping and reconstruction modules; its image mapping module is an autoencoding CNN motivated by CAOL [4]. By training image mapping modules in a supervised learning fashion, BCD-Net achieved fast image recovery within a few dozen of layers, for denoising low signal-to-noise-ratio (SNR) images and for reconstructing highly undersampled magnetic resonance imaging (MRI) data [4]. In addition, for *a)* denoising low SNR images, *b)* highly undersampled MRI, and *c)* low-count positron-emission tomography (PET), BCD-Net significantly improved image recovery accuracy compared to the existing state-of-the-art (data-driven) regularizers [4], [5]. Nonetheless, BCD-Net has no convergence guarantees, and for some imaging modalities, e.g., CT, PET, parallel MRI, etc., the method requires multiple iterations in each reconstruction module (see, e.g., [5]), diminishing image recovery speed.

This paper introduces a new recurrent CNN architecture where each layer consists of *1)* image mapping, *2)* extrapolation, and *3)* reconstruction modules, aiming for fast and convergent iterative image recovery. Because this recurrent CNN uses *momentum* terms in its extrapolation module, we call it *Momentum-Net*. Similar to BCD-Net, Momentum-Net uses *iteration-wise* image mapping modules that are trained in a supervised way to effectively remove iteration-wise artifacts and give "best" image estimates at each layer. In addition, this paper investigates fixed-point convergence behavior of Momentum-Net under some mild conditions, and influence of the mathematical property of autoencoding CNNs to Momentum-Net. Numerical experiments show that, for denoising low SNR images and sparse-view CT, the proposed Momentum-Net significantly improves both image recovery accuracy and speed compared to the state-of-the-art data-driven or trained regularizers, e.g., K-SVD [6] and pre-learned CAOs [1].

II. Iterative Image Recovery Using Unsupervised Autoencoding CNNs and BPG-M

This section motivates the architectures of Momentum-Net. Specifically, we introduce the updates of BPG-M for solving

The second and third authors contributed equally to this work and are listed in alphabetical order. This work is supported in part by the Keck Foundation, NIH U01 EB018753, and NIH R01 EB022075.

inverse problems with unsupervised autoencoding CNNs, i.e., learned CAOs [1].

A. Image recovery model using learned CAOs

Consider the following (block separable) convex optimization problem for recovering signal x from measurements y [1, (P3)]:

$$\underset{x\in\mathcal{X}}{\operatorname{argmin}}\ f(x;y)+\lambda\Big(\min_{\{z_k\}} r(x,\{z_k\})\Big),$$
$$r(x,\{z_k\}) := \sum_{k=1}^{K}\frac{1}{2}\|d_k\circledast x - z_k\|_2^2 + \alpha_k\|z_k\|_1, \quad (1)$$

where $\mathcal{X}$ is a closed convex set, $f(x;y)$ is a differentiable convex data fidelity term, $r(x,\{z_k\})$ is a regularizer trained via CAOL in an unsupervised way, i.e., the learned CAO [1], $\{z_k : k=1,\dots,K\}$ is a set of sparse codes, $\{d_k\in\mathbb{C}^R : k=1,\dots,K\}$ is a set of learned convolutional kernels, and R and K denote the size and number of learned filters, respectively. This model significantly improves image reconstruction accuracy in sparse-view CT (when ℓ^1 norm is replaced with nonconvex ℓ^0 quasi-norm) – see details in [1], [7], [8].

B. Convergent image recovery using learned CAOs: BPG-M

In BPG-M [1], [3], we consider a more general concept than Lipschitz continuity of the gradient and define the following:

Definition 1 (M-Lipschitz continuity [1, Def. 3.1])**.** *A function $g:\mathbb{R}^n\to\mathbb{R}^n$ is* M-Lipschitz continuous *on $\mathbb{R}^n$ if there exists a (symmetric) positive definite matrix M such that*

$$\|g(x)-g(y)\|_{M^{-1}} \le \|x-y\|_M, \quad \forall x,y,$$

where $\|x\|_M^2 := x^TMx$.

Suppose that *1)* the gradient of $f(x;y)$ is $M_x^{(i+1)}$-Lipschitz continuous, and *2)* the learned filters in (1) satisfies the tight-frame (TF) condition,

$$\sum_{k=1}^{K}\|d_k\circledast x\|_2^2 = \|x\|_2^2, \quad \forall x, \quad (2)$$

for some boundary conditions [1, Prop. 2.1]. BPG-M uses the following block-wise updates to solve (1):

$$x_0^{(i+1)} = \sum_{k=1}^{K}(Fd_k^*)\circledast\mathcal{T}_{\alpha_k}(d_k\circledast x^{(i)}), \quad (3)$$
$$\acute{x}^{(i+1)} = x^{(i)} + E_x^{(i+1)}\big(x^{(i)}-x^{(i-1)}\big), \quad (4)$$
$$x^{(i+1)} = \underset{x\in\mathcal{X}}{\operatorname{argmin}}\frac{1}{2}\Big\|x-\acute{x}^{(i+1)}+\big(M_x^{(i+1)}\big)^{-1}\nabla f(\acute{x}^{(i+1)};y)\Big\|_{M_x^{(i+1)}}^2 + \frac{\lambda}{2}\|x-x_0\|_2^2, \quad (5)$$

where extrapolation matrix $E_x^{(i+1)}$ is updated by

$$E_x^{(i+1)} = \delta w^{(i)}\cdot\big(M_x^{(i+1)}\big)^{-1/2}\big(M_x^{(i)}\big)^{1/2}, \quad (6)$$

Algorithm 1 Momentum-Net for convex $\{f(x;y),\mathcal{X}\}$

Require: $\{\mathcal{M}_{\theta^{(i)}},\kappa^{(i+1)}\in(0,\infty): i=1,\dots,N_{\text{Nets}}\}$, $\rho\in(0,1)$, $\{w^{(i)}\in[0,1]: i=1,\dots,N_{\text{Nets}}\}$, $\lambda>0$, $x^{(0)}=x^{(-1)}$, y

for $i=0,\dots,N_{\text{Nets}}-1$ **do**

Image mapping:

if $\kappa^{(i+1)}>1$ **then**

$x_0^{(i+1)} = (1-\rho)x^{(i)} + \frac{\rho}{\kappa^{(i+1)}}\mathcal{M}_{\theta^{(i+1)}}\big(x^{(i)}\big)$

else

$x_0^{(i+1)} = (1-\rho)x^{(i)} + \rho\mathcal{M}_{\theta^{(i+1)}}\big(x^{(i)}\big)$

end if

Calculate $M_x^{(i+1)}$ for $f(x;y)$ and $E_x^{(i+1)}$ by (6)

Extrapolation:

$\acute{x}^{(i+1)} = x^{(i)} + E_x^{(i+1)}\big(x^{(i)}-x^{(i-1)}\big)$

Image reconstruction:

$x^{(i+1)} = \underset{x\in\mathcal{X}}{\operatorname{argmin}}\frac{1}{2}\Big\|x-\acute{x}^{(i+1)}+\big(M_x^{(i+1)}\big)^{-1}\nabla f(\acute{x}^{(i+1)};y)\Big\|_{M_x^{(i+1)}}^2 + \lambda r\big(x,x_0^{(i+1)}\big)$

end for

and $M_x^{(i+1)}$ is a majorization matrix for $f(x;y)$, $\forall i$. In (3), the soft-thresholding operator $\mathcal{T}_\alpha(x):\mathbb{C}^N\to\mathbb{C}^N$ is

$$\mathcal{T}_\alpha(x)_n := \begin{cases} x_n-\alpha\cdot\operatorname{sign}(x_n), & |x_n|>\alpha,\\ 0, & \text{otherwise,}\end{cases}$$

$F\in\mathbb{C}^{R\times R}$ flips a column vector in the vertical direction (e.g., it rotates 2D filters by 180°), and $(\cdot)^*$ indicates complex conjugate. The update in (3) is derived by rewriting $\sum_{k=1}^K\|d_k\circledast x - z_k^{(i+1)}\|_2^2$ to $\|x-\sum_{k=1}^K(Fd_k^*)\circledast z_k^{(i+1)}\|_2^2$ with the TF condition above, where $z_k^{(i+1)} = \mathcal{T}_{\alpha_k}(d_k\circledast x^{(i)})$, $\forall k,i$. Here, for the $\{z_k\}$-updates, we do not use extrapolation, e.g., $\{w^{(i+1)}=0:\forall i\}$, in (6), because we can find sharp majorizers $\{M_{z_k}=I:\forall k\}$ (that are exact Hessian of $\{z_k\}$-updating problems in (1)). The BPG-M updates in (3)–(5) guarantee that any limit point of $\{x^{(i+1)},x_0^{(i+1)}: i\ge 0\}$ is a Nash point of (1) [3, Thm. 3.3]. The following section generalizes the BPG-M updates in (3)–(5) and constructs Momentum-Net.

III. Momentum-Net: Where BPG-M Meets Fixed-Point Iterations for Inverse Problems

A. Architecture

We first rewrite the BPG-M update (3) as a general image mapping function $\mathcal{M}_\theta:\mathbb{C}^N\to\mathbb{C}^N$ with $\mathcal{M}_\theta(x) = \sum_{k=1}^K(Fd_k^*)\circledast\mathcal{T}_{\alpha_k}(d_k\circledast x)$ that satisfies (2). We then establish Momentum-Net by considering the BPG-M updates in (3)–(5) and further generalizing image mapping operators $\mathcal{M}_\theta$ to iteration-wise image mapping operators $\{\mathcal{M}_{\theta^{(i+1)}}:\forall i\}$. Each layer of Momentum-Net consists of *1)* image mapping, *2)* extrapolation, and *3)* reconstruction modules, corresponding to the BPG-M updates (3)–(5). See the architecture of Momentum-Net in Algorithm 1.

The image mapping formula in Algorithm 1, i.e.,

$$x_0^{(i+1)} = (1-\rho)x^{(i)} + \frac{\rho}{\kappa^{(i+1)}}\mathcal{M}_{\theta^{(i+1)}}\big(x^{(i)}\big),$$

is motivated by an averaged operator in the fixed-point literature [9], but with additional normalization factor $1/\kappa^{(i+1)}$. The measure $\kappa^{(i+1)}$ is important for the ith image mapping function $\mathcal{M}_{\theta^{(i+1)}}$ in guaranteeing convergence of Momentum-Net [10] – see the definition of $\kappa^{(i+1)}$ in the next section. Note that *iteration-wise* image mapping operators in Momentum-Net are supervisedly trained to effectively remove iteration-wise artifacts and give "best" signal estimates at each layer.

Examples of the coupling function $r(x, x_0)$ of the image reconstruction formula in Algorithm 1 include a Gaussian prior $r(x, x_0) = \frac{1}{2}\|x - x_0\|_2^2$ [1], a compressible prior $r(x, x) = \|x - x_0\|_1$ [7], [8], etc.

B. Understanding general image mapping operators

To mathematically understand mapping operators $\{\mathcal{M}_{\theta^{(i+1)}} : \forall i\}$, we first introduce the κ-Lipschitz continuity of a mapping function:

Definition 2 (Lipschitz continuity of a mapping function)**.** *A function $\mathcal{M} : n \to n$ is κ-Lipschitz continuous if there exist Lipschitz constants $0 < \kappa < \infty$ such that*

$$\|\mathcal{M}(x) - \mathcal{M}(y)\|_2 \leq \kappa\|x - y\|_2, \quad \forall x, y \in \mathbb{R}^n.$$

The autoencoding CNN (using identical filters in both encoders and decoders, e.g., [4]) in BPG-M update (3) is an example of image mapping function $\mathcal{M}_\theta(x)$, i.e.,

$$\mathcal{M}_\theta(x) = \sum_{k=1}^{K} (F d_k^*) \circledast \mathcal{T}_{\alpha_k}(d_k \circledast x). \tag{7}$$

Particularly, (3) involves with a proximal operator, i.e., $\mathcal{M}_\theta(x) = \sum_{k=1}^{K} (F d_k^*) \circledast \mathrm{Prox}_{\|\cdot\|_1}(d_k \circledast x)$, where the proximal operator with a convex function g is defined by

$$\mathrm{Prox}_g(z) := \operatorname*{argmin}_x \frac{1}{2}\|x - z\|_2^2 + g(x). \tag{8}$$

The mapping operator in (7) is $R \cdot \lambda_{\max}(D^H D)$-Lipschitz continuous, where $\lambda_{\max}(\cdot)$ is the maximum eigenvalue of a matrix and $D := [d_1, \ldots, d_K]$. Thus, when the filters $\{d_k : \forall k\}$ satisfy the TF condition, e.g., learned CAOs that satisfies $D^H D = \frac{1}{R} I$ [1], the autoencoding CNN in (7) is 1-Lipschitz continuous, i.e., nonexpansive.

Some image mapping operators $\mathcal{M}_\theta(x)$ can have more complicated structures: they may *not* involve proximal operators and there may exist *no* corresponding explicit function in a form of (8). In Momentum-Net, we are particularly interested in such mapping operates, and Definition 2 is useful in understanding them in algorithmic convergence. A simple example is the autoencoding CNN that uses distinct filters for encoding and decoding [11]:

$$\mathcal{M}_\theta(x) = \sum_{k=1}^{K} w_k \circledast \mathcal{T}_{\alpha_k}(d_k \circledast x). \tag{9}$$

For this example, there is no explicit cost function that corresponds to (9). However, we can still have that the autoencoder in (9) is $R \cdot \sqrt{\lambda_{\max}(W^H W)} \cdot \sqrt{\lambda_{\max}(D^H D)}$-Lipschitz continuous, where $W := [w_1, \ldots, w_K]$ [10].

When the soft-thresholding operators in (9) are replaced with the ReLU operator [12], i.e., $\mathrm{ReLU}(x) := \max(x, 0)$, the modified autoencoder is also $R \cdot \sqrt{\lambda_{\max}(W^H W)} \cdot \sqrt{\lambda_{\max}(D^H D)}$-Lipschitz continuous, because ReLU is a proximal operator with indicator function of the nonnegativity constraint and this indicator function is convex. One can straightforwardly apply the techniques in this section, to estimate a Lipschitz constant of deep-layered autoencoding CNNs [1].

C. Convergence analysis

A key challenge in analyzing the convergence of Momentum-Net (i.e., Algorithm 1) is that the image mapping operator $\{\mathcal{M}_{\theta^{(i+1)}} : \forall i\}$ can change across iterations, and the corresponding $g_{\theta^{(i+1)}}(\cdot)$ is not explicitly defined in the form of (8). This section analyzes the convergence behavior of Algorithm 1 under the following assumptions:

Assumption i) The data fidelity function $f(x; y)$ is proper, convex, differentiable, and lower-bounded. The coupling function $r(x, x_0)$ is proper, convex, and lower semicontinuous. The set $\mathcal{X}$ of feasible points is convex and closed. Algorithm 1 has a fixed point.

Assumption ii) $\nabla f(x; y)$ is $M_x^{(i+1)}$-Lipschitz continuous, where $M_x^{(i+1)}$ is a majorization matrix that satisfies $m_x I_N \preceq M_x^{(i+1)}$ with $0 < m_x < \infty$, $\forall i$.

Assumption iii) The image mapping operators are identical across the layers, i.e., $\mathcal{M}_\theta = \mathcal{M}_{\theta^{(i+1)}}$, $\forall i$. The function $\mathcal{M}_\theta$ is κ-Lipschitz continuous with a Lipschitz constant $\kappa \in (0, \infty)$.

Assumption iv) The extrapolation matrices $E_x^{(i+1)} \succeq 0$ satisfy $(E_x^{(i+1)})^T M_x^{(i+1)} E_x^{(i+1)} \preceq \delta^2 \cdot M_x^{(i)}, \quad \delta < 1$.

Under Assumptions i–iv, we first show that any limit points of the sequence generated by Momentum-Net (Algorithm 1) converge to a fixed-point:

Theorem 3 (A limit point is a fixed-point)**.** *Let $\{\tilde{x}^{(i+1)} : i \geq 0\}$ be the sequence generated by Algorithm 1, where $\tilde{x} := [x^T, x_0^T]^T$. Then, any limit point $\bar{\tilde{x}}$ of $\{\tilde{x}^{(i+1)} : i \geq 0\}$ satisfies the fixed-point condition, i.e.,*

$$\bar{\tilde{x}} = \bar{\mathcal{A}}(\bar{\tilde{x}}), \tag{10}$$

where $\bar{\mathcal{A}} := \mathcal{A}_{\bar{M}_x}$, $\mathcal{A}_{M_x^{(i+1)}}(\tilde{x}^{(i)})$ denotes the ith functions in Algorithm 1 that apply to $\tilde{x}^{(i)}$, and $M_x \to \bar{M}_x$.

Proof. Specialize proofs in [10] for $\{\mathcal{M}_\theta = \mathcal{M}_{\theta^{(i+1)}} : i \geq 0\}$.

We can show that Theorem 3 holds for Algorithm 1 without having $\{\mathcal{M}_\theta = \mathcal{M}_{\theta^{(i+1)}} : i \geq 0\}$ in Assumption iii [10].

D. Momentum-Net vs BCD-Net

One can view Momentum-Net as a generalized version of BCD-Net in [4]. Each layer of BCD-Net consists of *1)* image mapping and *2)* reconstruction modules [4]:

$$x_0^{(i+1)} = \mathcal{M}_{\theta^{(i+1)}}\left(x^{(i)}\right)$$
$$x^{(i+1)} = \operatorname*{argmin}_{x \in \mathcal{X}} f(x; y) + \frac{\lambda}{2}\left\|x - x_0^{(i+1)}\right\|_2^2,$$

for $i = 0, \dots, N_{\text{Nets}} - 1$. By setting $\rho \to 1$ and $r(x, x_0) = \frac{1}{2}\|x - x_0\|_2^2$, Momentum-Net becomes BCD-Net for some image recovery applications, e.g, image denoising.

Compared to Momentum-Net, BCD-Net has some caveats. Besides having no convergence guarantees, BCD-Net may require multiple iterations for its reconstruction module in some imaging applications, e.g., CT, PET, parallel MRI, etc. For example, in low-count PET, many expectation-maximization iterations have been used at each reconstruction module [5]. In general, each iteration at the reconstruction module applies forward and backward projection operators of imaging systems, and requiring large computational complexities in some imaging applications, e.g., CT, PET, parallel MRI using many receive coils, etc. In other words, BCD-Net can be slow for such imaging applications.

Momentum-Net overcomes the multiple iteration issues of the reconstruction module in BCD-Net, because the image reconstruction module of Momentum-Net can have closed-form solution and use only a single iteration for a wide range of imaging applications. In addition, the closed-form solution at each reconstruction module is useful in stably guaranteeing the fixed-point convergence of Momentum-Net.

IV. Results and Discussion

A. Experimental setup

1) Imaging and image recovery: For image denoising experiments, we contaminated 64 512×512-sized slices of XCAT phantom [13] by low SNR additive white Gaussian noise (AWGN) with the standard deviation ≈ 202 in Hounsfield units (HU) (that correspond to ≈ 22 peak SNR (PSNR) value); we used 60 of them for training and the remaining four for testing. For sparse-view CT experiments, we simulated 152 highly undersampled (12.5% projection views) sinograms with fan-beam geometry and 10^5 incident photons per ray and electronic noise variance 5^2 (see details of the setup in [1]), while avoiding an inverse crime with 152 840×840-sized different slices of the XCAT phantom [13] ($\Delta_x = \Delta_y = 0.4883$ mm); we used 150 of them for training and the remaining two for testing. We reconstructed 420×420-sized images with a coarser grid ($\Delta_x = \Delta_y = 0.9766$ mm). (For simplicity, we report results from a single recovered image throughout the paper.)

For proposed Momentum-Net (using $\rho = 1 - \epsilon$, where ϵ is a machine epsilon), we set the regularization parameter λ as follows: for image denoising, we used the value in [4]; for sparse-view CT, $\lambda = 13 \times 10^6$ that is finely tuned to give the lowest root-mean-square-error (RMSE) value for BPG-M-based image recovery using learned CAOs of $K = 49$ and $R = 7 \times 7$ [1]. For the methods in comparison with Momentum-Net, we finely tuned their parameters to give the highest PSNR or lowest RMSE values (e.g., *K*-SVD [6] used 64×256-sized dictionary). We evaluated the quality of recovered images by PSNR or RMSE (in HU).

2) Training Momentum-Net: For image mapping operators $\{\mathcal{M}_{\theta^{(i+1)}} : i \geq 0\}$ in Momentum-Net, we trained general autoencoding CNNs in the form of (9) with $\{w_k^{(i)} \in \mathbb{R}^R, d_k^{(i)} \in \mathbb{R}^R, \alpha_k^{(i)} : k = 1, \dots, K, i = 1, \dots, N_{\text{Nets}}\}$, via stochastic gradient method, ADAM [14], using the mini-batch size 10. For image denoising experiments, we trained Momentum-Net with $K = 81$, $R = 8 \times 8$, $N_{\text{Nets}} = 5$, 10^3 epochs, and learning rates 10^{-3}, 10^{-4}, and 2.5, for $\{w_k^{(i)} : \forall i\}$, $\{d_k^{(i)} : \forall i\}$, and $\{\alpha_k^{(i)} : \forall i\}$, respectively. For sparse-view CT, we trained Momentum-Net with $K = 49$, $R = 7 \times 7$, $N_{\text{Nets}} = 100$, 10^2 epochs, and learning rates 10^{-3}, 10^{-3}, and 10^{-1}, for $\{w_k^{(i)} : \forall i\}$, $\{d_k^{(i)} : \forall i\}$, and $\{\alpha_k^{(i)} : \forall i\}$, respectively.

TABLE I
Comparison of different image denoising and CT reconstruction methods

Denoising images corrupted by low SNR AWGN				
Measure	Noisy	Wiener	*K*-SVD [6]	Momentum-Net
PSNR (dB)	22.1	29.8	36.7	**41.5**

Image reconstruction from sparse-view CT (12.5% projection views)				
Measure	FBP†	EP†	CAOL [1]	Momentum-Net
RMSE (HU)	82.8	40.8	34.7	**24.6**

†FBP and EP stands for filtered back-projection and edge preserving regularization.

B. Fast and accurate iterative image recovery via trained Momentum-Net

The proposed Momentum-Net significantly improves both image recovery accuracy and speed in both the image denoising and sparse-view CT reconstruction experiments, compared to the state-of-the-art data-driven or unsupervised regularizers, e.g., *K*-SVD [6] and autoencoding CNN (7) trained via CAOL [1]. See Table I and Fig. 1 for comparing image recovery accuracy; and see Fig. 2 for comparing image recovery speed. Note that *K*-SVD [6] requires more computational costs at each iteration, compared to that needed at each layer of Momentum-Net. Other than computations required in processing extracted patches, each iteration of *K*-SVD requires additional computations to learn dictionary, extract many overlapping patches, and aggregate the processed patches. The computational complexity of BPG-M-based image recovery using learned CAOs [1] at each iteration is identical to that of Momentum-Net at each layer (the filter dimension in their autoencoders is identical).

V. Conclusion

The proposed Momentum-Net achieves fast and accurate iterative image recovery in "extreme" imaging problems. Compared to the state-of-the-art data-driven regularizer, e.g., [6], or autoencoder trained in an unsupervised manner [1], Momentum-Net achieves faster and more accurate iterative image recovery in denoising low SNR images and sparse-view CT reconstruction. In addition, Momentum-Net can guarantee fixed-point convergence under some mild conditions on data fidelity function $f(x; y)$, coupling function $r(x, x_0)$, and set $\mathcal{X}$. In particular, the mathematical property of image mapping networks is important in understanding convergence behavior of Momentum-Net. Finally, different from BCD-Net [4], Momentum-Net can lead faster and more

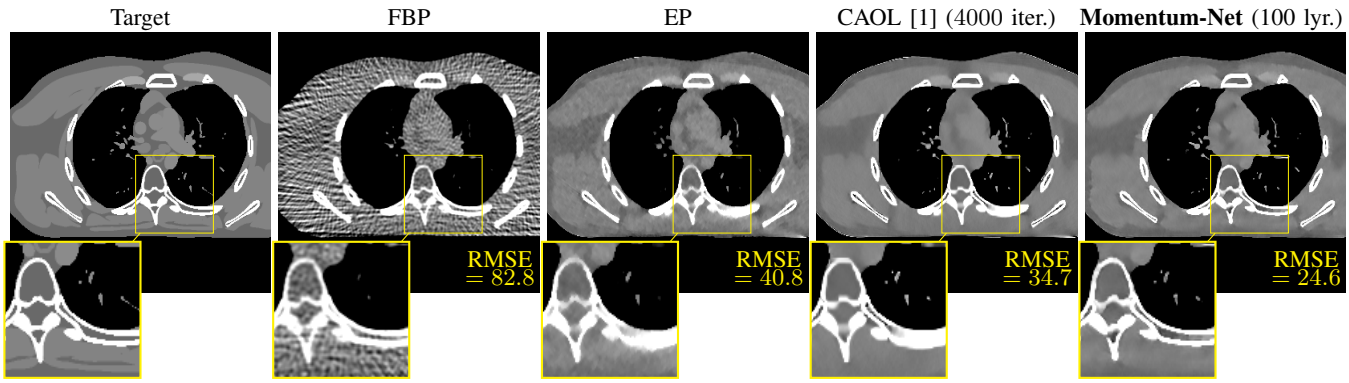
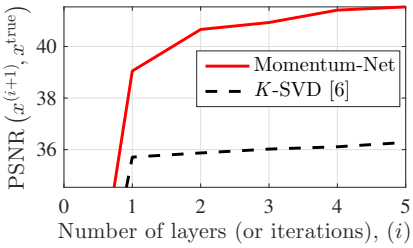
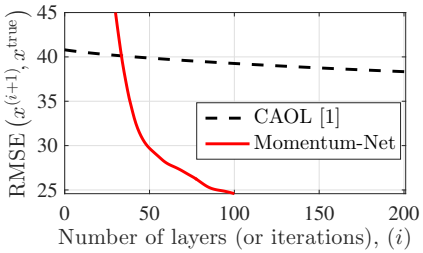


Fig. 1. Comparison of reconstructed images from different X-ray CT reconstruction models with sparse-views (fan-beam geometry with 12.5% projections views and 10^5 incident photons; display window $[800, 1200]$ HU). The proposed Momentum-Net outperforms the state-of-the-art autoencoding CNN (7) trained via CAOL in an unsupervised manner [1].



(a) Denoising images corrupted by low SNR AWGN



(b) Sparse-view CT reconstruction (12.5% proj. views)

Fig. 2. PSNR maximization or RMSE minimization behavior of image denoising and CT reconstruction (for CT reconstruction, BPG-M-based image recovery using learned CAOs [1] was initialized with EP solution; and Momentum-Net was initialized with FBP solution). The proposed Momentum-Net achieves faster and more accurate image recovery, compared to the state-of-the-art data-driven or unsupervised regularizer, e.g., K-SVD [6] and autoencoding CNN (7) trained via CAOL [1].

stable iterative image recovery by not requiring multiple iterations at each reconstruction module, for a wide range of imaging applications.

Future works include investigating its convergence guarantee with nonconvex $f(x; y)$ and convergence behavior to critical points. This aside, we expect sharper edge recovery can be made by training image mapping networks with more advanced training costs, e.g., perceptual similarity, local noise power spectrum, etc.

REFERENCES

[1] I. Y. Chun and J. A. Fessler, "Convolutional analysis operator learning: Acceleration, convergence, application, and neural networks," submitted, Jan. 2018.

[2] I. Y. Chun, D. Hong, B. Adcock, and J. A. Fessler, "Convolutional analysis operator learning: Dependence on training data and compressed sensing recovery guarantees," preprint, Aug. 2018.

[3] I. Y. Chun and J. A. Fessler, "Convolutional dictionary learning: Acceleration and convergence," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1697–1712, Apr. 2018.

[4] ——, "Deep BCD-net using identical encoding-decoding CNN structures for iterative image recovery," in *Proc. IEEE IVMSP Workshop*, Zagori, Greece, Jun. 2018.

[5] Y. K. D. Hongki Lim, Jeffrey A. Fessler and I. Y. Chun, "Application of trained Deep BCD-Net to iterative low-count pet image reconstruction," in *Proc. IEEE NSS-MIC* (to appear), Sydney, Australia, Nov. 2018.

[6] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[7] X. Zheng, I. Y. Chun, Z. Li, Y. Long, and J. A. Fessler, "Sparse-view X-ray CT reconstruction using ℓ_1 prior with learned transform," submitted, Oct. 2017.

[8] I. Y. Chun, X. Zheng, Y. Long, and J. A. Fessler, "Sparse-view X-ray CT reconstruction using ℓ_1 regularization with learned sparsifying transform," in *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, Xi'an, China, Jun. 2017, pp. 115–119.

[9] E. K. Ryu and S. Boyd, "Primer on monotone operator methods," *Appl. Comput. Math*, vol. 15, no. 1, pp. 3–43, Jan. 2016.

[10] I. Y. Chun, H. Lim, Z. Huang, and J. A. Fessler, "Fast and convergent Momentum-Net: Where block optimization meets fixed-point iterations for inverse problems," preprint, Sep. 2018.

[11] S. Ravishankar, I. Y. Chun, and J. A. Fessler, "Physics-driven deep training of dictionary-based algorithms for mr image reconstruction," in *Proc. Asilomar Conf. on Signals, Syst., and Comput.*, Pacific Grove, CA, Nov. 2017.

[12] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. ICML*, Haifa, Israel, Jun. 2010, pp. 807–814.

[13] W. P. Segars, M. Mahesh, T. J. Beck, E. C. Frey, and B. M. Tsui, "Realistic CT simulation using the 4D XCAT phantom," *Med. Phys.*, vol. 35, no. 8, pp. 3800–3808, Jul. 2008.

[14] D. P. Kingma and J. L. Ba, "ADAM: A method for stochastic optimization," in *Proc. ICLR* 2015, San Diego, CA, May 2015, pp. 1–15.