Momentum-Net: Fast and convergent iterative neural network for inverse problems

Il Yong Chun, *Member, IEEE,* Zhengyu Huang*, Hongki Lim*, *Student Member, IEEE,* and Jeffrey A. Fessler, *Fellow, IEEE*

Abstract—Iterative neural networks (INN) are rapidly gaining attention for solving inverse problems in imaging, image processing, and computer vision. INNs combine regression NNs and an iterative model-based image reconstruction (MBIR) algorithm, often leading to both good generalization capability and outperforming reconstruction quality over existing MBIR optimization models. This paper proposes the first fast and convergent INN architecture, *Momentum-Net*, by generalizing a block-wise MBIR algorithm that uses momentum and majorizers with regression NNs. For fast MBIR, Momentum-Net uses *momentum* terms in extrapolation modules, and noniterative MBIR modules at each iteration by using *majorizers*, where each iteration of Momentum-Net consists of three core modules: image refining, extrapolation, and MBIR. Momentum-Net guarantees convergence to a fixed-point for general differentiable (non)convex MBIR functions (or data-fit terms) and convex feasible sets, under two asymptomatic conditions. To consider data-fit variations across training and testing samples, we also propose a regularization parameter selection scheme based on the "spectral spread" of majorization matrices. Numerical experiments for light-field photography using a focal stack and sparse-view computational tomography demonstrate that, given identical regression NN architectures, Momentum-Net significantly improves MBIR speed and accuracy over several existing INNs; it significantly improves reconstruction quality compared to a state-of-the-art MBIR method in each application.

Index Terms—Iterative neural network, deep learning, model-based image reconstruction, inverse problems, block proximal extrapolated gradient method, block coordinate descent method, light-field photography, X-ray computational tomography.

1 Introduction

Deep regression neural network (NN) methods have been actively studied for solving diverse inverse problems, due to their effectiveness at mapping noisy signals into clean signals. Examples include image denoising [1]–[4], image deconvolution [5], [6], image super-resolution [7], [8], magnetic resonance imaging (MRI) [9], [10], X-ray computational tomography (CT) [11]–[13], and light-field (LF) photography [14], [15]. However, regression NNs with a greater mapping capability have increased overfitting/hallucination risks [16]–[19]. An alternative approach to solving inverse problems is an *iterative NN* (INN) that combines regression NNs – called "refiners" or denoisers – with an unrolled iterative model-based image reconstruction (MBIR) algorithm [20]–[27]. This alternative approach can regulate overfitting of regression NNs, by balancing physical data-fit of MBIR and prior information estimated by refining NNs [16], [18]. This "soft-refiner" approach has been successfully applied to several extreme imaging systems, e.g., highly undersampled MRI [20], [25], [28]–[30], low-dose or sparse-view CT [16], [19], [24], [27], [31], and low-count emission tomography [18], [32]–[34].

- *The authors indicated by asterisks (*) contributed equally to this work.*
- *This work is supported in part by the Keck Foundation, NIH U01 EB018753, NIH R01 EB023618, NIH R01 EB022075, and NSF IIS 1838179.*

- *Il Yong Chun was with the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48019 USA, and is now with the Department of Electrical Engineering, the University of Hawai'i at Mānoa, Honolulu, HI 96822 USA (email: iychun@hawaii.edu). Zhengyu Huang, Hongki Lim, and Jeffrey A. Fessler are with the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, MI 48019 USA (email: zyhuang@umich.edu; hongki@umich.edu; fessler@umich.edu).*

- *This paper has appendices. The prefix "A" indicate the numbers in section, theorem, equation, figure, table, and footnote in the appendices.*
- *This paper was presented in part at the Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, in Oct. 2018.*

1.1 Notation

This section provides mathematical notations. We use $f(x;y)$ to denote a function f of x given y. We use $\|\cdot\|_p$ to denote the ℓ^p-norm and write $\langle\cdot,\cdot\rangle$ for the standard inner product on $\mathbb{C}^N$. The weighted ℓ^2-norm with a Hermitian positive definite matrix A is denoted by $\|\cdot\|_A = \|A^{\frac{1}{2}}(\cdot)\|_2$. The Frobenius norm of a matrix is denoted by $\|\cdot\|_{\mathrm{F}}$. $(\cdot)^T$, $(\cdot)^H$, and $(\cdot)^*$ indicate the transpose, complex conjugate transpose (Hermitian transpose), and complex conjugate, respectively. $\mathrm{diag}(\cdot)$ denotes the conversion of a vector into a diagonal matrix or diagonal elements of a matrix into a vector. For (self-adjoint) matrices $A, B \in \mathbb{C}^{N\times N}$, the notation $B \preceq A$ denotes that $A - B$ is a positive semi-definite matrix.

1.2 From block-wise optimization to INN

To recover signals $x \in \mathbb{C}^N$ from measurements $y \in \mathbb{C}^m$, consider the following MBIR optimization problem:

$$\underset{x\in\mathcal{X}}{\operatorname{argmin}}\ F(x;y,z), \quad F(x;y,z) \triangleq f(x;y) + \frac{\gamma}{2}\|x-z\|_2^2, \tag{P0}$$

where $\mathcal{X}$ is a set of feasible points, $f(x;y)$ is data-fit function, γ is a regularization parameter, and $z \in \mathbb{C}^N$ is some high-quality approximation to the true unknown signal x.

The data-fit $f(x;y)$ measures deviations of model-based predictions of x from data y, considering models of imaging physics (or image formation) and noise statistics in y. In (P0), the signal recovery accuracy increases as the quality of z improves [17, Prop. 3]; however, it is difficult to obtain such z in practice. Alternatively, there has been a growing trend in learning sparsifying regularizers (e.g., convolutional regularizers [24], [35]–[38]) from training datasets and applying the trained regularizers to the following block-wise MBIR problem: $\operatorname{argmin}_{x\in\mathcal{X}} f(x;y) + \min_{\zeta} r(x,\zeta;\mathcal{O})$. Here, a learned regularizer $\min_{\zeta} r(x,\zeta;\mathcal{O})$ quantifies consistency between x and refined sparse signal ζ via some learned operators $\mathcal{O}$. Recently, we have constructed INNs by generalizing the corresponding block-wise MBIR updates with regression NNs without convergence analysis [25], [27]. In existing INNs, two major challenges exist: convergence and acceleration.

1.3 Challenges in existing INNs: Convergence

Existing convergence analysis has some practical limitations. The original form of plug-and-play (PnP [23], [39]–[41]) is motivated by the alternating direction method of multipliers (ADMM [42]), and its fixed-point convergence has been analyzed with consensus equilibrium perspectives [23]. However, similar to ADMM, its practical convergence depends on how one selects ADMM penalty parameters. For example, [22] reported unstable convergence behaviors of PnP-ADMM with fixed ADMM parameters. To moderate this problem, [41] proposed a scheme that adaptively controls the ADMM parameters based on relative residuals. Similar to the residual balancing technique [42, §3.4.1], the scheme in [41] requires tuning initial parameters. Regularization by Denoising (RED [22]) is an alternative that moderates some such limitations. In particular, RED aims to make a clear connection between optimization and a denoiser $\mathcal{D}$, by defining its prior term by (scaled) $x^T(x-\mathcal{D}(x))$. Nonetheless, [43] showed that many practical denoisers do not satisfy the Jacobian symmetry in [22], and proposed a less restrictive method, score-matching by denoising.

The convergence analysis of the INN inspired by the relaxed projected gradient descent (RPGD) method in [31] has the least restrictive conditions on the regression NN among the existing INNs. This method replaces the projector of a projected gradient descent method with an image refining NN. However, the RPGD-inspired INN directly applies an image refining NN to gradient descent updates of data-fit; thus, this INN relies heavily on the mapping performance of a refining NN and can have overfitting risks, similar to non-MBIR regression NNs, e.g., FBPConvNet [12]. In addition, it exploits the data-fit term only for the first few iterations [31, Fig. 5(c)]. We refer the perspective used in RPGD-inspired INN and its related works [26], [44] as "hard-refiner": different from soft-refiners, these methods do not use a refining NN as a regularizer. More recently, [26] presented convergence analysis for an INN inspired by a proximal gradient descent method. However, their analysis is based on noiseless measurements, which is typically impractical.

Broadly speaking, existing convergence analysis largely depends on the (firmly) nonexpansive property of image refining NNs [22], [23], [43], [31, PGD], [26]. However, except for a single-hidden layer convolutional NN (CNN), it is yet unclear which analytical conditions guarantee the non-expansiveness of general refining NNs [27]. To guarantee convergence of INNs even when using possibly *expansive* image refining NNs, we proposed a method that normalizes the output signal of image refining NNs by their Lipschitz constants [27]. However, if one uses expansive NNs that are identical across iterations, it is difficult to obtain "best" image recovery with that normalization scheme. The spectral normalization based training [45], [46] can ensure the non-expansiveness of refining NNs by single-step power iteration. However, similar to the normalization method in [27], refining NNs trained with the spectral normalization method [46] degraded the image reconstruction accuracy for an INN using iteration-wise refining NNs [19]. In addition, there does not yet exist theoretical convergence results when refining NNs change across iterations, yet iteration-wise refining NNs are widely studied [20], [21], [25], [28]. Finally, existing analysis considers only a narrow class of data-fit terms: most analyses consider a quadratic function with a linear imaging model [26], [31] or more generally, a convex cost function [23], [43], [46] that can be minimized with a practical closed-form solution. No theoretical convergence results exist for *general* (non)convex data-fit terms, iteration-wise NN denoisers, and a general set of feasible points.

1.4 Challenges in existing INNs: Acceleration

Compared to non-MBIR regression NNs that do not exploit the data-fit $f(x;y)$ in (P0), INNs require more computation because they consider the imaging physics. Computation increases as the imaging system or image formation model becomes larger-scale, e.g., LF photography from a focal stack, 3D CT, parallel MRI using many receive coils, and image super-resolution. Thus, acceleration becomes crucial for INNs.

First, consider the existing methods motivated by ADMM or block coordinate descent (BCD) method: examples include PnP-ADMM [23], [41], RED-ADMM [22], [43], MoDL [30], BCD-Net [16], [18], [25], etc. These methods can require multiple inner iterations to balance data-fit and prior information estimated by trained refining NNs, increasing total MBIR time. For example, in solving such problems, each outer iteration involves $x^{(i+1)} = \operatorname{argmin}_x F(x;y,z^{(i+1)})$, where F is given as in (P0) and $z^{(i+1)}$ is the output from the ith image refining NN. For LF imaging system using a focal stack data [47], solving the above problem requires multiple iterations, and the total computational cost scale with the numbers of photosensors and sub-aperture images. In addition, nonconvexity of the data-fit term $f(x;y)$ can break convergence guarantees of these methods, because in general, the proximal mapping $\operatorname{argmin}_x f(x;y)+\gamma\|x-z^{(i+1)}\|_2^2$ is no longer nonexpansive.

Second, consider the existing works motivated by gradient descent methods [21], [26], [28], [31]. These methods resolve the inner iteration issue; however, they lack a sophisticated step-size control or backtracking scheme that influences convergence guarantee and acceleration. Accelerated proximal gradient (APG) methods using *momentum* terms can significantly accelerate convergence rates for solving composite convex problems [48], [49], so we expect that INN methods in the second class have yet to be maximally accelerated. The work in [44] applied PnP to the APG method

[49]; [50] applied PnP to the primal-dual splitting (PDS) algorithm [51]. However, similar to RPGD [31], these are hard-refiner methods using some state-of-the-art denoisiers (e.g., BM3D [52]) but not trained NNs. Those methods lack convergence analyses and guarantees may be limited to convex data-fit function.

1.5 Contributions and organization of the paper

This paper proposes *Momentum-Net*, the first INN architecture that aims for fast and convergent MBIR. The architecture of Momentum-Net is motivated by applying the Block Proximal Extrapolated Gradient method using a Majorizer (BPEG-M) [24], [35] to MBIR using trainable convolutional autoencoders [24], [25], [37]. Specifically, each iteration of Momentum-Net consists of three core modules: image refining, extrapolation, and MBIR. At each Momentum-Net iteration, an extrapolation module uses *momentum* from previous updates to amplify the changes in subsequent iterations and accelerate convergence, and an MBIR module is *noniterative*. In addition, Momentum-Net resolves the convergence issues mentioned in §1.3: for general differentiable (non)convex data-fit terms and convex feasible sets, it guarantees convergence to a point that satisfies fixed-point and critical point conditions, under some mild conditions and two asymptotic conditions, i.e., *asymptotically nonexpansive paired refining NNs* and *asymptotically block-coordinate minimizer*.

The remainder of this paper is organized as follows. §2 constructs the Momentum-Net architecture motivated by BPEG-M algorithm that solves MBIR problem using a learnable convolutional regularizer, describes its relation to existing works, analyzes its convergence, and summarizes the benefits of Momentum-Net over existing INNs. §3 provides details of training INNs, including image refining NN architectures, single-hidden layer or "shallow" CNN (sCNN) and multi-hidden layer or "deep" CNN (dCNN), and training loss function, and proposes a regularization parameter selection scheme to consider data-fit variations across training and testing samples. §4 considers two extreme imaging applications: sparse-view CT and LF photography using a focal stack. §4 reports numerical experiments of applications where the proposed Momentum-Net using extrapolation significantly improves MBIR speed and accuracy, over the existing INNs, BCD-Net [22], [25], [30], Momentum-Net using *no extrapolation* [21], [28], ADMM-Net [20], [23], [41], and PnP-PDS [50] using refining NNs. Furthermore, §4 reports numerical experiments where Momentum-Net significantly improves reconstruction quality compared to a state-of-the-art MBIR method in each application.

2 Momentum-Net: Where BPEG-M meets NNs for inverse problems

2.1 Motivation: BPEG-M algorithm for MBIR using learnable convolutional regularizer

This section motivates the proposed Momentum-Net architecture, based on our previous works [24], [37]. Consider the following approach for recovering signal x from measurements y (see the setup of block multi-(non)convex problems in §A.1.1):

$$\underset{x\in\mathcal{X}}{\operatorname{argmin}}\ f(x;y)+\gamma\Big(\min_{\{\zeta_k\}} r(x,\{\zeta_k\};\{h_k\})\Big),$$
$$r(x,\{\zeta_k\};\{h_k\}) \triangleq \sum_{k=1}^{K}\frac{1}{2}\|h_k * x-\zeta_k\|_2^2+\beta_k\|\zeta_k\|_1, \quad (1)$$

where $\mathcal{X}$ is a closed set, $f(x;y)+\gamma r(x,\{\zeta_k\};\{h_k\})$ is a (continuosly) differentiable (non)convex function in x, $\min_{\{\zeta_k\}} r(x,\{\zeta_k\};\{h_k\})$ is a learnable convolutional regularizer [24], [36], $\{\zeta_k : k=1,\dots,K\}$ is a set of sparse features that correspond to $\{h_k * x\}$, $\{h_k \in \mathbb{C}^R : k=1,\dots,K\}$ is a set of trainable filters, and R and K denote the size and number of trained filters, respectively.

Problem (1) can be viewed as a two-block optimization problem in terms of the image x and the features $\{\zeta_k\}$. We solve (1) using the recent BPEG-M optimization framework [24], [35] that has attractive convergence guarantee and rapidly solved several block optimization problems [24], [35], [53]–[55]. BPEG-M has the following key ideas for each block optimization problem (see details in §A.1):

- M_b-Lipschitz continuity for the gradient of the bth block optimization problem, $\forall b$:

  Definition 1 (M-Lipschitz continuity [24])**.** *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*

  $$\|g(u)-g(v)\|_{M^{-1}} \le \|u-v\|_M, \quad \forall u,v\in\mathbb{R}^n.$$

  Definition 1 is a more general concept than the classical Lipschitz continuity.
- A sharper majorization matrix M that gives a tighter bound in Definition 1 leads to a tighter quadratic majorization bound in the following lemma:

  Lemma 2 (Quadratic majorization via M-Lipschitz continuous gradients [24])**.** *Let $f(u):\mathbb{R}^n\to\mathbb{R}$. If ∇f is M-Lipschitz continuous, then*

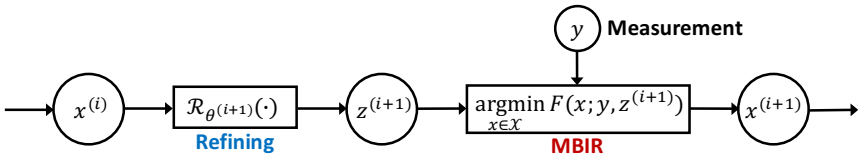
  $$f(u)\le f(v)+\langle\nabla_u f(v),u-v\rangle+\frac{1}{2}\|u-v\|_M^2, \quad \forall u,v\in\mathbb{R}^n.$$

  Having tighter majorization bounds, sharper majorization matrices tend to accelerate BPEG-M convergence.
- The majorized block problems are "proximable", i.e., proximal mapping of majorized function is "easily" computable depending on the properties of bth block majorizer and regularizer, M_b and r_b, where the proximal mapping operator is defined by

  $$\operatorname{Prox}_{r_b}^{M_b}(z) \triangleq \underset{u}{\operatorname{argmin}}\ \frac{1}{2}\|u-z\|_{M_b}^2+r_b(u), \quad \forall b. \quad (2)$$

- Block-wise extrapolation and momentum terms to accelerate convergence.

Suppose that *1)* gradient of $f(x;y)+\gamma r(x,\{\zeta_k\};\{h_k\})$ is M-Lipschitz continuous at an extrapolated point $\acute{x}^{(i+1)}$, $\forall i$; *2)* filters in (1) satisfy the tight-frame (TF) condition, $\sum_{k=1}^{K}\|h_k * u\|_2^2 = \|u\|_2^2$, $\forall u$, for some boundary conditions



(b) BCD-Net [25]

Fig. 1. Architectures of different INNs for MBIR. (a–b) The architectures of Momentum-Net and BCD-Net [25] are constructed by generalizing BPEG-M and BCD algorithms that solve MBIR problem using a convolutional regularizer trained via *convolutional analysis operator learning* (CAOL) [24], [36], respectively. (a) Removing extrapolation modules (i.e., setting the extrapolation matrices $\{E^{(i+1)}:\forall i\}$ as a zero matrix), Momentum-Net specializes to the existing gradient-descent-inspired INNs [21], [28]. When the MBIR cost function $F(x;y,z^{(i+1)})$ in (P1) has a sharp majorizer $\widetilde{M}^{(i+1)}$, $\forall i$, Momentum-Net (using $\rho\approx 1$) specializes to BCD-Net; see Examples 5–6. (b) BCD-Net is a general version of the existing INNs in [20], [22], [23], [30], [39]–[41] by using iteration-wise image refining NNs, i.e., $\{\mathcal{R}_{\theta^{(i+1)}}:\forall i\}$, or considering general convex data-fit $f(x;y)$.

[24]. Applying the BPEG-M framework (see Algorithm A.1) to solving (1) leads to the following block updates:

$$z^{(i+1)} = \sum_{k=1}^{K} \operatorname{flip}(h_k^*) * \mathcal{T}_{\beta_k}(h_k * x^{(i)}), \tag{3}$$

$$\acute{x}^{(i+1)} = x^{(i)} + E^{(i+1)}\big(x^{(i)} - x^{(i-1)}\big), \tag{4}$$

$$x^{(i+1)} = \operatorname{Prox}_{\mathbb{I}_{\mathcal{X}}}^{\widetilde{M}^{(i+1)}}\Big(\acute{x}^{(i+1)} - \big(\widetilde{M}^{(i+1)}\big)^{-1}\nabla F(\acute{x}^{(i+1)};y,z^{(i+1)})\Big), \tag{5}$$

where $E^{(i+1)}$ is an extrapolation matrix that is given in (8) or (9) below, $\widetilde{M}^{(i+1)}$ is a (scaled) majorization matrix for $\nabla F(x;y,z^{(i+1)})$ that is given in (7) below, $\forall i$, the proximal operator $\operatorname{Prox}_{\mathbb{I}_{\mathcal{X}}}^{\widetilde{M}^{(i+1)}}(\cdot)$ in (5) is given by (2), and $\mathbb{I}_{\mathcal{X}}(x)$ is the characteristic function of set $\mathcal{X}$ (i.e., $\mathbb{I}_{\mathcal{X}}$ equals to 0 if $x\in\mathcal{X}$, and ∞ otherwise).

Proximal mapping update (3) has a *single-hidden layer convolutional autoencoder* architecture that consists of encoding convolution, nonlinear thresholding, and decoding convolution, where $\operatorname{flip}(\cdot)$ flips a filter along each dimension, and the soft-thresholding operator $\mathcal{T}_\alpha(u):\mathbb{C}^N\to\mathbb{C}^N$ is defined by

$$(\mathcal{T}_\alpha(u))_n \triangleq \begin{cases} u_n - \alpha\cdot\operatorname{sign}(u_n), & |u_n|>\alpha, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

for $n=1,\dots,N$, in which $\operatorname{sign}(\cdot)$ is the sign function. See details of deriving BPEG-M updates (3)–(5) in §A.1.4. The BPEG-M updates in (3)–(5) guarantee convergence to a critical point, when MBIR problem (1) satisfies some mild conditions, e.g., lower-boundedness and existence of critical points; see Assumption S.1 in §A.1.3.

The following section generalizes the BPEG-M updates in (3)–(5) and constructs the Momentum-Net architecture.

2.2 Architecture

This section establishes the INN architecture of *Momentum-Net* by generalizing BPEG-M updates (3)–(5) that solve (1). Specifically, we replace the proximal mapping in (3) with a general image refining NN $\mathcal{R}_\theta(\cdot)$, where θ denotes the trainable parameters. To effectively remove iteration-wise artifacts and give "best" signal estimates at each iteration, we further generalize a refining NN $\mathcal{R}_\theta(\cdot)$ to *iteration-wise* image refining NNs $\{\mathcal{R}_{\theta^{(i+1)}}(\cdot): i=0,\dots,N_{\text{iter}}-1\}$, where $\theta^{(i+1)}$ denotes the parameters for the ith iteration refining NN $\mathcal{R}_{\theta^{(i+1)}}$, and N_{iter} is the number of Momentum-Net iterations. The iteration-wise NNs are particularly useful for reducing overfitting risks in regression, because $\mathcal{R}_{\theta^{(i+1)}}$ is responsible for removing noise features only at the ith iteration, and thus one does not need to greatly increase dimensions of its parameter $\theta^{(i+1)}$ [16], [18]. In low-dose CT reconstruction, for example, the refining NNs at the early and later iterations remove streak artifacts and Gaussian-like noise, respectively [16].

Algorithm 1 Momentum-Net

Require: $\{\mathcal{R}_{\theta^{(i)}}: i=1,\dots,N_{\text{iter}}\}$, $\rho\in(0,1)$, $\gamma>0$, $x^{(0)}=x^{(-1)}$, y

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

Calculate $\widetilde{M}^{(i+1)}$ by (7), and $E^{(i+1)}$ by (8) or (9)

Image refining:

$$z^{(i+1)} = (1-\rho)x^{(i)} + \rho\mathcal{R}_{\theta^{(i+1)}}\big(x^{(i)}\big) \tag{Alg.1.1}$$

Extrapolation:

$$\acute{x}^{(i+1)} = x^{(i)} + E^{(i+1)}\big(x^{(i)} - x^{(i-1)}\big) \tag{Alg.1.2}$$

MBIR:

$$x^{(i+1)} = \operatorname{Prox}_{\mathbb{I}_{\mathcal{X}}}^{\widetilde{M}^{(i+1)}}\Big(\acute{x}^{(i+1)} - \big(\widetilde{M}^{(i+1)}\big)^{-1}\nabla F(\acute{x}^{(i+1)};y,z^{(i+1)})\Big) \tag{Alg.1.3}$$

end for

Each iteration of Momentum-Net consists of *1)* image refining, *2)* extrapolation, and *3)* MBIR modules, corresponding to the BPEG-M updates (3), (4), and (5), respectively. See the architecture of Momentum-Net in Fig. 1(a) and Algorithm 1. At the ith iteration, Momentum-Net performs the following three processes:

- *Refining*: The ith image refining module gives the "refined" image $z^{(i+1)}$, by applying the ith refining NN, $\mathcal{R}_{\theta^{(i+1)}}$, to an input image at the ith iteration, $x^{(i)}$ (i.e., image estimate from the $(i-1)$th iteration). Different from existing INNs, e.g., ADMM-Net [20], PnP-ADMM [23], [41], RED [22], MoDL [30], BCD-Net [25] (see Fig. 1(b)), TNRD [21], [28], we apply ρ-relaxation with $\rho \in (0, 1)$; see (Alg.1.1). The parameter ρ controls the strength of inference from refining NNs, but does not affect the convergence guarantee of Momentum-Net. Proper selection of ρ can improve MBIR accuracy (see §A.10).
- *Extrapolation*: The ith extrapolation module gives the extrapolated point $\acute{x}^{(i+1)}$, based on *momentum* terms $x^{(i)} - x^{(i-1)}$; see (Alg.1.2). Intuitively speaking, momentum is information from previous updates to amplify the changes in subsequent iterations. Its effectiveness has been shown in diverse optimization literature, e.g., convex optimization [48], [49] and block optimization [24], [35].
- *MBIR*: Given a refined image $z^{(i+1)}$ and a measurement vector y, the ith MBIR module (Alg.1.3) applies the proximal operator $\text{Prox}_{\mathbb{I}_{\mathcal{X}}}^{\widetilde{M}^{(i+1)}}(\cdot)$ to the *extrapolated gradient update using a quadratic majorizer* of $F(x; y, z^{(i+1)})$, where F is defined in (P0). Intuitively speaking, this step solves a *majorized* version of the following MBIR problem at the extrapolated point $\acute{x}^{(i+1)}$:

$$\min_{x \in \mathcal{X}} F(x; y, z^{(i+1)}), \tag{P1}$$

  and gives a reconstructed image $x^{(i+1)}$. In Momentum-Net, we consider (non)convex differentiable MBIR cost functions F with M-Lipschitz continuous gradients, and a convex and closed set $\mathcal{X}$. For a wide range of large-scale inverse imaging problems, the majorized MBIR problem (Alg.1.3) has a practical closed-form solution and thus, does not require an iterative solver, depending on the properties of practically invertible majorization matrices $M^{(i+1)}$ and constraints. Examples of $M^{(i+1)}$-$\mathcal{X}$ combinations that give a noniterative solution for (Alg.1.3) include scaled identity and diagonal matrices with a box constraint and the non-negativity constraint, and matrices decomposable by unitary transforms, e.g., a circulant matrix [56], [57], with $\mathcal{X} = \mathbb{C}^N$. The updated image $x^{(i+1)}$ is the input to the next Momentum-Net iteration.

The followings are details of Momentum-Net in Algorithm 1. A scaled majorization matrix is

$$\widetilde{M}^{(i+1)} = \lambda \cdot M^{(i+1)} \succ 0, \qquad \lambda \geq 1, \tag{7}$$

where $M^{(i+1)} \in \mathbb{R}^{N \times N}$ is a symmetric positive definite majorization matrix of $\nabla F(x; y, z^{(i+1)})$ in the sense of M-Lipschitz continuity (see Definition 1). In (7), $\lambda = 1$ and $\lambda > 1$ for convex and nonconvex $F(x; y, z^{(i+1)})$ (or convex and nonconvex $f(x; y)$), respectively. We design the extrapolation matrices as follows:

for *convex* F,

$$E^{(i+1)} = \delta^2 m^{(i)} \cdot \left(M^{(i+1)}\right)^{-\frac{1}{2}} \left(M^{(i)}\right)^{\frac{1}{2}}; \tag{8}$$

for *nonconvex* F,

$$E^{(i+1)} = \delta^2 m^{(i)} \cdot \frac{\lambda - 1}{2(\lambda + 1)} \cdot \left(M^{(i+1)}\right)^{-\frac{1}{2}} \left(M^{(i)}\right)^{\frac{1}{2}}, \tag{9}$$

for some $\delta < 1$ and $\{0 \leq m^{(i)} \leq 1 : \forall i\}$. We update the momentum coefficients $\{m^{(i+1)} : \forall i\}$ by the following formula [24], [35]:

$$m^{(i+1)} = \frac{\theta^{(i)} - 1}{\theta^{(i+1)}}, \qquad \theta^{(i+1)} = \frac{1 + \sqrt{1 + 4(\theta^{(i)})^2}}{2}; \tag{10}$$

if $F(x; y, z^{(i+1)})$ has a sharp majorizer, i.e., $\nabla F(x; y, z^{(i+1)})$ has $M^{(i+1)}$ such that the corresponding bound in Definition 1 is tight, then we set $m^{(i+1)} = 0$, $\forall i$. §A.11 lists parameters of Momentum-Net, and summarizes selection guidelines or gives default values.

2.3 Relations to previous works

Several existing MBIR methods can be viewed as a special case of Momentum-Net:

Example 3. (MBIR model (1) using convolutional autoencoders that satisfy the TF condition [24]). The BPEG-M updates in (3)–(5) are special cases of the modules in Momentum-Net (Algorithm 1), with $\{\mathcal{R}_{\theta^{(i+1)}}(\cdot) = \sum_{k=1}^{K} \text{flip}(h_k^*) * \mathcal{T}_{\beta_k}(h_k * (\cdot)) : \forall i\}$ (i.e., single hidden-layer convolutional autoencoder [24]) and $\rho \approx 1$. These give a clear mathematical connection between a denoiser (3) and cost function (1). One can find a similar relation between a multi-hidden layer CNN and a multi-layer convolutional regularizer [24, Appx.].

Example 4. (INNs inspired by gradient descent method, e.g., TNRD [21], [28]). Removing extrapolation modules, i.e., setting $\{E^{(i+1)} = 0 : \forall i\}$ in (Alg.1.2), and setting $\rho \approx 1$, Momentum-Net becomes the existing INN in [21], [28].

Example 5. (BCD-Net for image denoising [25]). To obtain a clean image $x \in \mathbb{R}^N$ from a noisy image $y \in \mathbb{R}^N$ corrupted by an additive white Gaussian noise (AWGN), MBIR problem (P1) considers the data-fit $f(x; y) = \frac{1}{2}\|y - x\|_W^2$ with the inverse covariance matrix $W = \frac{1}{\sigma^2} I$, where σ^2 is a variance of AWGN, and the box constraint $\mathcal{X} = [0, U]^N$ with an upper bound $U > 0$. For this $f(x; y)$, the MBIR module (Alg.1.3) can use the exact majorizer $\{\widetilde{M}^{(i+1)} = (\frac{1}{\sigma^2} + \gamma) I\}$ and one does not need to use the extrapolation module (Alg.1.2), i.e., $\{E^{(i+1)} = 0\}$. Thus, Momentum-Net (with $\rho \approx 1$) becomes BCD-Net.

Example 6. (BCD-Net for undersampled single-coil MRI [25]). To obtain an object magnetization $x \in \mathbb{R}^N$ from a k-space data $y \in \mathbb{C}^m$ obtained by undersampling (e.g., compressed sensing [58]) MRI, MBIR problem (P1) considers the data-fit $f(x; y) = \frac{1}{2}\|y - Ax\|_W^2$ with an undersampling Fourier operator A (disregarding relaxation effects and considering Cartesian k-space), the inverse covariance matrix $W = \frac{1}{\sigma^2} I$, where σ^2 is a variance of complex AWGN [59], and $\mathcal{X} = \mathbb{C}^N$. For this $f(x; y)$, the MBIR module (Alg.1.3) can use the exact majorizer $\{\widetilde{M}^{(i+1)} = F_{\text{disc}}^H (\frac{1}{\sigma^2} P + \gamma I) F_{\text{disc}}\}$ that is practically invertible, where F_{disc} is the discrete Fourier transform and P is a diagonal matrix with either 0 or 1 (their positions correspond to sampling pattern in k-space), and the extrapolation module (Alg.1.2) uses the zero extrapolation matrices $\{E^{(i+1)} = 0\}$. Thus, Momentum-Net (with $\rho \approx 1$) becomes BCD-Net.

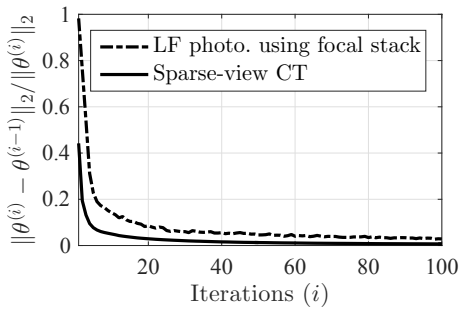


Fig. 2. Convergence behavior of Momentum-Net's dCNNs refiners $\{\mathcal{R}_{\theta^{(i)}}\}$ in different applications ($\theta^{(i)}$ denotes the parameter vector of the ith iteration refiner $\mathcal{R}_{\theta^{(i)}}$, for $i = 1, \dots, N_{\text{iter}}$; see details of $\{\mathcal{R}_{\theta^{(i)}}\}$ in (19) and §4.2.1; $N_{\text{iter}} = 100$). Sparse-view CT (fan-beam geometry with 12.5% projections views): $\mathcal{R}_{\theta^{(i)}}$ quickly converges, where majorization matrices of training data-fits have similar condition numbers. LF photography using a focal stack (five detectors and reconstructed LFs consists of 9×9 sub-aperture images): $\mathcal{R}_{\theta^{(i)}}$ has slower convergence, where majorization matrices of training data-fits have largely different condition numbers.

The following section analyzes the convergence of Momentum-Net.

2.4 Convergence analysis

In practice, INNs, i.e., "unrolled" or PnP methods using refining NNs, are trained and used with a specific number of iterations. Nevertheless, similar to optimization algorithms, studying convergence properties of INNs with $N_{\text{iter}} \to \infty$ [23], [31], [46] is important; in particular, it is crucial to know if a given INN tends to converge as N_{iter} increases. For INNs using iteration-wise refining NNs, e.g., BCD-Net [25] and proposed Momentum-Net, we expect that refiners converge, i.e., their image refining capacity converges, because information provided by data-fit function $f(x; y)$ in MBIR (e.g., likelihood) reaches some "bound" after a certain number of iterations. Fig. 2 illustrates that dCNN parameters of Momentum-Net tend to converge for different applications. (The similar behavior was reported for sCNN refiners in BCD-Net [16].) Although refiners do not *completely* converge, in practice, one could use a refining NN at a sufficiently large iteration number, e.g., $N_{\text{iter}} = 100$ in Momentum-Net, for the later iterations.

There are two key challenges in analyzing the convergence of Momentum-Net in Algorithm 1: both challenges relate to its image refining modules (Alg.1.1). First, image refining NNs $\mathcal{R}_{\theta^{(i+1)}}$ change across iterations; even if they are identical across iterations, they are not necessarily nonexpansive operators [60], [61] in practice. Second, the iteration-wise refining NNs are not necessarily proximal mapping operators, i.e., they are not written explicitly in the form of (2). This section proposes two new asymptotic definitions to overcome these challenges, and then uses those conditions to analyze convergence properties of Momentum-Net in Algorithm 1.

2.4.1 Preliminaries

To resolve the challenge of iteration-wise refining NNs and the practical difficulty in guaranteeing their non-expansiveness, we introduce the following generalized definition of the non-expansiveness [60], [61].

Definition 7 (Asymptotically nonexpansive paired operators)**.** *A sequence of paired operators $(\mathcal{R}_{\theta^{(i)}}, \mathcal{R}_{\theta^{(i+1)}})$ is asymptotically nonexpansive if there exist a summable nonnegative sequence $\{\epsilon^{(i+1)} \geq 0 : \sum_{i=0}^{\infty} \epsilon^{(i+1)} < \infty\}$ such that*[1]

$$\|\mathcal{R}_{\theta^{(i+1)}}(u) - \mathcal{R}_{\theta^{(i)}}(v)\|_2^2 \leq \|u - v\|_2^2 + \epsilon^{(i+1)}, \qquad \forall u, v, i. \tag{11}$$

When $\mathcal{R}_{\theta^{(i+1)}} = \mathcal{R}_\theta$ and $\epsilon^{(i+1)} = 0$, $\forall i$, Definition 7 becomes the standard non-expansiveness of a mapping operator $\mathcal{R}_\theta$. If we replace the inequality ($\leq$) with the strict inequality ($<$) in (11), then we say that the sequence of paired operators $(\mathcal{R}_{\theta^{(i+1)}}, \mathcal{R}_{\theta^{(i+1)}})$ is asymptotically contractive. (This stronger assumption is used to prove convergence of BCD-Net in Proposition A.5.) Definition 7 also implies that mapping operators $\mathcal{R}_{\theta^{(i+1)}}$ converge to some nonexpansive operator, if the corresponding parameters $\theta^{(i+1)}$ converge.

Definition 7 incorporates a pairing property because Momentum-Net uses iteration-wise image refining NNs. Specifically, the pairing property helps prove convergence of Momentum-Net, by connecting image refining NNs at adjacent iterations. Furthermore, the asymptotic property in Definition 7 allows Momentum-Net to use *expansive* refining NNs (i.e., mapping operators having a Lipschitz constant larger than 1) for some iterations, while guaranteeing convergence; see Figs. 3(a3) and 3(b3). Suppose that refining NNs are identical across iterations, i.e., $\mathcal{R}_{\theta^{(i+1)}} = \mathcal{R}_\theta$, $\forall i$, similar to some existing INNs, e.g., PnP [23], RED [22], and other methods in §1.3. In such cases, if $\mathcal{R}_\theta$ is expansive, Momentum-Net may diverge; this property corresponds to the limitation of existing methods described in §1.3. Momentum-Net moderates this issue by using iteration-wise refining NNs that satisfy the asymptotic paired non-expansiveness in Definition 7.

Because the sequence $\{z^{(i+1)} : \forall i\}$ in (Alg.1.1) is not necessarily updated with a proximal mapping, we introduce a generalized definition of block-coordinate minimizers [53, (2.3)] for $z^{(i+1)}$-updates:

Definition 8 (Asymptotic block-coordinate minimizer)**.** *The update $z^{(i+1)}$ is an asymptotic block-coordinate minimizer if there exists a summable nonnegative sequence $\{\Delta^{(i+1)} \geq 0 : \sum_{i=0}^{\infty} \Delta^{(i+1)} < \infty\}$ such that*

$$\left\| z^{(i+1)} - x^{(i)} \right\|_2^2 \leq \left\| z^{(i)} - x^{(i)} \right\|_2^2 + \Delta^{(i+1)}, \qquad \forall i. \tag{12}$$

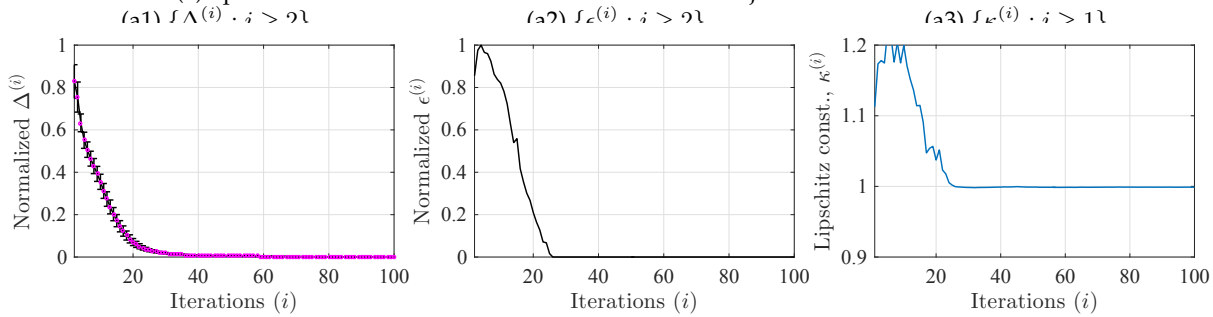
Definition 8 implies that as $i \to \infty$, the updates $\{z^{(i+1)} : i \geq 0\}$ approach a block-coordinate minimizer trajectory that satisfies (12) with $\{\Delta^{(i+1)} = 0 : i \geq 0\}$. In particular, $\Delta^{(i+1)}$ quantifies how much the update $z^{(i+1)}$ in (Alg.1.1) perturbs a block-coordinate minimizer trajectory. The bound $\|z^{(i+1)} - x^{(i)}\|_2^2 \leq \|z^{(i)} - x^{(i)}\|_2^2$ always holds, $\forall i$, when one uses the proximal mapping in (3) within the BPEG-M framework. Note that while applying trained Momentum-Net, (12) is easy to examine empirically, whereas (11) is harder to check.

2.4.2 Assumptions

This section introduces and interprets the assumptions for convergence analysis of Momentum-Net in Algorithm 1:

1. One could replace the bound in (11) with $\|\mathcal{R}_{\theta^{(i+1)}}(u) - \mathcal{R}_{\theta^{(i)}}(v)\|_2^2 \leq (1 + \epsilon^{(i+1)})\|u - v\|_2^2$ (and summable $\{\epsilon^{(i+1)} : \forall i\}$), and the proofs for our main arguments go through.

(a) Sparse-view CT: Condition numbers of data-fit majorizers have *mild* variations.



(b) LF photography using a focal stack: Condition numbers of data-fit majorizers have *large* variations.

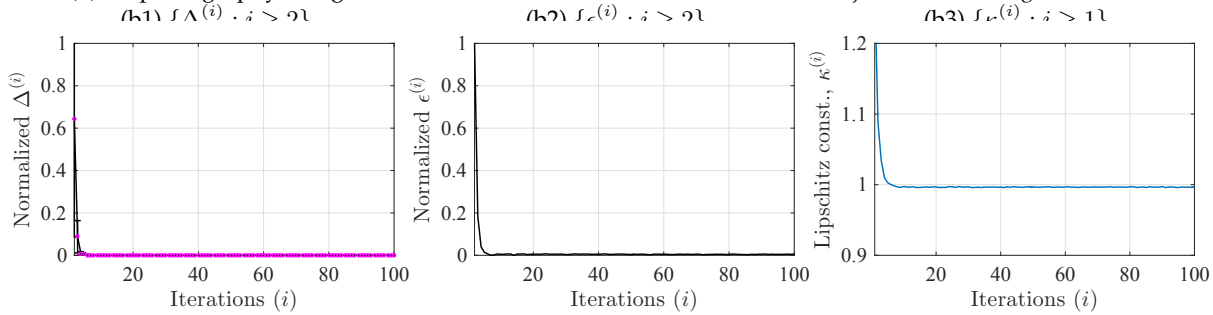


Fig. 3. Empirical measures related to Assumption 4 for guaranteeing convergence of Momentum-Net using *dCNNs* refiners (for details, see (19) and §4.2.1), in different applications. See estimation procedures in §A.2. (a) The sparse-view CT reconstruction experiment used fan-beam geometry with 12.5% projections views. (b) The LF photography experiment used five detectors and reconstructed LFs consisting of 9×9 sub-aperture images. (a1, b1) For both the applications, we observed that $\Delta^{(i)} \to 0$. This implies that the $z^{(i+1)}$-updates in (Alg.1.1) satisfy the asymptotic block-coordinate minimizer condition in Assumption 4. (Magenta dots denote the mean values and black vertical error bars denote standard deviations.) (a2) Momentum-Net trained from training data-fits, where their majorization matrices have *mild* condition number variations, shows that $\epsilon^{(i)} \to 0$. This implies that paired NNs $(\mathcal{R}_{\theta^{(i+1)}}, \mathcal{R}_{\theta^{(i)}})$ in (Alg.1.1) are asymptotically nonexpansive. (b2) Momentum-Net trained from training training data-fits, where their majorization matrices have *mild* condition number variations, shows that $\epsilon^{(i)}$ becomes close to zero, but does not converge to zero in one hundred iterations. (a3, b3) The NNs, $\mathcal{R}_{\theta^{(i+1)}}$ in (Alg.1.1), become nonexpansive, i.e., its Lipschitz constant $\kappa^{(i)}$ becomes less than 1, as i increases.

- *Assumption 1)* In MBIR problems (P1), (non)convex $F(x; y, z^{(i+1)})$ is (continuously) differentiable, proper, and lower-bounded in $\mathrm{dom}(F)$,[2] $\forall i$, and $\mathcal{X}$ is convex and closed. Algorithm 1 has a fixed-point.
- *Assumption 2)* $\nabla F(x; y, z^{(i+1)})$ is $M^{(i+1)}$-Lipschitz continuous with respect to x (see Definition 1), where $M^{(i+1)}$ is a iteration-wise majorization matrix that satisfies $m_{F,\min} I_N \preceq M^{(i+1)} \preceq m_{F,\max} I_N$ with $0 < m_{F,\min} \leq m_{F,\max} < \infty$, $\forall i$.
- *Assumption 3)* The extrapolation matrices $E^{(i+1)} \succeq 0$ in (8)–(9) satisfy the following conditions:

  for *convex* F,

  $$\left(E^{(i+1)}\right)^T M^{(i+1)} E^{(i+1)} \preceq \delta^2 \cdot M^{(i)}, \qquad \delta < 1; \tag{13}$$

  for *nonconvex* F,

  $$\left(E^{(i+1)}\right)^T M^{(i+1)} E^{(i+1)} \preceq \frac{\delta^2 (\lambda - 1)^2}{4(\lambda + 1)^2} \cdot M^{(i)}, \qquad \delta < 1. \tag{14}$$

- *Assumption 4)* The sequence of paired operators $(\mathcal{R}_{\theta^{(i+1)}}, \mathcal{R}_{\theta^{(i)}})$ is asymptotically nonexpansive with a summable sequence $\{\epsilon^{(i+i)} \geq 0\}$; the update $z^{(i+1)}$ is an asymptotic block-coordinate minimizer with a summable sequence $\{\Delta^{(i+i)} \geq 0\}$. The mapping functions $\{\mathcal{R}_{\theta^{(i+1)}} : \forall i\}$ are continuous with respect to input points and the corresponding parameters $\{\theta^{(i+1)} : \forall i\}$ are bounded.

2. $F : \mathbb{R}^n \to (-\infty, +\infty]$ is proper if $\mathrm{dom} F \neq \emptyset$. F is lower bounded in $\mathrm{dom}(F) \triangleq \{u : F(u) < \infty\}$ if $\inf_{u \in \mathrm{dom}(F)} F(u) > -\infty$.

Assumption 1 is a slight modification of Assumption S.1 of BPEG-M, and Assumptions 2–3 are identical to Assumptions S.2–S.3 of BPEG-M; see Assumptions S.1–S.3 in §A.1.3. The extrapolation matrix designs (8) and (9) satisfy conditions (13) and (14) in Assumption 3, respectively.

We provide empirical justifications for the first two conditions in Assumption 4. First, Figs. 3(a2) and A.1(a2) illustrate that paired refining NNs $(\mathcal{R}_{\theta^{(i+1)}}, \mathcal{R}_{\theta^{(i)}})$ of Momentum-Net appear to be asymptotically nonexpansive in an application that has mild condition number variations across training data-fit majorization matrices. Figs. 3(a3), 3(b3), A.1(a3), and A.1(b3) illustrate for different applications that refining NNs $\{\mathcal{R}_{\theta^{(i+1)}}\}$ become nonexpansive: their Lipschitz constants at the first several iterations are larger than 1, and their Lipschitz constants in later iterations become less than 1. Alternatively, the asymptotic non-expansiveness of paired operators $(\mathcal{R}_{\theta^{(i+1)}}, \mathcal{R}_{\theta^{(i)}})$ can be satisfied by a stronger assumption that the sequence $\{\mathcal{R}_{\theta^{(i+1)}}\}$ converges to some nonexpansive operator. (Fig. 2 illustrates that dCNN parameters of Momentum-Net appear to converge.)

Figs. 3(a3), 3(b3), A.1(a3), and A.1(b3) illustrate for different applications that the $z^{(i+1)}$-updates are asymptotic block-coordinate minimizers. Lemma A.4 and §A.3 in the appendices provide a *probabilistic* justification for the asymptotic block-coordinate minimizer condition.

2.4.3 Main convergence results

This section analyzes fixed-point and critical point convergence of Momentum-Net in Algorithm 1, under the assumptions in the previous section. We first show that differences between two consecutive iterates generated by Momentum-Net converge to zero:

Proposition 9 (Convergence properties)**.** *Under Assumptions 1–4, let* $\{x^{(i+1)}, z^{(i+1)} : i \geq 0\}$ *be the sequence generated by Algorithm 1. Then, the sequence satisfies*

$$\sum_{i=0}^{\infty} \left\| \begin{bmatrix} x^{(i+1)} \\ z^{(i+1)} \end{bmatrix} - \begin{bmatrix} x^{(i)} \\ z^{(i)} \end{bmatrix} \right\|_2^2 < \infty, \tag{15}$$

and hence $\left\| \begin{bmatrix} x^{(i+1)} \\ z^{(i+1)} \end{bmatrix} - \begin{bmatrix} x^{(i)} \\ z^{(i)} \end{bmatrix} \right\|_2 \to 0.$

Proof. See §A.4 in the appendices.

Using Proposition 9, our main theorem provides that any limit points of the sequence generated by Momentum-Net satisfy critical point and fixed-point conditions:

Theorem 10 (A limit point satisfies both critical point and fixed-point conditions)**.** *Under Assumptions 1–4 above, let* $\{x^{(i+1)}, z^{(i+1)} : i \geq 0\}$ *be the sequence generated by Algorithm 1. Consider either a fixed majorization matrix with general structure, i.e.,* $M^{(i+1)} = M$ *for* $i \geq 0$*, or a sequence of diagonal majorization matrices, i.e.,* $\{M^{(i+1)} : i \geq 0\}$*. Then, any limit point* $\bar{x}$ *of* $\{x^{(i+1)}\}$ *satisfies both the critical point condition:*

$$\langle \nabla F(\bar{x}; y, \bar{z}), x - \bar{x} \rangle \geq 0, \qquad \forall x \in \mathcal{X}, \tag{16}$$

where $\bar{z}$ *is a limit point of* $\{z^{(i+1)}\}$*, and the fixed-point condition:*

$$\begin{bmatrix} \bar{x} \\ \bar{x} \end{bmatrix} = \mathcal{A}^{\bar{M}}_{\mathcal{R}_{\bar{\theta}}} \left(\begin{bmatrix} \bar{x} \\ \bar{x} \end{bmatrix} \right), \tag{17}$$

where $\begin{bmatrix} x^{(i+1)} \\ x^{(i)} \end{bmatrix} = \mathcal{A}^{M^{(i+1)}}_{\mathcal{R}_{\theta^{(i+1)}}} \left(\begin{bmatrix} x^{(i)} \\ x^{(i-1)} \end{bmatrix} \right)$*,* $\mathcal{A}^{M^{(i+1)}}_{\mathcal{R}_{\theta^{(i+1)}}}(\cdot)$ *denotes performing the* ith *updates in Algorithm 1, and* $\bar{\theta}$ *and* $\bar{M}$ *is a limit point of* $\{\theta^{(i+1)}\}$ *and* $\{M^{(i+1)}\}$*, respectively.*

Proof. See §A.5 in the appendices.

Observe that, if $\mathcal{X} = \mathbb{R}^N$ or $\bar{x}$ is an interior point of $\mathcal{X}$, (16) reduces to the first-order optimality condition $0 \in \partial F(\bar{x}; y, \bar{z})$, where $\partial F(x)$ denotes the limiting subdifferential of F at x. With additional isolation and boundedness assumptions for the points satisfying (16) and (17), we obtain whole sequence guarantees:

Corollary 11 (Whole sequence convergence)**.** *Consider the construction in Theorem 10. Let* $\mathcal{S}$ *be the set of points satisfying the critical point condition in (16) and the fixed-point condition in (17). If* $\{x^{(i+1)} : i \geq 0\}$ *is bounded, then* $\operatorname{dist}(x^{(i+1)}, \mathcal{S}) \to 0$*, where* $\operatorname{dist}(u, \mathcal{V}) \triangleq \inf\{\|u - v\| : v \in \mathcal{V}\}$ *denotes the distance from* u *to* $\mathcal{V}$*, for any point* $u \in \mathbb{R}^N$ *and any subset* $\mathcal{V} \subset \mathbb{R}^N$*. If* $\mathcal{S}$ *contains uniformly isolated points, i.e., there exists* $\eta > 0$ *such that* $\|u - v\| \geq \eta$ *for any distinct points* $u, v \in \mathcal{S}$*, then* $\{x^{(i+1)}\}$ *converges to a point in* $\mathcal{S}$*.*

Proof. See §A.6 in the appendices.

The boundedness assumption for $\{x^{(i+1)}\}$ in Corollary 11 is standard in block-wise optimization, e.g., [24], [35], [53], [55], [62]. The assumption can be satisfied if the set $\mathcal{X}$ is bounded (e.g., box constraints), one chooses appropriate regularization parameters in Algorithm 1 [24], [35], [55], the function $F(x; y, z)$ is coercive [62], or the level set is bounded [53]. However, for general $F(x; y, z)$, it is hard to verify the isolation condition for the points in $\mathcal{S}$ in practice. Instead, one may use Kurdyka-Łojasiewicz property [53], [62] to analyze the whole sequence convergence with some appropriate modifications.

For simplicity, we focused our discussion to noniterative MBIR module (Alg.1.3). However, Momentum-Net practically converges with *any* proximable MRIR function (Alg.1.3) that may need an iterative solver, if sufficient inner iterations are used. To maximize the computational benefit of Momentum-Net, one needs to make sure that majorized MBIR function (Alg.1.3) is better proximable over its original form (P1).

2.5 Benefits of Momentum-Net

Momentum-Net has several benefits over existing INNs:

- *Benefits from refining module*: The image refining module (Alg.1.1) can use iteration-wise image refining NNs $\{\mathcal{R}_{\theta^{(i+1)}} : i \geq 0\}$: those are particularly useful to reduce overfitting risks by reducing dimensions of their parameters $\theta^{(i+1)}$ at each iteration [16], [18], [19]. Iteration-wise refining NNs require less memory for training, compared to methods that use a single refining NN for all iterations, e.g., [63]. Different from the existing methods mentioned in §1.3, Momentum-Net does not require (firmly) nonexpansive mapping operators $\{\mathcal{R}_{\theta^{(i+1)}}\}$ to guarantee convergence. Instead, $\{\mathcal{R}_{\theta^{(i+1)}}\}$ in (Alg.1.1) assumes a generalized notion of the (firm) non-expansiveness condition assumed for convergence of the existing methods that use identical refining NNs across iterations, including PnP [20], [23], [39]–[41], [46], RED [22], [43], etc. The generalized concept is the first practical condition to guarantee convergence of INNs using iteration-wise refining NNs; see Definition 7.
- *Benefits from extrapolation module*: The extrapolation module (Alg.1.2) uses the momentum terms $x^{(i)} - x^{(i-1)}$ that accelerate the convergence of Momentum-Net. In particular, compared to the existing gradient-descent-inspired INNs, e.g., TNRD [21], [28], Momentum-Net converges faster. (Note that the way the authors of [43] used momentum is less conventional. The corresponding method, RED-APG [43, Alg. 6], still can require multiple inner iterations to solve its quadratic MBIR problem, similar to BCD-Net-type methods.)
- *Benefits from MBIR module*: The MBIR module (Alg.1.3) does not require multiple inner iterations for a wide range of imaging problems and has both theoretical and practical benefits. Note first that convergence analysis of INNs (including Momentum-Net) assumes that their MBIR operators are *noniterative*. In other words, related convergence theory (e.g., Proposition A.5) is inapplicable if iterative methods, particularly with insufficient number of iterations, are applied to MBIR modules. Different from the existing BCD-Net-type methods [20], [22], [23], [25], [30], [39]–[41], [43] that can require iterative solvers for their MBIR modules, MBIR module (Alg.1.3) of Momentum-Net can have practical close-form solution (see examples in §2.2), and its corresponding convergence analysis (see §2.4) can hold stably for a

wide range of imaging applications. Second, combined with extrapolation module (Alg.1.2), noniterative MBIR modules (Alg.1.3) lead to faster MBIR, compared to the existing BCD-Net-type methods that can require multiple inner iterations for their MBIR modules for convergence. Third, Momentum-Net guarantees convergence even for nonconvex MBIR cost function $F(x;y,z)$ or nonconvex data-fit $f(x;y)$ of which the gradient is M-Lipschitz continuous (see Definition 1), while existing INNs overlooked nonconvex $F(x;y,z)$ or $f(x;y)$.

Furthermore, §A.7 analyzes the sequence convergence of BCD-Net [25], and describes the convergence benefits of Momentum-Net over BCD-Net.

3 Training INNs

This section describes training of all the INNs compared in this paper.

3.1 Architecture of refining NNs and their training

For all INNs in this paper, we train the refining NN at each iteration to remove artifacts from the input image $x^{(i)}$ that is fed from the previous iteration. For the ith iteration NN, we first consider the following sCNN architecture, residual single-hidden layer convolutional autoencoder:

$$\mathcal{R}_{\theta^{(i+1)}}(u) = \sum_{k=1}^{K} d_k^{(i+1)} * \mathcal{T}_{\exp(\alpha_k^{(i+1)})}\big(e_k^{(i+1)} * u\big) + u, \quad (18)$$

where $\theta^{(i+1)} = \{d_k^{(i+1)}, \alpha_k^{(i+1)}, e_k^{(i+1)} : \forall k\}$ is the parameter set of the ith image refining NN, $\{d_k^{(i+1)}, e_k^{(i+1)} \in \mathbb{C}^R : k = 1,\dots,K\}$ is a set of K decoding and encoding filters of size R, $\{\exp(\alpha_k^{(i+1)}) : k = 1,\dots,K\}$ is a set of K thresholding values, and $\mathcal{T}_\alpha(u)$ is the soft-thresholding operator with parameter α defined in (6), for $i = 0,\dots,N_{\text{iter}}-1$. We use the exponential function $\exp(\cdot)$ to prevent the thresholding parameters $\{\alpha_k\}$ from becoming negative during training. We observed that the residual convolutional autoencoder in (18) gives better results compared to the convolutional autoencoder, i.e., (18) without the second term [18], [25]. This corresponds to the empirical result in [64], [65] that having skip connections (e.g., the second term in (18)) can improve generalization. The sequence of paired sCNN refiners (18) can satisfy the asymptotic non-expansiveness, if its convergent refiner satisfies that

$$\sigma_{\max}(\overline{D}^H \overline{D}) \le 1/R, \qquad \sigma_{\max}(\overline{E}^H \overline{E}) \le 1/R,$$

where $\sigma_{\max}(\cdot)$ is the largest eigenvalue of a matrix, $\overline{D} \triangleq [\bar{d}_1,\dots,\bar{d}_K,\delta_R]$, $\overline{E} \triangleq [\bar{e}_1,\dots,\bar{e}_K,\delta_R]$, $\{\bar{d}_k,\bar{e}_k : \forall k\}$ are limit point filters, and δ_R is the Kronecker delta filter of size R.

For dCNN refiners, we use the following residual multi-hidden layer CNN architecture, a simplified DnCNN [4] using fewer layers, no pooling, and no batch normalization [46] (we drop superscript indices $(\cdot)^{(i)}$ for simplicity):

$$\mathcal{R}_\theta(u) = u - \sum_{k=1}^{K} e_k^{[L]} * u_k^{[L-1]}, \quad (19)$$

$$u_k^{[1]} = \text{ReLU}\Big(e_k^{[1]} * u\Big), \quad u_k^{[l]} = \text{ReLU}\left(\sum_{k'=1}^{K} e_{k,k'}^{[l]} * z_{k'}^{[l-1]}\right),$$

for $k = 1,\dots,K$ and $l = 2,\dots,L-1$, where $\theta = \{e_k^{[l]}, e_{k,k'}^{[l]} : \forall k,k',l\}$ is the parameter set of each refining NN, K is the number of feature maps, L is the number of layers, $\{e_k^{[l]} \in \mathbb{R}^R : k = 1,\dots,K, l = 1, L\}$ is a set of filters at the first and last dCNN layer, $\{e_{k,k'}^{[l]} \in \mathbb{R}^R : k,k' = 1,\dots,K, l = 2,\dots,L-1\}$ is a set of filters for remaining dCNN layer, and the rectified linear unit activation function is defined by $\text{ReLU}(u) \triangleq \max(0,u)$.

The training process of Momentum-Net requires S high-quality training images, $\{x_s : s = 1,\dots,S\}$, S training measurements simulated via imaging physics, $\{y_s : s = 1,\dots,S\}$, and S data-fits $\{f_s(x;y_s) : s = 1,\dots,S\}$ and the corresponding majorization matrices $\{M_s^{(i)}, \widetilde{M}_s^{(i)} : s = 1,\dots,S, i = 1,\dots,N_{\text{iter}}\}$. Different from [16], [25], [66] that train convolutional autoencoders from the patch perspective, we train the image refining NNs in (18)–(19) from the convolution perspective (that does not store many overlapping patches, e.g., see [24]). From S training pairs $(x_s, x_s^{(i)})$, where $\{x_s^{(i)} : s = 1,\dots,S\}$ is a set of S reconstructed images at the $(i-1)$th Momentum-Net iteration, we train the ith iteration image refining NN in (18) by solving the following optimization problem:

$$\theta^{(i+1)} = \underset{\theta}{\text{argmin}} \frac{1}{2S} \sum_{s=1}^{S} \left\| x_s - \mathcal{R}_\theta(x_s^{(i)}) \right\|_2^2, \quad \text{(P2)}$$

where $\theta^{(i+1)}$ is given as in (18), for $i = 0,\dots,N_{\text{iter}}-1$ (see some related properties in §A.8). We solve the training optimization problems (P2) by mini-batch stochastic optimization with the subdifferentials computed by the PyTorch `Autograd` package.

3.2 Regularization parameter selection based on "spectral spread"

When majorization matrices of training data-fits $\{f_s(x;y_s) : s = 1,\dots,S\}$ have similar spectral properties, e.g., condition numbers, the regularization parameter γ in (P1) is trainable by substituting (Alg.1.1) into (Alg.1.3) and modifying the training cost (P2). However, the condition numbers of data-fit majorizers can greatly differ due a variety of imaging geometries or image formation systems, or noise levels in training measurements, etc. See such examples in §4.1–4.2.

To train Momentum-Net with diverse training data-fits, we propose a parameter selection scheme based on the "spectral spread" of their majorization matrices $\{M_{f_s}^{(i)}\}$. For simplicity, consider majorization matrices of the form $\widetilde{M}_s^{(i)} = \widetilde{M}_s = \lambda(M_{f_s} + \gamma_s I)\ \forall i$, where the factor λ is selected by (7) and M_{f_s} is a symmetric positive semidefinite majorization matrix for $f_s(x;y_s)$, $\forall s$. We select the regularization parameter γ_s for the sth training sample as

$$\gamma_s = \frac{\sigma_{\text{spread}}(M_{f_s})}{\chi}, \quad (20)$$

where the spectral spread of a symmetric positive definite matrix is defined by $\sigma_{\text{spread}}(\cdot) \triangleq \sigma_{\max}(\cdot) - \sigma_{\min}(\cdot)$ for $\sigma_{\max}(M_{f_s}) > \sigma_{\min}(M_{f_s}) \ge 0$, and $\sigma_{\min}(\cdot)$ is the smallest eigenvalue of a matrix. For the sth training sample, a tunable factor χ controls γ_s in (20) according to $\sigma_{\text{spread}}(M_{f_s})$, $\forall s$. The proposed parameter selection scheme also applies to testing Momentum-Net, based on the tuned factor $\chi^\star$ in its

training. We observed that the proposed parameter selection scheme (20) gives better MBIR accuracy than the condition number based selection scheme that is similarly used in selecting ADMM parameters [17] (for the two applications in §4). One may further apply this scheme to iteration-wise majorization matrices $\widetilde{M}_s^{(i)}$ and select iteration-wise regularization parameters $\gamma_s^{(i)}$ accordingly. For comparing different INNs, we apply (20) to all INNs.

Algorithm 2 BCD-Net [25]

Require: $\{\mathcal{R}_{\theta^{(i)}} : i = 1, \dots, N_{\text{iter}}\}$, $\gamma > 0$, $x^{(0)} = x^{(-1)}$, y

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

Image refining:

$$z^{(i+1)} = \mathcal{R}_{\theta^{(i+1)}}\left(x^{(i)}\right) \quad \text{(Alg.2.1)}$$

MBIR:

$$x^{(i+1)} = \underset{x \in \mathcal{X}}{\operatorname{argmin}} F(x; y, z^{(i+1)}) \quad \text{(Alg.2.2)}$$

end for

4 Experimental results and discussion

We investigated two extreme imaging applications: sparse-view CT and LF photography using a focal stack. In particular, these two applications lack a practical closed-form solution for the MBIR modules of BCD-Net and ADMM-Net [20], e.g., solving (Alg.2.2). For these applications, we compared the performances of the following five INNs: BCD-Net [25] (i.e., generalization of RED [22] and MoDL [30]), ADMM-Net [20], i.e., PnP-ADMM [23], [41] using iteration-wise refining NNs, Momentum-Net *without extrapolation* (i.e., generalization of TNRD [21], [28]), PDS-Net, i.e., PnP-PDS [50] using iteration-wise refining NNs, and the proposed Momentum-Net using extrapolation.

4.1 Experimental setup: Imaging

4.1.1 Sparse-view CT

To reconstruct a linear attenuation coefficient image $x \in \mathbb{R}^N$ from post-log sinogram $y \in \mathbb{R}^m$ in sparse-view CT, the MBIR problem (P1) considers a data-fit $f(x; y) = \frac{1}{2}\|y - Ax\|_W^2$ and the non-negativity constraint $\mathcal{X} = [0, \infty)^N$, where $A \in \mathbb{R}^{m \times N}$ is an undersampled CT system matrix, $W \in \mathbb{R}^{m \times m}$ is a diagonal weighting matrix with elements $\{W_{m',m'} = p_{m'}'^2/(p_{m'}' + \sigma^2) : \forall m'\}$ based on a Poisson-Gaussian model [17], [67] for the pre-log raw measurements $p \in \mathbb{R}^m$ with electronic readout noise variance σ^2.

We simulated 2D sparse-view sinograms of size $m = 888 \times 123$ – 'detectors or rays' × 'regularly spaced projection views or angles', where 984 is the number of full views – with GE LightSpeed fan-beam geometry corresponding to a monoenergetic source with 10^5 incident photons per ray and no background events, and electronic noise variance $\sigma^2 = 5^2$. We avoided an inverse crime in imaging simulation and reconstructed images of size $N = 420 \times 420$ with a coarser grid $\Delta_x = \Delta_y = 0.9766$ mm; see details in [37, §V-A2].

4.1.2 LF photography using a focal stack

To reconstruct a LF $x = [x_1^T, \dots, x_{C'}'^T]^T \in \mathbb{R}^{SN'}$ that consists of C' sub-aperture images from focal stack measurements $y = [y_1^T, \dots, y_C^T]^T \in \mathbb{R}^{CN'}$ that are collected by C photosensors, the MBIR problem (P1) considers a data-fit $f(x; y) = \frac{1}{2}\|y - Ax\|_2^2$ and a box constraint $\mathcal{X} = [0, U]^{C'N'}$ with $U = 1$ (or 255 without rescaling), where $A \in \mathbb{R}^{CN' \times C'N'}$ is a system matrix of LF imaging system using a focal stack that is constructed blockwise with $C \cdot C'$ different convolution matrices $\{\tau_c A_{c,c'}' \in \mathbb{R}^{N' \times N'} : c = 1, \dots, C, c' = 1, \dots, C'\}$ [47], [68], $\tau_c \in (0, 1]$ is a transparency coefficient for the cth detector, and N' is the size of sub-aperture images, $x_{c'}$, $\forall c'$.[3] In general, a LF photography system using a focal stack is extremely under-determined, because $C \ll C'$.

To avoid an inverse crime, our imaging simulation used higher-resolution synthetic LF dataset [70] (we converted the original RGB sub-aperture images to grayscale ones by the "`rgb2gray.m`" function in MATLAB, for simplicity and smaller memory requirements in training). We simulated $C = 5$ focal stack images of size $N' = 255 \times 255$ with 40 dB AWGN that models electronic noise at sensors, and setting transparency coefficients τ_c as 1, for $c = 1, \dots, C$. The sensor positions were chosen such that five sensors focus at equally spaced depths; specifically, the closest sensor to scenes and farthest sensor from scenes focus at two different depths that correspond to '$\texttt{disp}_{\min} + 0.2$' and '$\texttt{disp}_{\max} - 0.2$', respectively, where $\texttt{disp}_{\max}$ and $\texttt{disp}_{\min}$ are the approximate maximum and minimum disparity values specified in [70]. We reconstructed 4D LFs that consist of $S = 9 \times 9$ sub-aperture images of size $N' = 255 \times 255$, with a coarser grid $\Delta_x = \Delta_y = 0.13572$ mm.

4.2 Experimental setup: INNs

4.2.1 Parameters of INNs

The parameters for the INNs compared in sparse-view CT experiments were defined as follows. We considered two BCD-Nets (see Algorithm 2): for one BCD-Net, we applied the APG method [49] with 10 inner iterations to (Alg.2.2), and set $N_{\text{iter}} = 30$; for the other BCD-Net, we applied the APG method with 3 inner iterations to (Alg.2.2), and set $N_{\text{iter}} = 45$. For ADMM-Net, we used the identical configurations as BCD-Net and set the ADMM penalty parameter to γ in (P1), similar to [16]. For Momentum-Net without extrapolation, we chose $N_{\text{iter}} = 100$ and $\rho = 1 - \varepsilon$. For the proposed Momentum-Net, we chose $N_{\text{iter}} = 100$ and $\rho = 0.5$. For PDS-Net, we set the first step size to $\gamma_1 = \gamma^{-1}$ and the second step size to $\gamma_2 = \gamma_1^{-1}\sigma_{\max}^{-1}(M)$, per [50]. For performance comparisons between different INNs, all the INNs used sCNN refiners (18) with $\{R, K = 7^2\}$ to avoid the overfitting/hallucination risks. For Momentum-Net using dCNN refiners, we chose $L = 4$ layer dCNN (19) using $R = 3^2$ filters and $K = 64$ feature maps, following [46]. (The chosen parameters gave lower RMSE values than $\{L = 6, R = 3^3, K = 64\}$, for identical regularization parameters.) For comparing different MBIR methods, Momentum

3. Traditionally, one obtains focal stacks by physically moving imaging sensors and taking separate exposures across time. Transparent photodetector arrays [47], [69] allow one to collect focal stack data in a single exposure, making a practical LF camera using a focal stack. If some photodetectors are not perfectly transparent, one can use $\tau_c < 1$, for some c.

used extrapolation, i.e., (Alg.1.2) with (8) and (10), and $\{R=7^2, K=9^2\}$ for (18). We designed the majorization matrices as $\{\widetilde{M}^{(i+1)} = \mathrm{diag}(A^T W A 1) + \gamma I \,:\, i \geq 0\}$, using Lemma A.7 ($A$ and W have nonnegative entries) and setting $\lambda = 1$ by (7). We set an initial point of INNs, $x^{(0)}$, to filtered-back projection (FBP) using a Hanning window. The regularization parameters of all INNs were selected by the scheme in §3.2 with $\chi^\star = 167.64$. (This factor was estimated from the carefully chosen regularization parameter for sparse-view CT MBIR experiments using learned convolutional regularizers in [24].)

The parameters for the INNs compared in experiments of LF photography using a focal stack were defined as follows. We considered two BCD-Nets and two ADMM-Nets with the identical parameters listed above. For Momentum-Net without extrapolation and the proposed Momentum-Net, we set $N_{\text{iter}} = 100$ and $\rho = 1 - \varepsilon$. For PDS-Net, we used the identical parameter setup described above. For performance comparisons between different INNs, all the INNs used sCNN refiners (18) with $\{R = 5^2, K = 32\}$ (to avoid the overfitting risks) in the epipolar domain. For Momentum-Net using dCNN refiners, we chose $L=6$ layer dCNN (19) using $R = 3^2$ filters and $K = 16$ feature maps. (The chosen parameters gave most accurate performances over the following setups, $\{L = 4, R = 3^2, K = 16, 32, 64\}$, given the identical regularization parameters.) To generate $\mathcal{R}_{\theta^{(i+1)}}(x^{(i)})$ in (Alg.1.1), we applied a sCNN (18) with $\{R=5^2, K=32\}$ or a dCNN (19) with $\{L=6, R=3^2, K=16\}$ to two sets of horizontal and vertical epipolar plane images, and took the average of two LFs that were permuted back from refined horizontal and vertical epipolar plane image sets, $\forall i$.[4] We designed the majorization matrices as $\{\widetilde{M}^{(i+1)} = \mathrm{diag}(A^T A 1) + \gamma I \,:\, i \geq 0\}$, using Lemma A.7 and setting $\lambda = 1$ by (7). We set an initial point of INNs, $x^{(0)}$, to $A^T y$ rescaled in the interval $[0, 1]$ (i.e., dividing by its max value). The regularization parameters (i.e., γ in BCD-Net/Momentum-Net, the ADMM penalty parameter in ADMM-Net, and the first step size in PDS-Net) were selected by the proposed scheme in §3.2 with $\chi^\star = 1.5$. (We tuned the factor to achieve the best performances).

For different combinations of INNs and sCNN refiner (18)/dCNN refiner (19), we use the following naming convention: 'the INN name'-'sCNN' or 'dCNN'.

4.2.2 Training INNs

For sparse-view CT experiments, we trained all the INNs from the chest CT dataset with $\{x_s, y_s, f_s(x; y_s) = \frac{1}{2}\|y_s - Ax\|^2_{W_s}, \widetilde{M}_s \,:\, s = 1, \ldots, S, S = 142\}$; we constructed the dataset by using XCAT phantom slices [71]. The CT experiment has mild data-fit variations across training samples: the standard deviation of the condition numbers ($\triangleq \sigma_{\max}(\cdot)/\sigma_{\min}(\cdot)$) of $\{M_{f_s} = \mathrm{diag}(A^T W_s A 1) : \forall s\}$ is 1.1. For experiments of LF photography using a focal stack, we trained all the INNs from the LF photography dataset with $\{y_s, f_s(x; y_s) = \frac{1}{2}\|y_s - A_s x\|_2^2, \widetilde{M}_s : s = 1, \ldots, S, S = 21\}$ and two sets of ground truth epipolar images, $\{x_{s,\text{epi-h}}, x_{s,\text{epi-v}} : s = 1, \ldots, S, S = 21 \cdot (255 \cdot 9)\}$; we constructed the dataset by excluding four unrealistic "stratified" scenes from the original LF dataset in [70] that consists of 28 LFs with diverse scene parameter and camera settings. The LF experiment has large data-fit variations across training samples: the standard deviation of the condition numbers of $\{M_{f_s} = \mathrm{diag}(A_s^T A_s 1) : \forall s\}$ is 2245.5.

In training INNs for both the applications, if not specified, we used identical training setups. At each iteration of INNs, we solved (P2) with the mini-batch version of Adam [72] and trained iteration-wise sCNNs (18) or dCNNs (19). We selected the batch size and the number of epochs as follows: for sparse-view CT reconstruction, we chose them as 20 & 300, and 20 & 200 for sCNN and dCNN refiners, respectively; for LF photography using a focal stack, we chose them as 200 & 200, and 200 & 100, for sCNN and dCNN refiners, respectively. We chose the learning rates for filters in sCNNs and dCNNs, and thresholding values $\{\alpha_k^{(i+1)} : \forall k, i\}$ in sCNNs (18), as 10^{-3} and 10^{-1}, respectively; we reduced the learning rates by 10% every 10 epochs. At the first iteration, we initialized filter coefficients with Kaiming uniform initialization [73]; in the later iteration, i.e., at the ith INN iteration, for $i \geq 2$, we initialized filter coefficients from those learned from the previous iteration, i.e., $(i-1)$th iteration (this also applies to initializing thresholding values).

4.2.3 Testing trained INNs

In sparse-view CT reconstruction experiments, we tested trained INNs to two samples where ground truth images and the corresponding inverse covariance matrices (i.e., W in §4.1.1) sufficiently differ from those in training samples (i.e., they are a few cm away from training images). We evaluated the reconstruction quality by the most conventional error metric in CT application, RMSE (in HU), in a region of interest (ROI), where RMSE and HU stand for root-mean-square error and (modified) Hounsfield unit, respectively, and the ROI was a circular region that includes all the phantom tissues. The RMSE is defined by $\mathrm{RMSE}(x^\star, x^{\text{true}}) \triangleq (\sum_{j=1}^{N_{\text{ROI}}} (x_j^\star - x_j^{\text{true}})^2 / N_{\text{ROI}})^{1/2}$, where $x^\star$ is a reconstructed image, x^{true} is a ground truth image, and N_{ROI} is the number of pixels in a ROI. In addition, we compared the trained Momentum-Net (using extrapolation) to a standard MBIR method using a hand-crafted EP regularizer, and an MBIR model using a learned convolutional regularizer [24], [37] which is the state-of-the-art MBIR method within an unsupervised learning setup. We finely tuned their regularization parameters to achieve the lowest RMSE. See details of these two MBIR models in §A.9.2.

In experiments of LF photography using a focal stack, we tested trained INNs to three samples of which scene parameter and camera settings are different from those in training samples (all training and testing samples have different camera and scene parameters). We evaluated the reconstruction quality by the most conventional error metric in LF photography application, PSNR (in dB), where PSNR stands for peak signal-to-noise. In addition, we compared the trained Momentum-Net (using extrapolation) to MBIR method using the state-of-the-art non-trained regularizer, 4D EP introduced in [47]. (The low-rank plus sparse tensor

4. Epipolar images are 2D slices of a 4D LF $L_F(c_x, c_y, c_u, c_v)$, where (c_x, c_y) and (c_u, c_v) are spatial and angular coordinates, respectively. Specifically, each horizontal epipolar plane image are obtained by fixing c_y and c_v, and varying c_x and c_u; and each vertical epipolar image are obtained by fixing c_x and c_u, and varying c_y and c_v.

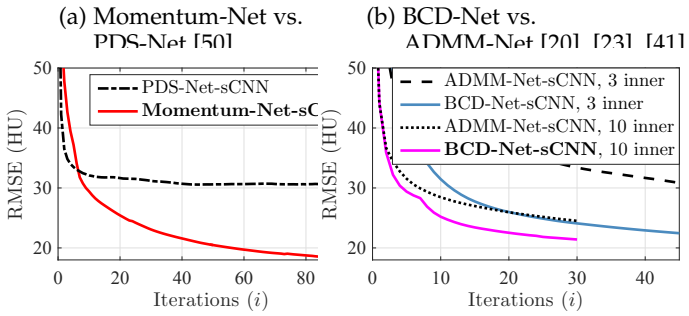


Fig. 4. RMSE minimization comparisons between different INNs for sparse-view CT (fan-beam geometry with 12.5% projections views and 10^5 incident photons; (a) averaged RMSE values across two test refined images; (b) averaged RMSE values across two test reconstructed images).

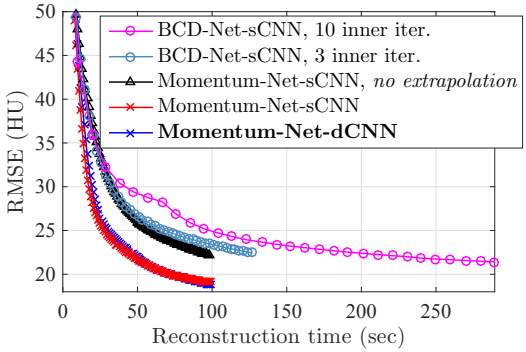


Fig. 5. RMSE minimization comparisons between different INNs for sparse-view CT (fan-beam geometry with 12.5% projections views and 10^5 incident photons; averaged RMSE values across two test reconstructed images).

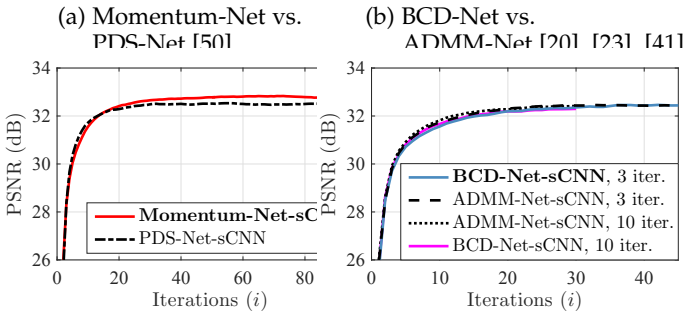


Fig. 6. PSNR maximization comparisons between different INNs in LF photography using a focal stack (LF photography systems with $C = 5$ detectors obtain a focal stack of LFs consisting of $S = 81$ sub-aperture images; (a) averaged RMSE values across two test refined images); (b) averaged RMSE values across two test reconstructed images.

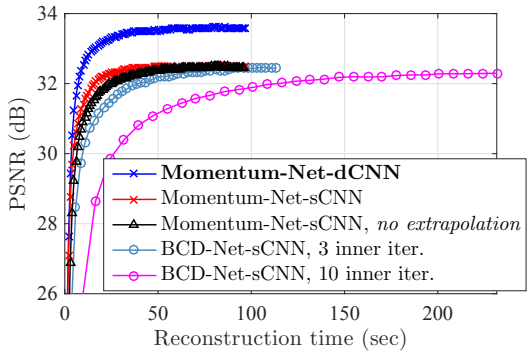


Fig. 7. PSNR maximization comparisons between different INNs in LF photography using a focal stack (LF photography systems with $C = 5$ detectors obtain a focal stack of LFs consisting of $S = 81$ sub-aperture images; averaged PSNR values across three test reconstructed images).

decomposition model [68], [74] failed when inverse crimes and measurement noise are considered.) We finely tuned its regularization parameter to achieve the lowest RMSE values. See details of this MBIR model in §A.9.3. We further investigated impacts of the LF MBIR quality on a higher-level depth estimation application, by applying the robust Spinning Parallelogram Operator (SPO) depth estimation method [75] to reconstructed LFs.

For comparing Momentum-Net with PDS-Net, we measured quality of refined images, $z^{(i+1)}$ in (Alg.1.1), because PDS-Net is a hard-refiner.

The imaging simulation and reconstruction experiments were based on the Michigan image reconstruction toolbox [76], and training INNs, i.e., solving (P2), was based on PyTorch (for sparse-view CT, we used ver. 1.2.0; for LF photography using a focal stack, we used ver. 0.3.1). For sparse-view CT, single-precision MATLAB and PyTorch implementations were tested on 2.6 GHz Intel Core i7 CPU with 16 GB RAM, and 1405 MHz Nvidia Titan Xp GPU with 12 GB RAM, respectively. For LF photography using a focal stack, they were tested on 3.5 GHz AMD Threadripper 1920X CPU with 32 GB RAM, and 1531 MHz Nvidia GTX 1080 Ti GPU with 11 GB RAM, respectively.

4.3 Comparisons between different INNs

First, compare sCNN results in Figs. 4–5 and Figs. 6–7, for sparse-view CT and LF photography using a focal stack, respectively. For both applications, the proposed Momentum-Net using extrapolation significantly improves MBIR speed and accuracy, compared to the existing soft-refining INNs, [21]–[23], [28], [30] that correspond to BCD-Net [25] or Momentum-Net using *no extrapolation*, and ADMM-Net [20], [23], [41], and the existing hard-refining INN PDS-Net [50]. (Note that BCD-Net and Momentum-Net require slightly less computational complexity per INN iteration, compared to ADMM-Net and PDS-Net, respectively, due to having fewer modules.) Fig. 5 shows that to reach the 24 HU RMSE value in sparse-view CT reconstruction, the proposed Momentum-Net decreases MBIR time by 53.3% and 62.5%, compared to Momentum-Net without extrapolation and BCD-Net using three inner iterations, respectively. Fig. 7 shows that to reach the 32 dB PSNR value in LF reconstruction from a focal stack, the proposed Momentum-Net decreases MBIR time by 36.5% and 61.5%, compared to Momentum-Net without extrapolation and BCD-Net using three inner iterations, respectively. In addition, Figs. 5 and 7 show that using extrapolation, i.e., (Alg.1.2) with (8)–(10), improves the performance of Momentum-Net versus iterations.

We conjecture that the larger performance gap between soft-refiner Momentum-Net and hard-refiner PDS-Net, in Fig. 4(a) compared to Fig. 6(a), is because the LF problem needs stronger regularization, i.e., a smaller tuned factor $\chi^\star$ in (20), than the CT problem. Similarly, comparing Fig. 4(b) to Fig. 6(b) shows that the LF problem has small performance gaps between BCD-Net and ADMM-Net.

For both the applications, using dCNN refiners (19) instead of sCNN refiners (18) has a negligible effect on total

run time of Momentum-Net, because reconstruction time of MBIR modules (Alg.1.3) (in CPUs) dominates inference time of image refining modules (Alg.1.1) (in GPUs). Compare results between Momentum-Net-sCNN and -dCNN in Figs. 5 & 7 and Tables A.1 & A.2.

4.4 Comparisons between different MBIR methods

In sparse-view CT using 12.5% of the full projection views, Fig. 8(b)–(e) and Table A.1(b)–(f) show that the proposed Momentum-Net achieves significantly better reconstruction quality compared to the conventional EP MBIR method and the state-of-the-art MBIR method within an unsupervised learning setup, MBIR model using a learned convolutional regularizer [24], [37]. In particular, Momentum-Net recovers both low- and high-contrast regions (e.g., soft tissues and bones, respectively) more accurately than MBIR using a learned convolutional regularizer; see Fig. 8(c)–(e). In addition, when their shallow convolutional autoencoders need identical computational complexities, Momentum-Net achieves much faster MBIR compared to MBIR using a learned convolutional regularizer; see Table A.1(c)–(d).

In LF photography using five focal sensors, regardless of scene parameters and camera settings, Momentum-Net consistently achieves significantly more accurate image recovery, compared to MBIR model using the state-of-the-art non-trained regularizer, 4D EP [47]. The effectiveness of Momentum-Net is more evident for a scene with less fine details. See Fig. 9(b)–(d) and Table A.2(b)–(d). Regardless of the scene distances from LF imaging systems, the reconstructed LFs by Momentum-Net significantly improve the depth estimation accuracy over those reconstructed by the state-of-the-art non-trained regularizer, 4D EP [47]. See Fig. 10(c)–(e) and Table A.3(c)–(e).

In general, Momentum-Net needs more computations per iteration than EP MBIR, because its refining NNs use more and larger filters than the small finite-difference filters in EP MBIR, and EP MBIR algorithms can be often further accelerated by gradient approximations, e.g., ordered-subsets methods [77], [78].

5 Conclusions

Developing rapidly converging INNs is important, because *1)* it leads to fast MBIR by reducing the computational complexity in calculating data-fit gradients or applying refining NNs, and *2)* training INNs with many iterations requires long training time or it is challenging when refining NNs are fixed across INN iterations. The proposed Momentum-Net framework is applicable for a wide range of inverse problems, while achieving fast and convergent MBIR. To achieve fast MBIR, Momentum-Net uses momentum in extrapolation modules, and noniterative MBIR modules at each iteration via majorizers. For sparse-view CT and LF photography using a focal stack, Momentum-Net achieves faster and more accurate MBIR compared to the existing soft-refining INNs, [21]–[23], [28], [30] that correspond to BCD-Net [25] or Momentum-Net using *no extrapolation*, and ADMM-Net [20], [23], [41], and the existing hard-refining INN PDS-Net [50]. When an application needs strong regularization strength, e.g., LF photography using limited detectors, using dCNN refiners with moderate depth significantly improves the MBIR accuracy of Momentum-Net compared to sCNNs, only marginally increasing total MBIR time. In addition, Momentum-Net guarantees convergence to a fixed-point for general differentiable (non)convex MBIR functions (or data-fit terms) and convex feasible sets, under some mild conditions and two asymptotic conditions. The proposed regularization parameter selection scheme uses the "spectral spread" of majorization matrices, and is useful to consider data-fit variations across training/testing samples.

There are a number of avenues for future work. First, we expect to further improve performances of Momentum-Net (e.g., MBIR time and accuracy) by using sharper majorizer designs. Second, we expect to further reduce MBIR time of Momentum-Net with the stochastic gradient perspective (e.g., ordered subset [77], [78]). On the regularization parameter selection side, our future work is learning the factor χ in (20) from datasets while training refining NNs.

References

[1] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[2] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Proc. NIPS*, Lake Tahoe, NV, Dec. 2012, pp. 341–349.

[3] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. NIPS*, Barcelona, Spain, Dec. 2016, pp. 2802–2810.

[4] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Feb. 2017.

[5] L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *Proc. NIPS*, Montreal, Canada, Dec. 2014, pp. 1790–1798.

[6] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Proc. IEEE CVPR*, Boston, MA, Jun. 2015, pp. 769–777.

[7] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[8] J. Kim, K. J. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE CVPR*, Las Vegas, NV, Jun. 2016.

[9] G. Yang, S. Yu, H. Dong, G. Slabaugh, P. L. Dragotti, X. Ye, F. Liu, S. Arridge, J. Keegan, Y. Guo *et al.*, "DAGAN: Deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction," *IEEE Trans. Image Process.*, vol. 37, no. 6, pp. 1310–1321, Jun. 2018.

[10] T. M. Quan, T. Nguyen-Duc, and W. Jeong, "Compressed sensing MRI reconstruction using a generative adversarial network with a cyclic loss," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1488–1497, June 2018.

[11] H. Chen, Y. Zhang, M. K. Kalra, F. Lin, P. Liao, J. Zhou, and G. Wang, "Low-dose CT with a residual encoder-decoder convolutional neural network (RED-CNN)," *IEEE Trans. Med. Imag.*, vol. PP, no. 99, p. 0, Jun. 2017.

[12] 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, Sep. 2017.

[13] J. Ye, Y. Han, and E. Cha, "Deep convolutional framelets: A general deep learning framework for inverse problems," *SIAM J. Imaging Sci.*, vol. 11, no. 2, pp. 991–1048, Apr. 2018.

[14] G. Wu, M. Zhao, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field reconstruction using deep convolutional network on EPI," in *Proc. IEEE CVPR*, Honolulu, HI, Jul. 2017, pp. 1638–1646.

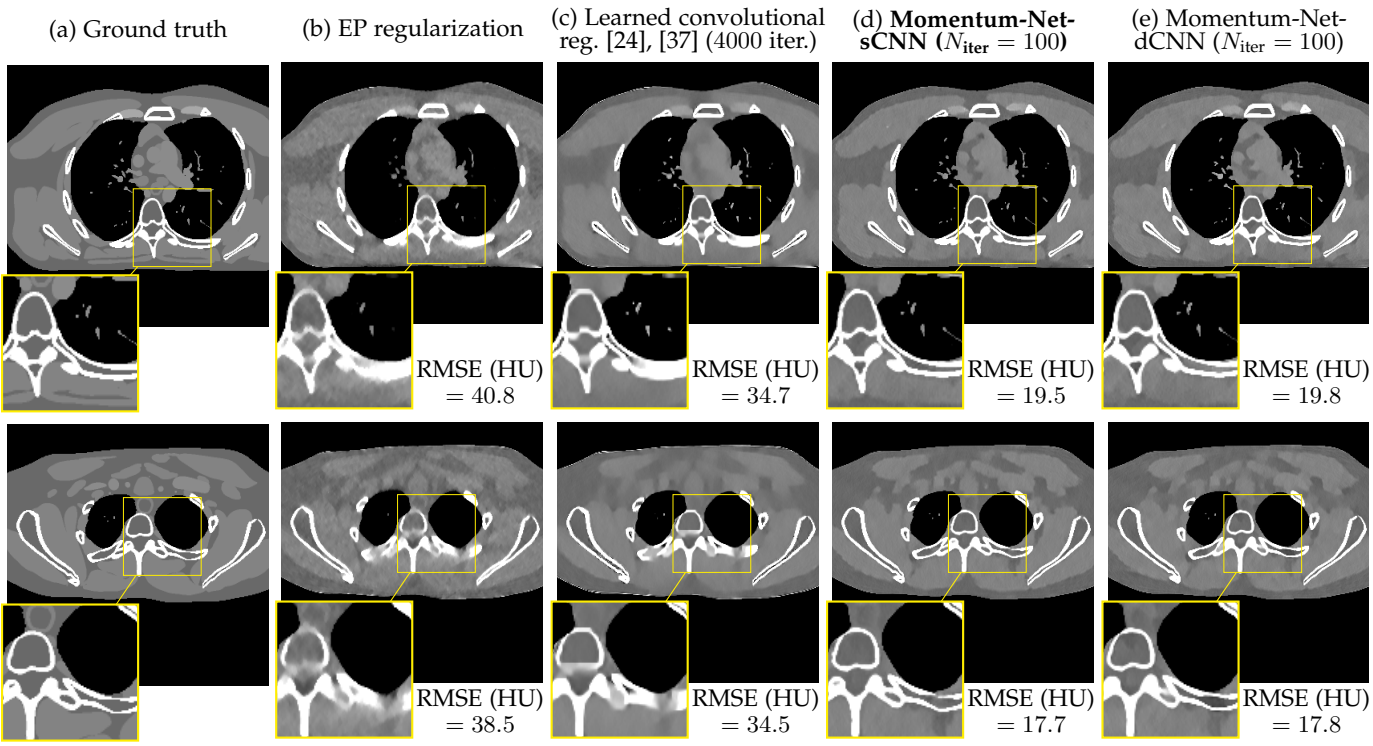


Fig. 8. Comparison of reconstructed images from different MBIR methods in sparse-view CT (fan-beam geometry with 12.5% projections views and 10^5 incident photons; images outside zoom-in boxes are magnified to better show differences; display window $[800, 1200]$ HU). See also Fig. A.2.

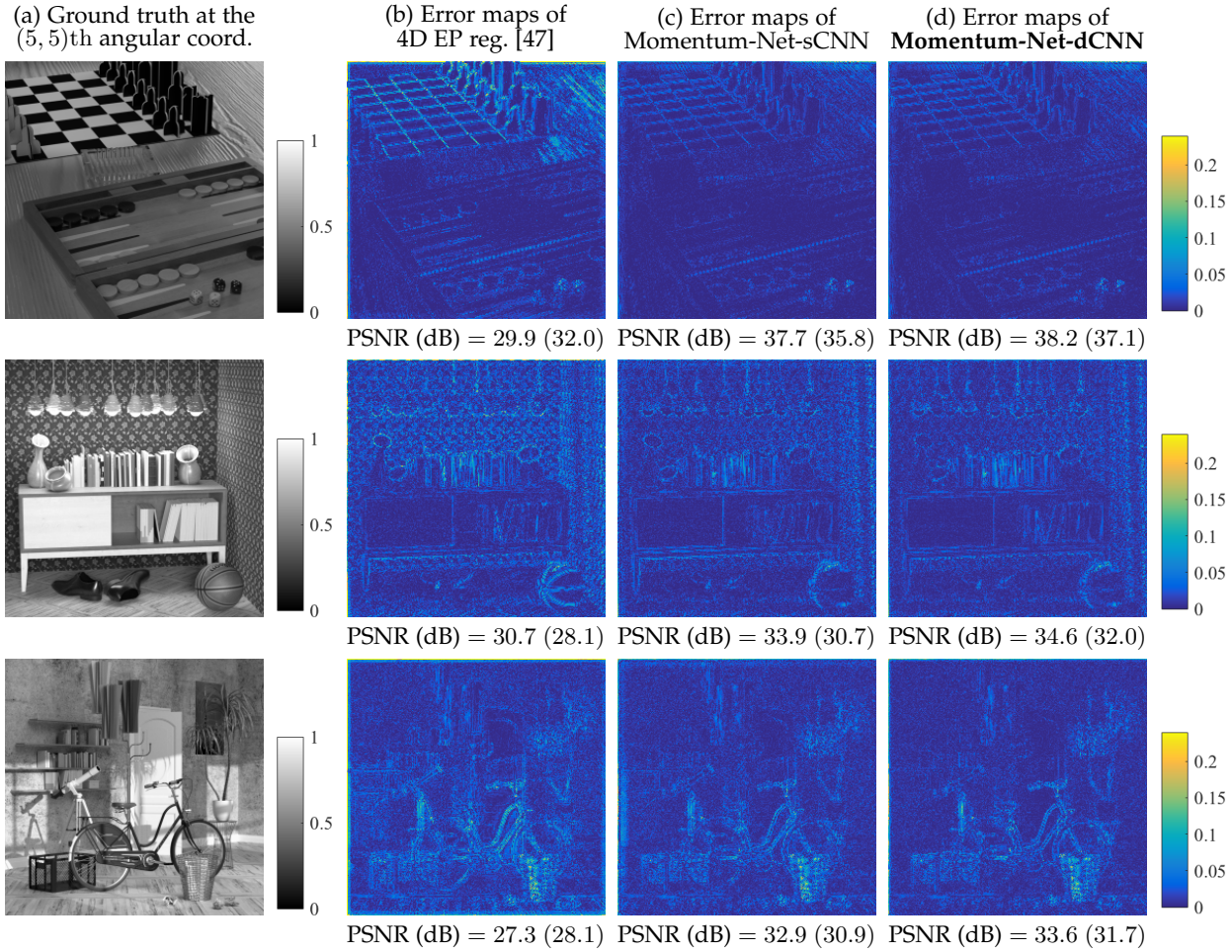


Fig. 9. Error map comparisons of reconstructed sub-aperture images from different MBIR methods in LF photography using a focal stack (LF photography systems with $C = 5$ detectors capture a focal stack of LFs consisting of $S = 81$ sub-aperture images; sub-aperture images at the $(5,5)$th angular coordinate; the PSNR values in parenthesis were measured from reconstructed LFs). See also Fig. A.2.

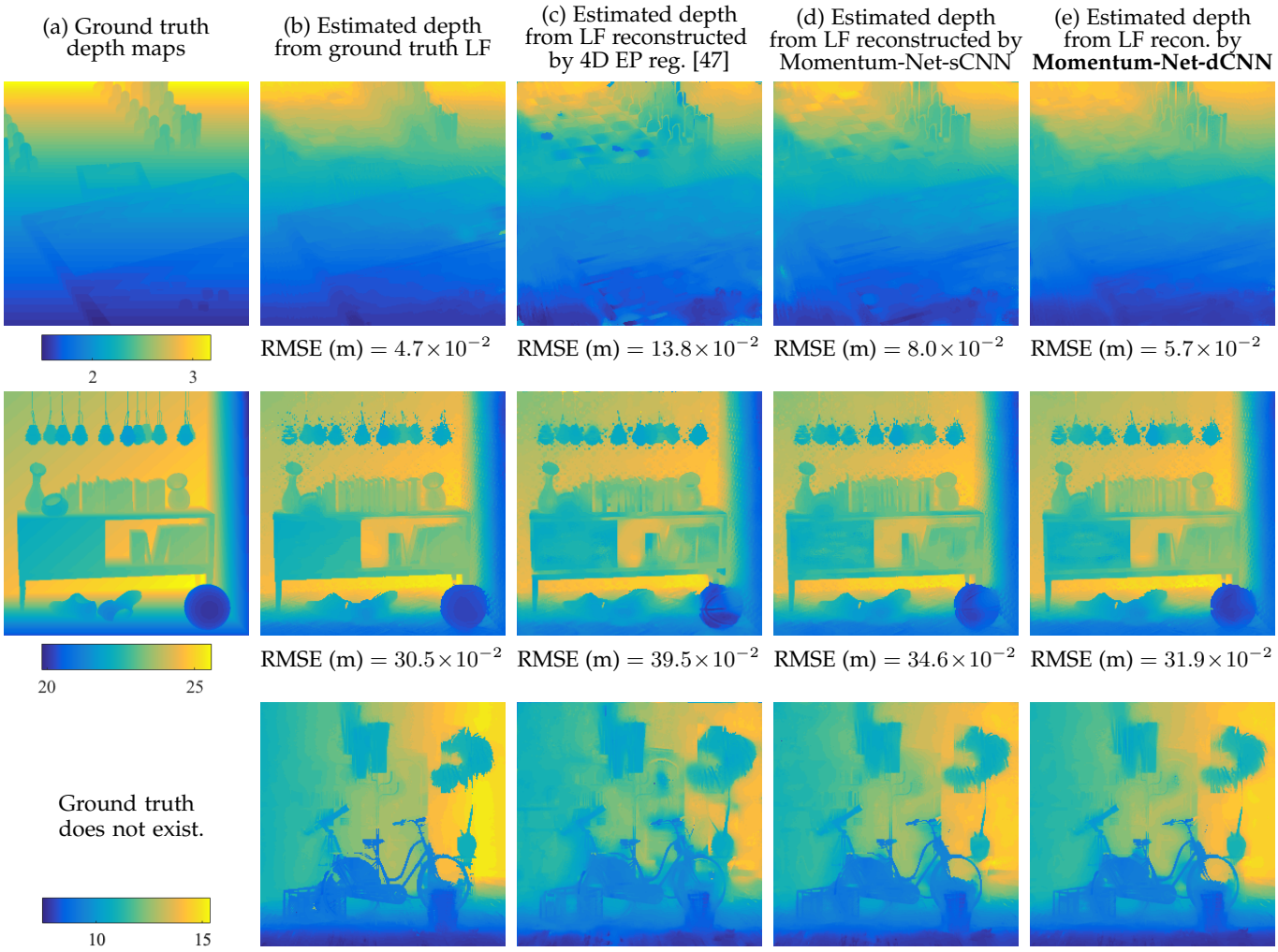


Fig. 10. Comparisons of estimated depths from LFs reconstructed by different MBIR methods in LF photograph using a focal stack (LF photography systems with $C = 5$ detectors capture a focal stack of LFs consisting of $S = 81$ sub-aperture images; SPO depth estimation [75] was applied to reconstructed LFs in Fig. 9; display window in meters). See also Fig. A.2.

[15] M. Gupta, A. Jauhari, K. Kulkarni, S. Jayasuriya, A. C. Molnar, and P. K. Turaga, "Compressive light field reconstructions using deep learning," in *Proc. IEEE CVPR Workshops*, Honolulu, HI, Jul. 2017, pp. 1277–1286.

[16] I. Y. Chun, X. Zheng, Y. Long, and J. A. Fessler, "BCD-Net for low-dose CT reconstruction: Acceleration, convergence, and generalization," in *Proc. Med. Image Computing and Computer Assist. Interven.*, Shenzhen, China, Oct. 2019, pp. 31–40.

[17] 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, Feb. 2019. [Online]. Available: http://arxiv.org/abs/1711.00905

[18] H. Lim, I. Y. Chun, Y. K. Dewaraja, and J. A. Fessler, "Improved low-count quantitative PET reconstruction with an iterative neural network," *IEEE Trans. Med. Imag.* (to appear), May 2020. [Online]. Available: http://arxiv.org/abs/1906.02327

[19] S. Ye, Y. Long, and I. Y. Chun, "Momentum-net for low-dose CT image reconstruction," *arXiv preprint eess.IV:2002.12018*, Feb. 2020.

[20] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-Net for compressive sensing MRI," in *Proc. NIPS*, Long Beach, CA, Dec. 2016, pp. 10–18.

[21] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[22] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM J. Imaging Sci.*, vol. 10, no. 4, pp. 1804–1844, Oct. 2017.

[23] G. T. Buzzard, S. H. Chan, S. Sreehari, and C. A. Bouman, "Plug-and-play unplugged: Optimization free reconstruction using consensus equilibrium," *SIAM J. Imaging Sci.*, vol. 11, no. 3, pp. 2001–2020, Sep. 2018.

[24] I. Y. Chun and J. A. Fessler, "Convolutional analysis operator learning: Acceleration and convergence," *IEEE Trans. Image Process.*, vol. 29, pp. 2108–2122, 2020.

[25] ——, "Deep BCD-net using identical encoding-decoding CNN structures for iterative image recovery," in *Proc. IEEE IVMSP Workshop*, Zagori, Greece, Jun. 2018, pp. 1–5.

[26] M. Mardani, Q. Sun, D. Donoho, V. Papyan, H. Monajemi, S. Vasanawala, and J. Pauly, "Neural proximal gradient descent for compressive imaging," in *Proc. NIPS*, Montreal, Canada, Dec. 2018, pp. 9573–9583.

[27] I. Y. Chun, H. Lim, Z. Huang, and J. A. Fessler, "Fast and convergent iterative signal recovery using trained convolutional neural networkss," in *Proc. Allerton Conf. on Commun., Control, and Comput.*, Allerton, IL, Oct. 2018, pp. 155–159.

[28] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, "Learning a variational network for reconstruction of accelerated MRI data," *Magn. Reson. Med.*, vol. 79, no. 6, pp. 3055–3071, Nov. 2017.

[29] M. Mardani, E. Gong, J. Y. Cheng, S. S. Vasanawala, G. Zaharchuk, L. Xing, and J. M. Pauly, "Deep generative adversarial neural networks for compressive sensing (GANCS) MRI," *IEEE Trans. Med. Imag.*, 2018.

[30] H. K. Aggarwal, M. P. Mani, and M. Jacob, "MoDL: Model based deep learning architecture for inverse problems," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 394–405, Feb. 2019.

[31] H. Gupta, K. H. Jin, H. Q. Nguyen, M. T. McCann, and M. Unser, "CNN-based projected gradient descent for consistent CT image reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1440–1453, Jun. 2018.

[32] K. Kim, D. Wu, K. Gong, J. Dutta, J. H. Kim, Y. D. Son, H. K. Kim, G. El Fakhri, and Q. Li, "Penalized PET reconstruction using deep learning prior and local linear fitting," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1478–1487, Jun. 2018.

[33] H. Lim, J. A. Fessler, Y. K. Dewaraja, and I. Y. Chun, "Application of trained Deep BCD-Net to iterative low-count PET image reconstruction," in *Proc. IEEE NSS-MIC*, Sydney, Australia, Nov. 2018, pp. 1–4.

[34] H. Lim, I. Y. Chun, J. A. Fessler, and Y. K. Dewaraja, "Improved low count quantitative SPECT reconstruction with a trained deep learning based regularizer," *J. Nuc. Med. (Abs. Book)*, vol. 60, no. supp. 1, p. 42, May 2019.

[35] 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.

[36] I. Y. Chun, D. Hong, B. Adcock, and J. A. Fessler, "Convolutional analysis operator learning: Dependence on training data," *IEEE Signal Process. Lett.*, vol. 26, no. 8, pp. 1137–1141, Jun. 2019. [Online]. Available: http://arxiv.org/abs/1902.08267

[37] I. Y. Chun and J. A. Fessler, "Convolutional analysis operator learning: Application to sparse-view CT," in *Proc. Asilomar Conf. on Signals, Syst., and Comput.*, Pacific Grove, CA, Oct. 2018, pp. 1631–1635.

[38] C. Crockett, D. Hong, I. Y. Chun, and J. A. Fessler, "Incorporating handcrafted filters in convolutional analysis operator learning for ill-posed inverse problems," in *Proc. IEEE Workshop CAMSAP*, Guadeloupe, West Indies, Dec. 2019, pp. 316–320.

[39] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Trans. Comput. Imag.*, vol. 2, no. 4, pp. 408–423, Aug. 2016.

[40] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE CVPR*, Honolulu, HI, Jul. 2017, pp. 4681–4690.

[41] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 84–98, Nov. 2017.

[42] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. & Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[43] E. T. Reehorst and P. Schniter, "Regularization by denoising: Clarifications and new interpretations," *IEEE Trans. Comput. Imag.*, vol. 5, no. 1, pp. 52–67, Mar. 2019.

[44] U. S. Kamilov, H. Mansour, and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1872–1876, Dec. 2017.

[45] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. ICLR*, Vancouver, Canada, Apr. 2018.

[46] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin, "Plug-and-play methods provably converge with properly trained denoisers," in *Proc. ICML*, Long Beach, CA, Jun. 2019, pp. 5546–5557.

[47] M.-B. Lien, C.-H. Liu, I. Y. Chun, S. Ravishankar, H. Nien, M. Zhou, J. A. Fessler, Z. Zhong, and T. B. Norris, "Ranging and light field imaging with transparent photodetectors," *Nature Photonics*, vol. 14, no. 3, pp. 143–148, Jan. 2020.

[48] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.*, vol. 140, no. 1, pp. 125–161, Aug. 2013.

[49] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.

[50] S. Ono, "Primal-dual plug-and-play image restoration," *IEEE Signal Process. Lett.*, vol. 24, no. 8, pp. 1108–1112, Aug. 2017.

[51] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[52] 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, Jul. 2007.

[53] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imaging Sci.*, vol. 6, no. 3, pp. 1758–1789, Sep. 2013.

[54] ——, "A fast patch-dictionary method for whole image recovery," *Inverse Probl. Imag.*, vol. 10, no. 2, pp. 563–583, May 2016.

[55] ——, "A globally convergent algorithm for nonconvex optimization based on block coordinate update," *J. Sci. Comput.*, vol. 72, no. 2, pp. 700–734, Aug. 2017.

[56] M. J. Muckley, D. C. Noll, and J. A. Fessler, "Fast, iterative subsampled spiral reconstruction via circulant majorizers," in *Proc. Int. Soc. Magn. Res. Med.*, Singapore, May 2016, p. 521.

[57] M. G. McGaffin and J. A. Fessler, "Algorithmic design of majorizers for large-scale inverse problems," *ArXiv preprint stat.CO/1508.02958*, Oct. 2015.

[58] I. Y. Chun and B. Adcock, "Compressed sensing and parallel acquisition," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 1–23, May 2017. [Online]. Available: http://arxiv.org/abs/1601.06214

[59] S. Aja-Fernández, G. Vegas-Sánchez-Ferrero, and A. Tristán-Vega, "Noise estimation in parallel MRI: GRAPPA and SENSE," *Magn. Reson. Imaging*, vol. 32, no. 3, pp. 281–290, Apr. 2014.

[60] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM J. Control Optm.*, vol. 14, no. 5, pp. 877–898, Aug. 1976.

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

[62] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization or nonconvex and nonsmooth problems," *Math. Program.*, vol. 146, no. 1-2, pp. 459–494, Aug. 2014.

[63] D. Gilton, G. Ongie, and R. Willett, "Neumann networks for linear inverse problems in imaging," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 328–343, Oct. 2019.

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Las Vegas, NV, June 2016.

[65] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. NIPS*, Montreal, Canada, Dec. 2018, pp. 6389–6399.

[66] Z. Li, I. Y. Chun, and Y. Long, "Image-domain material decomposition using an iterative neural network for dual-energy CT," in *Proc. IEEE ISBI*, Iowa City, IA, Apr. 2009, pp. 651–655.

[67] I. Y. Chun and T. Talavage, "Efficient compressed sensing statistical X-ray/CT reconstruction from fewer measurements," in *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med*, Lake Tahoe, CA, Jun. 2013, pp. 30–33.

[68] C. J. Blocker, I. Y. Chun, and J. A. Fessler, "Low-rank plus sparse tensor models for light-field reconstruction from focal stack data," in *Proc. IEEE IVMSP Workshop*, Zagori, Greece, Jun. 2018, pp. 1–5.

[69] D. Zhang, Z. Xu, Z. Huang, A. R. Gutierrez, I. Y. Chun, C. J. Blocker, G. Cheng, Z. Liu, J. A. Fessler, Z. Zhong, and T. B. Norris, "Graphene-based transparent photodetector array for multiplane imaging," in *Proc. Conf. on Lasers and Electro-Optics*, San Jose, CA, May 2019, p. SM4J.2.

[70] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4D light fields," in *Proc. ACCV*, Taipei, Taiwan, Nov. 2016, pp. 19–34.

[71] 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.

[72] 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.

[73] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE ICCV*, Santiago, Chile, Dec. 2015, pp. 1026–1034.

[74] M. H. Kamal, B. Heshmat, R. Raskar, P. Vandergheynst, and G. Wetzstein, "Tensor low-rank and sparse light field photography," *Comput. Vis. Image Und.*, vol. 145, pp. 172–181, Apr. 2016.

[75] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong, "Robust depth estimation for light field via spinning parallelogram operator," *Comput. Vis. Image Und.*, vol. 145, pp. 148–159, Apr. 2016.

[76] J. A. Fessler, "Michigan image reconstruction toolbox (MIRT) for Matlab," Available from http://web.eecs.umich.edu/~fessler, 2016.

[77] H. M. Hudson and R. S. Larkin, "Accelerated image reconstruction using ordered subsets of projection data," *IEEE Trans. Med. Imag.*, vol. 13, no. 4, pp. 601–609, Dec. 1994.

[78] H. Erdogan and J. A. Fessler, "Ordered subsets algorithms for transmission tomography," *Phys. Med. Biol.*, vol. 44, no. 11, p. 2835, Jul. 1999.

Momentum-Net: Fast and convergent iterative neural network for inverse problems (Appendices)

This supplementary material for [1] *a)* reviews the Block Proximal Extrapolated Gradient method using a Majorizer (BPEG-M) [2], [3], *b)* lists parameters of Momentum-Net, and summarizes selection guidelines or gives default values, *c)* compares the convergence properties between Momentum-Net and BCD-Net, and *d)* provides mathematical proofs or detailed descriptions to support several arguments in the main manuscript. We use the prefix "A" for the numbers in section, theorem, equation, figure, table, and footnote in the supplement.

A.1 BPEG-M: Review

This section explains *block multi-(non)convex* optimization problems, and summarizes the state-of-the-art method for block multi-(non)convex optimization, BPEG-M [2], [3], along with its convergence guarantees.

A.1.1 Block multi-(non)convex optimization

In a block optimization problem, the variables of the underlying optimization problem are treated either as a single block or multiple disjoint blocks. In *block multi-(non)convex* optimization, we consider the following problem:

$$\min_{u} F(u_1,\dots,u_B) \triangleq f(u_1,\dots,u_B) + \sum_{b=1}^{B} r_b(u_b) \tag{A.1}$$

where variable u is decomposed into B blocks $u_1,\dots,u_B$ ($\{u_b \in \mathbb{R}^{n_b} : b = 1,\dots,B\}$), f is assumed to be (continuously) differentiable, but functions $\{r_b : b = 1,\dots,B\}$ are not necessarily differentiable. The function r_b can incorporate the constraint $u_b \in \mathcal{U}_b$, by allowing r_b's to be extended-valued, e.g., $r_b(u_b) = \infty$ if $u_b \notin \mathcal{U}_b$, for $b = 1,\dots,B$. It is standard to assume that both f and $\{r_b\}$ are proper and closed, and the sets $\{\mathcal{U}_b\}$ are closed. We consider either that (A.1) has block-wise convexity (but (A.1) is jointly nonconvex in general) [2], [4] or that f, $\{r_b\}$, or $\{\mathcal{U}_b\}$ are not necessarily convex [3], [5]. Importantly, r_b can include (non)convex and nonsmooth ℓ^p (quasi-)norm, $p \in [0,1]$. The next section introduces our optimization framework that solves (A.1).

The following sections review BPEG-M [2], [3], the state-of-the-art optimization framework for solving block multi-(non)convex problems, when used with sufficiently sharp majorizers. BPEG-M uses block-wise extrapolation, majorization, and proximal mapping. By using a more general Lipschitz continuity (see Definition 1) for block-wise gradients, BPEG-M is particularly useful for rapidly calculating majorizers involved with large-scale problems, and successfully applied to some large-scale machine learning and computational imaging problems; see [2], [3], [6] and references therein.

A.1.2 BPEG-M

This section summarizes the BPEG-M framework. Using Definition 1 and Lemma 2, the proposed method, BPEG-M, is given as follows. To solve (A.1), we minimize majorizers of F cyclically over each block $u_1,\dots,u_B$, while fixing the remaining blocks at their previously updated variables. Let $u_b^{(i+1)}$ be the value of u_b after its ith update, and define

$$f_b^{(i+1)}(u_b) \triangleq f\Big(u_1^{(i+1)},\dots,u_{b-1}^{(i+1)},u_b,u_{b+1}^{(i)},\dots,u_B^{(i)}\Big),$$

for all b,i. At the bth block of the ith iteration, we apply Lemma 2 to functional $f_b^{(i+1)}(u_b)$ with a $M^{(i+1)}$-Lipschitz continuous gradient at the extrapolated point $\acute{u}_b^{(i+1)}$, and minimize a majorized function. In other words, we consider the updates

$$\begin{aligned} u_b^{(i+1)} &= \operatorname*{argmin}_{u_b} \langle \nabla f_b^{(i+1)}(\acute{u}_b^{(i+1)}), u_b - \acute{u}_b^{(i+1)}\rangle + \frac{1}{2}\Big\|u_b - \acute{u}_b^{(i+1)}\Big\|^2_{\widetilde{M}_b^{(i+1)}} + r_b(u_b) \\ &= \operatorname{Prox}_{r_b}^{\widetilde{M}_b^{(i+1)}}\Big(\underbrace{\acute{u}_b^{(i+1)} - \big(\widetilde{M}_b^{(i+1)}\big)^{-1}\nabla f_b^{(i+1)}(\acute{u}_b^{(i+1)})}_{\textit{extrapolated gradient step using a majorizer of } f_b^{(i+1)}}\Big), \end{aligned} \tag{A.2}$$

where

$$\acute{u}_b^{(i+1)} = u_b^{(i)} + E_b^{(i+1)}\big(u_b^{(i)} - u_b^{(i-1)}\big), \tag{A.3}$$

Algorithm A.1 BPEG-M [2], [3]

Require: $\{u_b^{(0)} = u_b^{(-1)} : \forall b\}$, $\{w_b^{(i)} \in [0,1], \forall b, i\}$, $i = 0$
  while a stopping criterion is not satisfied **do**
    for $b = 1, \ldots, B$ **do**
      Calculate $\widetilde{M}_b^{(i+1)}$ by (A.4), and $E_b^{(i+1)}$ to satisfy (A.5) or (A.6)
      $\acute{u}_b^{(i+1)} = u_b^{(i)} + E_b^{(i+1)}\big(u_b^{(i)} - u_b^{(i-1)}\big)$
      $u_b^{(i+1)} = \mathrm{Prox}_{r_b}^{\widetilde{M}_b^{(i+1)}}\Big(\acute{u}_b^{(i+1)} - \big(\widetilde{M}_b^{(i+1)}\big)^{-1}\nabla f_b^{(i+1)}(\acute{u}_b^{(i+1)})\Big)$
    end for
    $i = i + 1$
  end while

the proximal operator is defined by (2), $\nabla f_b^{(i+1)}(\acute{u}_b^{(i+1)})$ is the block-partial gradient of f at $\acute{u}_b^{(i+1)}$, a *scaled majorization matrix* is given by

$$\widetilde{M}_b^{(i+1)} = \lambda_b \cdot M_b^{(i+1)} \succ 0, \qquad \lambda_b \geq 1, \tag{A.4}$$

and $M_b^{(i+1)} \in \mathbb{R}^{n_b \times n_b}$ is a symmetric positive definite *majorization matrix* of $\nabla f_b^{(i+1)}(u_b)$. In (A.3), the $\mathbb{R}^{n_b \times n_b}$ matrix $E_b^{(i+1)} \succeq 0$ is an *extrapolation matrix* that accelerates convergence in solving block multi-convex problems [2]. We design it to satisfy conditions (A.5) or (A.6) below. In (A.4), $\{\lambda_b = 1 : \forall b\}$ and $\{\lambda_b > 1 : \forall b\}$, for block multi-convex and block multi-nonconvex problems, respectively.

For some $f_b^{(i+1)}$ having sharp majorizers, we expect that extrapolation (A.3) has no benefits in accelerating convergence, and use $\{E_b^{(i+1)} = 0 : \forall i\}$. Other than the blocks having sharp majorizers, one can apply some increasing momentum coefficient formula [7], [8] to the corresponding extrapolation matrices. The choice in [2]–[4] accelerated BPEG-M for some machine learning and data science applications. Algorithm A.1 summarizes these updates.

A.1.3 Convergence results

This section summarizes convergence results of Algorithm A.1 under the following assumptions:

- *Assumption S.1)* In (A.1), F is proper and lower bounded in $\mathrm{dom}(F) \triangleq \{u : F(u) < \infty\}$. In addition,
  for *block multi-convex* (A.1), f is differentiable and (A.1) has a Nash point or block-coordinate minimizer[A.1] (see its definition in [4, (2.3)–(2.4)]);
  for *block multi-nonconvex* (A.1), f is continuously differentiable, r_b is lower semicontinuous[A.2], $\forall b$, and (A.1) has a critical point $u^\star$ that satisfies $0 \in \partial F(u^\star)$.
- *Assumption S.2)* $\nabla f_b^{(i+1)}(u_b)$ is M-Lipschitz continuous with respect to u_b, i.e.,

$$\Big\|\nabla f_b^{(i+1)}(u) - \nabla f_b^{(i+1)}(v)\Big\|_{\left(M_b^{(i+1)}\right)^{-1}} \leq \|u - v\|_{M_b^{(i+1)}},$$

  for $u, v \in \mathbb{R}^{n_b}$, where $M_b^{(i+1)}$ is a bounded majorization matrix.
- *Assumption S.3)* The extrapolation matrices $E_b^{(i+1)} \succeq 0$ satisfy that

$$\text{for } \textit{block multi-convex} \text{ (A.1)}, \ \big(E_b^{(i+1)}\big)^T M_b^{(i+1)} E_b^{(i+1)} \preceq \delta^2 \cdot M_b^{(i)}; \tag{A.5}$$

$$\text{for } \textit{block multi-nonconvex} \text{ (A.1)}, \ \big(E_b^{(i+1)}\big)^T M_b^{(i+1)} E_b^{(i+1)} \preceq \frac{\delta^2(\lambda_b - 1)^2}{4(\lambda_b + 1)^2} \cdot M_b^{(i)}, \tag{A.6}$$

  with $\delta < 1$, $\forall b, i$.

Theorem A.1 (*Block multi-convex* (A.1): A limit point is a Nash point [2])**.** *Under Assumptions S.1–S.3, let* $\{u^{(i+1)} : i \geq 0\}$ *be the sequence generated by Algorithm A.1. Then any limit point of* $\{u^{(i+1)} : i \geq 0\}$ *is a Nash point of (A.1).*

Theorem A.2 (*Block multi-nonconvex* (A.1): A limit point is a critical point [3])**.** *Under Assumptions S.1–S.3, let* $\{u^{(i+1)} : i \geq 0\}$ *be the sequence generated by Algorithm A.1. Then any limit point of* $\{u^{(i+1)} : i \geq 0\}$ *is a critical point of (A.1).*

Remark A.3. Theorems A.1–A.2 imply that, if there exists a critical point for (A.1), i.e., $0 \in \partial F(u^\star)$, then any limit point of $\{u^{(i+1)} : i \geq 0\}$ is a critical point. One can further show global convergence under some conditions: if $\{u^{(i+1)} : i \geq 0\}$ is bounded and the critical points are isolated, then $\{u^{(i+1)} : i \geq 0\}$ converges to a critical point [2, Rem. 3.4], [4, Cor. 2.4].

A.1.4 Application of BPEG-M to solving block multi-(non)convex problem (1)

For update (3), we do not use extrapolation, i.e., (A.3), since the corresponding majorization matrices are sharp, so one obtains tight majorization bounds in Lemma 2. See, for example, [3, §V-B]. For updates (3) and (5), we rewrite $\sum_{k=1}^K \|h_k * x - \zeta_k\|_2^2$ as $\|x - \sum_{k=1}^K \mathrm{flip}(h_k^*) * \zeta_k\|_2^2$ by using the TF condition in §2.1 [3, §VI], [6].

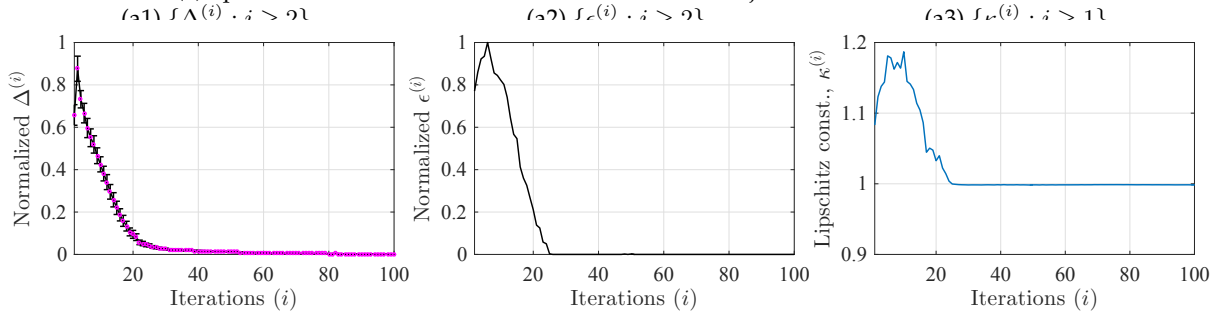
A.1. Given a feasible set $\mathcal{U}$, a point $u^\star \in \mathrm{dom}(F) \cup \mathcal{U}$ is a critical point (or stationary point) of F if the directional derivative $d^T \nabla F(u^\star) \geq 0$ for any feasible direction d at $u^\star$. If $u^\star$ is an interior point of $\mathcal{U}$, then the condition is equivalent to $0 \in \partial F(u^\star)$.

A.2. F is lower semicontinuous at point u_0 if $\liminf_{u \to u_0} F(u) \geq F(u_0)$.

A.2 Empirical measures related to the convergence of Momentum-Net using sCNN refiners

This section provides empirical measures related to Assumption 4 for Momentum-Net using single-hidden layer autoencoders (18); see Fig. A.1 below. We estimated the sequence $\{\epsilon^{(i)} : i = 2, \dots, N_{\text{lyr}}\}$ in Definition 7, the sequence $\{\Delta^{(i)} : i = 2, \dots, N_{\text{lyr}}\}$ in Definition 8, and the Lipschitz constants $\{\kappa^{(i)} : i = 1, \dots, N_{\text{lyr}}\}$ of refining NNs $\{\mathcal{R}_{\theta^{(i)}} : \forall i\}$, based on a hundred sets of randomly selected training samples related with the corresponding bounds of the measures, e.g., u and v in (11) are training input to $\mathcal{R}_{\theta^{(i+1)}}$ and $\mathcal{R}_{\theta^{(i)}}$ in (Alg.1.1), respectively.

(a) Sparse-view CT: Condition numbers of data-fit majorizers have *mild* variations.



(b) LF photography using focal stack: Condition numbers of data-fit majorizers have *strong* variations.

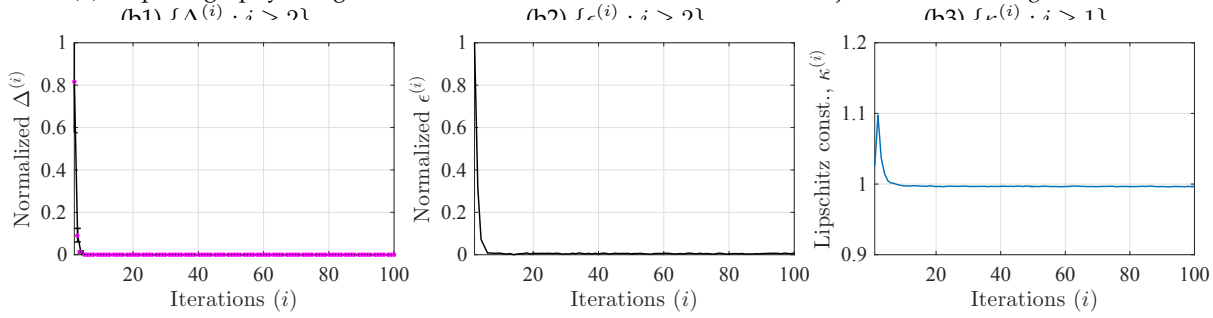


Fig. A.1. Empirical measures related to Assumption 4 for guaranteeing convergence of Momentum-Net using sCNN refiners (for details, see (18) and §4.2.1), in different applications. (a) The sparse-view CT reconstruction experiment used fan-beam geometry with 12.5% projections views. (b) The LF photography experiment used five detectors and reconstructed LFs consisting of 9×9 sub-aperture images. (a1, b1) For both the applications, we observed that $\Delta^{(i)} \to 0$. This implies that the $z^{(i+1)}$-updates in (Alg.1.1) satisfy the asymptotic block-coordinate minimizer condition in Assumption 4. (Magenta dots denote the mean values and black vertical error bars denote standard deviations.) (a2) Momentum-Net trained from training data-fits, where their majorization matrices have *mild* condition number variations, shows that $\epsilon^{(i)} \to 0$. This implies that paired NNs $(\mathcal{R}_{\theta^{(i+1)}}, \mathcal{R}_{\theta^{(i)}})$ in (Alg.1.1) are asymptotically nonexpansive. (b2) Momentum-Net trained from training training data-fits, where their majorization matrices have *mild* condition number variations, shows that $\epsilon^{(i)}$ becomes close to zero, but does not converge to zero in one hundred iterations. (a3, b3) The NNs, $\mathcal{R}_{\theta^{(i+1)}}$ in (Alg.1.1), become nonexpansive, i.e., its Lipschitz constant $\kappa^{(i)}$ becomes less than 1, as i increases.

A.3 Probabilistic justification for the asymptotic block-coordinate minimizer condition in Assumption 4

This section introduces a useful result for an asymptotic block-coordinate minimizer $z^{(i+1)}$: the following lemma provides a *probabilistic* bound for $\|x^{(i)} - z^{(i+1)}\|_2^2$ in (12), given a subgaussian vector $z^{(i+1)} - z^{(i)}$ with independent and zero-mean entries.

Lemma A.4 (Probabilistic bounds for $\|x^{(i)} - z^{(i+1)}\|_2^2$)**.** *Assume that $z^{(i+1)} - z^{(i)}$ is a zero-mean subgaussian vector of which entries are independent and zero-mean subgaussian variables. Then, each bound in (12) holds with probability at least*

$$1 - \exp\left(\frac{-\left(\|z^{(i+1)} - z^{(i)}\|_2^2 + \Delta^{(i+1)} \right)^2}{8\rho \cdot \sigma^{(i+1)} \cdot \|\mathcal{R}_{\theta^{(i+1)}}(x^{(i)}) - x^{(i)}\|_2^2} \right),$$

where $\sigma^{(i+1)}$ is a subgaussian parameter for $z^{(i+1)} - z^{(i)}$, and a random variable is subgaussian with parameter σ if $\mathbb{P}\{|\cdot| \geq t\} \leq 2\exp(-\frac{t^2}{2\sigma})$ for $t \geq 0$.

Proof. First, observe that

$$\begin{aligned} \left\| x^{(i)} - z^{(i+1)} \right\|_2^2 &= \left\| x^{(i)} - z^{(i)} - (z^{(i+1)} - z^{(i)}) \right\|_2^2 \\ &= \left\| x^{(i)} - z^{(i)} \right\|_2^2 + \left\| z^{(i+1)} - z^{(i)} \right\|_2^2 - 2\langle x^{(i)} - z^{(i)}, z^{(i+1)} - z^{(i)} \rangle \end{aligned}$$

$$= \left\|x^{(i)} - z^{(i)}\right\|_2^2 + \left\|z^{(i+1)} - z^{(i)}\right\|_2^2 - 2\langle z^{(i+1)} - z^{(i)} + \rho(x^{(i)} - \mathcal{R}_{\theta^{(i+1)}}(x^{(i)})), z^{(i+1)} - z^{(i)}\rangle \tag{A.7}$$

$$= \left\|x^{(i)} - z^{(i)}\right\|_2^2 - \left\|z^{(i+1)} - z^{(i)}\right\|_2^2 + 2\rho\langle \mathcal{R}_{\theta^{(i+1)}}(x^{(i)}) - x^{(i)}, z^{(i+1)} - z^{(i)}\rangle \tag{A.8}$$

where the inequality (A.7) holds by $x^{(i)} = \rho x^{(i)} - \rho\mathcal{R}_{\theta^{(i+1)}} + z^{(i+1)}$ via (Alg.1.1). We now obtain a probablistic bound for the third quantity in (A.8) via a concentration inequality. The concentration inequality on the sum of independent zero-mean subgaussian variables (e.g., [9, Thm. 7.27]) yields that for any $t^{(i+1)} \geq 0$

$$\mathbb{P}\left\{\langle \mathcal{R}_{\theta^{(i+1)}}(x^{(i)}) - x^{(i)}, z^{(i+1)} - z^{(i)}\rangle \geq t^{(i+1)}\right\} \leq \exp\left(-\frac{(t^{(i+1)})^2}{2\sigma^{(i+1)}\|\mathcal{R}_{\theta^{(i+1)}}(x^{(i)}) - x^{(i)}\|_2^2}\right) \tag{A.9}$$

where $\sigma^{(i+1)}$ is given as in Lemma A.4. Applying the result (A.9) with $t^{(i+1)} = \frac{1}{2\rho}(\|z^{(i+1)} - z^{(i)}\|_2^2 + \Delta^{(i+1)})$ to the bound (A.8) completes the proofs. □

Lemma A.4 implies that, given sufficiently large $\Delta^{(i+1)}$, or sufficiently small $\sigma^{(i+1)}$ (e.g., variance for a Gaussian random vector $z^{(i+1)} - z^{(i)}$) or $\|\mathcal{R}_{\theta^{(i+1)}}(x^{(i)}) - x^{(i)}\|_2^2$, bound (12) is satisfied with high probability, for each i. In particular, $\Delta^{(i+1)}$ can be large for the first several iterations; if paired operators $(\mathcal{R}_{\theta^{(i+1)}}, \mathcal{R}_{\theta^{(i)}})$ in (Alg.1.1) map their input images to similar output images (e.g., the trained NNs $\mathcal{R}_{\theta^{(i+1)}}$ and $\mathcal{R}_{\theta^{(i)}}$ have good refining capabilities for $x^{(i)}$ and $x^{(i-1)}$), then $\sigma^{(i+1)}$ is small; if the regularization parameter γ in (Alg.1.3) is sufficiently large, then $\|\mathcal{R}_{\theta^{(i+1)}}(x^{(i)}) - x^{(i)}\|_2^2$ is small.

A.4 PROOFS OF PROPOSITION 9

First, we show that $\sum_{i=0}^{\infty}\|x^{(i+1)} - x^{(i)}\|_2^2 < \infty$ for convex and nonconvex $F(x; y, z^{(i+1)})$ cases.

- *Convex $F(x; y, z^{(i+1)})$ case*: Using Assumption 2 and $\{\widetilde{M}^{(i+1)} = M^{(i+1)} : \forall i\}$ for the convex case via (7), we obtain the following results for any $\mathcal{X}$:

$$F\left(x^{(i)}; y, z^{(i)}\right) - F\left(x^{(i+1)}; y, z^{(i+1)}\right) + \gamma\Delta^{(i+1)}$$

$$\geq F\left(x^{(i)}; y, z^{(i+1)}\right) - F\left(x^{(i+1)}; y, z^{(i+1)}\right) \tag{A.10}$$

$$\geq \frac{1}{2}\left\|x^{(i+1)} - \acute{x}^{(i+1)}\right\|_{M^{(i+1)}}^2 + \left(\acute{x}^{(i+1)} - x^{(i)}\right)^T M^{(i+1)}\left(x^{(i+1)} - \acute{x}^{(i+1)}\right) \tag{A.11}$$

$$= \frac{1}{2}\left\|x^{(i+1)} - x^{(i)}\right\|_{M^{(i+1)}}^2 - \frac{1}{2}\left\|E^{(i+1)}\left(x^{(i)} - x^{(i-1)}\right)\right\|_{M^{(i+1)}}^2 \tag{A.12}$$

$$\geq \frac{1}{2}\left\|x^{(i+1)} - x^{(i)}\right\|_{M^{(i+1)}}^2 - \frac{\delta^2}{2}\left\|x^{(i)} - x^{(i-1)}\right\|_{M^{(i)}}^2 \tag{A.13}$$

where the inequality (A.10) uses the condition (12) in Assumption 4, the inequality (A.11) is obtained by using the results in [2, Lem. S.1], the equality (A.12) uses the extrapolation formula (Alg.1.2) and the symmetry of $M^{(i+1)}$, the inequality (A.13) holds by Assumption 3.
Summing the inequality of $F(x^{(i)}; y, z^{(i)}) - F(x^{(i+1)}; y, z^{(i+1)}) + \gamma\Delta^{(i+1)}$ in (A.13) over $i = 0, \ldots, N_{\text{lyr}} - 1$, we obtain

$$\begin{aligned} F\left(x^{(0)}; y, z^{(0)}\right) - F\left(x^{(N_{\text{lyr}})}; y, z^{(N_{\text{lyr}})}\right) + \gamma\sum_{i=0}^{N_{\text{lyr}}-1}\Delta^{(i+1)} &\geq \sum_{i=0}^{N_{\text{lyr}}-1}\frac{1}{2}\left\|x^{(i+1)} - x^{(i)}\right\|_{M^{(i+1)}}^2 - \frac{\delta^2}{2}\left\|x^{(i)} - x^{(i-1)}\right\|_{M^{(i)}}^2 \\ &\geq \sum_{i=0}^{N_{\text{lyr}}-1}\frac{1-\delta^2}{2}\left\|x^{(i+1)} - x^{(i)}\right\|_{M^{(i+1)}}^2 \\ &\geq \sum_{i=0}^{N_{\text{lyr}}-1}\frac{m_{F,\min}(1-\delta^2)}{2}\left\|x^{(i+1)} - x^{(i)}\right\|_2^2 \end{aligned} \tag{A.14}$$

where the inequality (A.14) holds by Assumption 2. Due to the lower boundedness of $F(x; y, z)$ in Assumption 1 and the summability of $\{\Delta^{(i+1)} \geq 0 : \forall i\}$ in Assumption 4, taking $N_{\text{lyr}} \to \infty$ gives

$$\sum_{i=0}^{\infty}\left\|x^{(i+1)} - x^{(i)}\right\|_2^2 < \infty. \tag{A.15}$$

- *Nonconvex $F(x; y, z^{(i+1)})$ case*: Using Assumption 2, we obtain the following results without assuming that $F(x; y, z^{(i+1)})$ is convex:

$$F\left(x^{(i)}; y, z^{(i)}\right) - F\left(x^{(i+1)}; y, z^{(i+1)}\right) + \gamma\Delta^{(i+1)}$$

$$\geq F\left(x^{(i)}; y, z^{(i+1)}\right) - F\left(x^{(i+1)}; y, z^{(i+1)}\right) \tag{A.16}$$

$$\geq \frac{\lambda-1}{4}\left\|x^{(i+1)}-x^{(i)}\right\|^2_{M^{(i+1)}} - \frac{(\lambda+1)^2}{\lambda-1}\left\|x^{(i)}-\acute{x}_b^{(i+1)}\right\|^2_{M^{(i+1)}} \tag{A.17}$$

$$= \frac{\lambda-1}{4}\left\|x^{(i+1)}-x^{(i)}\right\|^2_{M^{(i+1)}} - \frac{(\lambda+1)^2}{\lambda-1}\left\|E^{(i+1)}\left(x^{(i)}-x^{(i-1)}\right)\right\|^2_{M^{(i+1)}} \tag{A.18}$$

$$\geq \frac{\lambda-1}{4}\left(\left\|x^{(i+1)}-x^{(i)}\right\|^2_{M^{(i+1)}} - \delta^2\left\|x^{(i)}-x^{(i-1)}\right\|^2_{M^{(i)}}\right) \tag{A.19}$$

where the inequality (A.16) uses the condition (12) in Assumption 4, the inequality (A.17) use the results in [3, §S.3], the equality (A.18) holds by (Alg.1.2), the inequality (A.19) is obtained by Assumption 3.
Summing the inequality of $F(x^{(i)};y,z^{(i)})-F(x^{(i+1)};y,z^{(i+1)})+\gamma\Delta^{(i+1)}$ in (A.19) over $i=0,\ldots,N_{\text{lyr}}-1$, we obtain

$$\begin{aligned}
F\Big(x^{(0)};y,z^{(0)}\Big)-F\Big(x^{(N_{\text{lyr}})};y,z^{(N_{\text{lyr}})}\Big)+\gamma\cdot\sum_{i=0}^{N_{\text{lyr}}-1}\Delta^{(i+1)} &\geq \sum_{i=0}^{N_{\text{lyr}}-1}\frac{\lambda-1}{4}\left(\left\|x^{(i+1)}-x^{(i)}\right\|^2_{M^{(i+1)}}-\delta^2\left\|x^{(i)}-x^{(i-1)}\right\|^2_{M^{(i)}}\right)\\
&\geq \sum_{i=0}^{N_{\text{lyr}}-1}\frac{(\lambda-1)(1-\delta^2)}{2}\left\|x^{(i+1)}-x^{(i)}\right\|^2_{M^{(i+1)}}\\
&\geq \sum_{i=0}^{N_{\text{lyr}}-1}\frac{m_{F,\min}(\lambda-1)(1-\delta^2)}{2}\left\|x^{(i+1)}-x^{(i)}\right\|^2_2,
\end{aligned}$$

where we follow the arguments in obtaining (A.14) above. Again, using the lower boundedness of $F(x;y,z)$ and the summability of $\{\Delta^{(i+1)}\geq 0:\forall i\}$, taking $N_{\text{lyr}}\to\infty$ gives the result (A.15) for nonconvex $F(x;y,z^{(i+1)})$.

Second, we show that $\sum_{i=0}^{\infty}\|z^{(i+1)}-z^{(i)}\|_2^2<\infty$. Observe

$$\begin{aligned}
\left\|z^{(i+1)}-z^{(i)}\right\|^2_2 &= \left\|(1-\rho)\Big(x^{(i)}-x^{(i-1)}\Big)+\rho\Big(\mathcal{R}_{\theta^{(i+1)}}(x^{(i)})-\mathcal{R}_{\theta^{(i)}}(x^{(i-1)})\Big)\right\|^2_2\\
&\leq (1-\rho)\left\|x^{(i)}-x^{(i-1)}\right\|^2_2+\rho\left\|\mathcal{R}_{\theta^{(i+1)}}(x^{(i)})-\mathcal{R}_{\theta^{(i)}}(x^{(i-1)})\right\|^2_2\\
&\leq \left\|x^{(i)}-x^{(i-1)}\right\|^2_2+\rho\epsilon^{(i+1)}
\end{aligned}\tag{A.20}$$

where the first equality uses the image mapping formula in (Alg.1.1), the first inequality holds by applying Jensen's inequality to the (convex) squared ℓ^2-norm, the second inequality is obtained by using the asymptotically non-expansiveness of the paired operators $(\mathcal{R}_{\theta^{(i+1)}},\mathcal{R}_{\theta^{(i)}})$ in Assumption 4. Summing the inequality of $\|z^{(i+1)}-z^{(i)}\|_2^2$ in (A.20) over $i=0,\ldots,N_{\text{lyr}}-1$, we obtain

$$\sum_{i=0}^{N_{\text{lyr}}-1}\left\|z^{(i+1)}-z^{(i)}\right\|^2_2\leq\sum_{i=0}^{N_{\text{lyr}}-2}\left\|x^{(i+1)}-x^{(i)}\right\|^2_2+\rho\sum_{i=0}^{N_{\text{lyr}}-1}\epsilon^{(i+1)},\tag{A.21}$$

where we used $x^{(0)}=x^{(-1)}$ as given in Algorithm 1. By taking $N_{\text{lyr}}\to\infty$ in (A.21), using result (A.15), and the summability of the sequence $\{\epsilon^{(i+1)}:i\geq 0\}$, we obtain

$$\sum_{i=0}^{\infty}\left\|z^{(i+1)}-z^{(i)}\right\|^2_2<\infty.\tag{A.22}$$

Combining the results in (A.15) and (A.22) completes the proofs.

A.5 PROOFS OF THEOREM 10

Let $\bar{x}$ be a limit point of $\{x^{(i)}:i\geq 0\}$ and $\{x^{(i_j)}\}$ be the subsequence converging to $\bar{x}$. Let $\bar{z}$ be a limit point of $\{z^{(i)}:i\geq 0\}$ and $\{z^{(i_j)}\}$ be the subsequence converging to $\bar{z}$. The closedness of $\mathcal{X}$ implies that $\bar{x}\in\mathcal{X}$. Using the results in Proposition 9, $\{x^{(i_j+1)}\}$ and $\{z^{(i_j+1)}\}$ also converge to $\bar{x}$ and $\bar{z}$, respectively. Taking another subsequence if necessary, the subsequence $\{M^{(i_j+1)}\}$ converges to some $\bar{M}$, since $M^{(i+1)}$ is bounded by Assumption 2. The subsequences $\{\theta^{(i_j+1)}\}$ converge to some $\bar{\theta}$, since $x^{(i_j+1)}\to\bar{x}$, $z^{(i_j+1)}\to\bar{z}$, and $\{\theta^{(i+1)}\}$ is bounded via Assumption 4.

Next, we show that the convex proximal minimization (A.23) below is continuous in the sense that the output point $x^{(i_j+1)}$ continuously depends on the input points $\acute{x}^{(i_j+1)}$ and $z^{(i_j+1)}$, and majorization matrix $\widetilde{M}^{(i_j+1)}$:

$$\begin{aligned}
x^{(i_j+1)} &= \operatorname*{argmin}_{x\in\mathcal{X}}\langle\nabla F(\acute{x}^{(i_j+1)};y,z^{(i_j+1)}),x-\acute{x}^{(i_j+1)}\rangle+\frac{1}{2}\left\|x-\acute{x}^{(i_j+1)}\right\|^2_{\widetilde{M}^{(i_j+1)}}\\
&= \operatorname{Prox}^{\widetilde{M}^{(i_j+1)}}_{\mathbb{I}_{\mathcal{X}}}\Big(\acute{x}^{(i_j+1)}-\big(\widetilde{M}^{(i_j+1)}\big)^{-1}\nabla F(\acute{x}^{(i_j+1)};y,z^{(i_j+1)})\Big).
\end{aligned}\tag{A.23}$$

where the proximal mapping operator $\operatorname{Prox}^{\widetilde{M}^{(i_j+1)}}_{\mathbb{I}_{\mathcal{X}}}(\cdot)$ is given as in (2). We consider the two cases of majorization matrices $\{M^{(i+1)}\}$ given in Theorem 10:

- For a sequence of diagonal majorization matrices, i.e., $\{M^{(i+1)} : i \geq 0\}$, one can obtain the continuity of the convex proximal minimization (A.23) with respect to $\acute{x}^{(i_j+1)}$, $z^{(i_j+1)}$, and $\widetilde{M}^{(i_j+1)}$, by extending the existing results in [10, Thm. 2.26], [11] with the separability of (A.23) to element-wise optimization problems.
- For a fixed general majorization matrix, i.e., $M = M^{(i+1)}$, $\forall i$, we obtain that the convex proximal minimization (A.24) below is continuous with respect to the input points $\acute{x}^{(i_j+1)}$ and $z^{(i_j+1)}$:

$$x^{(i_j+1)} = \underset{x\in\mathcal{X}}{\operatorname{argmin}} \langle \nabla F(\acute{x}^{(i_j+1)}; y, z^{(i_j+1)}), x - \acute{x}^{(i_j+1)}\rangle + \frac{1}{2}\Big\|x - \acute{x}^{(i_j+1)}\Big\|^2_{\widetilde{M}} \tag{A.24}$$

$$= \operatorname{Prox}^{\widetilde{M}}_{\mathbb{I}_{\mathcal{X}}}\big(\acute{x}^{(i_j+1)} - \widetilde{M}^{-1}\nabla F(\acute{x}^{(i_j+1)}; y, z^{(i_j+1)})\big)$$

$$= \big(\mathrm{Id} + \widetilde{M}^{-1}\hat{\partial}\mathbb{I}_{\mathcal{X}}\big)^{-1}\big(\acute{x}^{(i_j+1)} - \widetilde{M}^{-1}\nabla F(\acute{x}^{(i_j+1)}; y, z^{(i_j+1)})\big) \tag{A.25}$$

where $\hat{\partial} f(x)$ is the subdifferential of f at x and Id denotes the identity operator, and the proximal mapping of $\mathbb{I}_{\mathcal{X}}$ relative to $\|\cdot\|_{\widetilde{M}}$ is uniquely determined by the resolvent of the operator $\widetilde{M}^{-1}\hat{\partial}\mathbb{I}_{\mathcal{X}}$ in (A.25).
First, we obtain that the operator $\widetilde{M}^{-1}\hat{\partial}\mathbb{I}_{\mathcal{X}}$ is monotone. For a convex extended-valued function $f_e : \mathbb{R}^N \to \mathbb{R}\cup\{\infty\}$, observe that $\widetilde{M}^{-1}\hat{\partial} f_e$ is a monotone operator:

$$\langle \widetilde{M}^{-1}\hat{\partial} f_e(u) - \widetilde{M}^{-1}\hat{\partial} f_e(v), u - v\rangle = \langle \underbrace{\widetilde{M}^{-1}\widetilde{M}}_{=I}\hat{\partial} f_e(\widetilde{M}\tilde{u}) - \underbrace{\widetilde{M}^{-1}\widetilde{M}}_{=I}\hat{\partial} f_e(\widetilde{M}\tilde{v}), \widetilde{M}\tilde{u} - \widetilde{M}\tilde{v}\rangle \geq 0, \qquad \forall u, v, \tag{A.26}$$

where the equality uses the variable change $\{u = \widetilde{M}\tilde{u}, v = \widetilde{M}\tilde{v}\}$, a chain rule of the subdifferential of a composition of a convex extended-valued function and an affine mapping [12, §7], and the symmetry of $\widetilde{M}$, and the inequality holds because the subdifferential of convex extended-valued function is a monotone operator [13, §4.2]. Because characteristic function of a convex set is extended-valued function, the result in (A.26) implies that the operator $\widetilde{M}^{-1}\hat{\partial}\mathbb{I}_{\mathcal{X}}$ is monotone. Second, note that the resolvent of a monotone operator $\widetilde{M}^{-1}\hat{\partial}\mathbb{I}_{\mathcal{X}}$ (with a parameter 1), i.e., $(\mathrm{Id}+\widetilde{M}^{-1}\hat{\partial}\mathbb{I}_{\mathcal{X}})^{-1}$ in (A.25), is nonexpansive [10, §6] and thus continuous. We now obtain that the convex proximal minimization (A.24) is continuous with respect to the input points $\acute{x}^{(i_j+1)}$ and $z^{(i_j+1)}$, because the proximal mapping operator $(\mathrm{Id}+\widetilde{M}^{-1}\hat{\partial}\mathbb{I}_{\mathcal{X}})^{-1}$ in (A.25), the affine mapping $\widetilde{M}^{-1}$, and $\nabla F(x; y, z)$ are continuous with respect to their input points.

For the two cases above, using the fact that $x^{(i_j+1)} \to \bar{x}$, $\acute{x}^{(i_j+1)} \to \bar{x}$, $z^{(i_j+1)} \to \bar{z}$, and $M^{(i_j+1)} \to \bar{M}$ (or $\bar{M} = M$ for the $\{M^{(i+1)} = M\}$ case) as $j \to \infty$, (A.23) becomes

$$\bar{x} = \underset{x\in\mathcal{X}}{\operatorname{argmin}} \langle \nabla F(\bar{x}; y, \bar{z}), x - \bar{x}\rangle + \frac{1}{2}\|x - \bar{x}\|^2_{\bar{M}}. \tag{A.27}$$

Thus, $\bar{x}$ satisfies the first-order optimality condition of $\min_{x\in\mathcal{X}} F(x; y, \bar{z})$:

$$\langle \nabla F(\bar{x}; y, \bar{z}), x - \bar{x}\rangle \geq 0, \qquad \text{for any } x \in \mathcal{X},$$

and this completes the proof of the first result.

Next, note that the result in Proposition 9 imply

$$\left\|\mathcal{A}^{M^{(i+1)}}_{\mathcal{R}_{\theta^{(i+1)}}}\left(\begin{bmatrix} x^{(i)} \\ x^{(i-1)}\end{bmatrix}\right) - \begin{bmatrix} x^{(i)} \\ x^{(i-1)}\end{bmatrix}\right\|_2 \to 0. \tag{A.28}$$

Additionally, note that a function $\mathcal{A}^{M^{(i+1)}}_{\mathcal{R}_{\theta^{(i+1)}}} - I$ is continuous. To see this, observe that the convex proximal mapping in (Alg.1.3) is continuous (see the obtained results above), and $\mathcal{R}_{\theta^{(i+1)}}$ is continuous (see Assumption 4). Combining (A.28), the convergence of $\{M^{(i_j+1)}, \mathcal{R}_{\theta^{(i_j+1)}}\}$, and the continuity of $\mathcal{A}^{M^{(i+1)}}_{\mathcal{R}_{\theta^{(i+1)}}} - I$, we obtain $[\bar{x}^T, \bar{x}^T]^T = \mathcal{A}^{\bar{M}}_{\mathcal{R}_{\bar{\theta}}}([\bar{x}^T, \bar{x}^T]^T)$, and this completes the proofs of the second result.

A.6 PROOFS OF COROLLARY 11

To prove the first result, we use proof by contradiction. Suppose that $\operatorname{dist}(x^{(i)}, \mathcal{S}) \nrightarrow 0$. Then there exists $\epsilon > 0$ and a subsequence $\{x^{(i_j)}\}$ such that $\operatorname{dist}(x^{(i_j)}, \mathcal{S}) \geq \epsilon$, $\forall j$. However, the boundedness assumption of $\{x^{(i_j)}\}$ in Corollary 11 implies that there must exist a limit point $\bar{x} \in \mathcal{S}$ via Theorem 10. This is a contradiction, and gives the first result (via the result in Proposition 9). Under the isolation point assumption in Corollary 11, using the obtained results, $\|x^{(i+1)} - x^{(i)}\|_2 \to 0$ (via Proposition 9) and $\operatorname{dist}(x^{(i+1)}, \mathcal{S}) \to 0$, and the following the proofs in [4, Cor. 2.4], we obtain the second result.

A.7 MOMENTUM-NET VS. BCD-NET

This section compares the convergence properties of Momentum-Net (Algorithm 1) and BCD-Net (Algorithm 2). We first show that for convex $f(x;y)$ and $\mathcal{X}$, the sequence of reconstructed images generated by BCD-Net converges:

Proposition A.5 (Sequence convergence)**.** *In Algorithm 2, let $f(x;y)$ be convex and subdifferentiable, and $\mathcal{X}$ be convex. Assume that the paired operators $(\mathcal{R}_{\theta^{(i+1)}}, \mathcal{R}_{\theta^{(i)}})$ are asymptotically contractive, i.e.,*

$$\|\mathcal{R}_{\theta^{(i+1)}}(u) - \mathcal{R}_{\theta^{(i)}}(v)\|_2 < \|u - v\|_2 + \epsilon^{(i+1)},$$

with $\sum_{i=0}^{\infty} \epsilon^{(i+1)} < \infty$ and $\{\epsilon^{(i+i)} \in [0,\infty) : \forall i\}$, $\forall u, v, i$. Then, the sequence $\{x^{(i+1)} : i \geq 0\}$ generated by Algorithm 2 is convergent.

Proof. We rewrite the updates in Algorithm 2 as follows:

$$\begin{aligned} x^{(i+1)} &= \underset{x\in\mathcal{X}}{\operatorname{argmin}}\ f(x;y) + \frac{\gamma}{2}\left\|x - \mathcal{R}_{\theta^{(i+1)}}(x^{(i)})\right\|_2^2 = \operatorname{Prox}^{\gamma I}_{f+\mathbb{I}_{\mathcal{X}}}\big(\mathcal{R}_{\theta^{(i+1)}}(x^{(i)})\big) \\ &= \big(\mathrm{Id} + \gamma^{-1}\hat{\partial}(f(x;y) + \mathbb{I}_{\mathcal{X}})\big)^{-1}\big(\mathcal{R}_{\theta^{(i+1)}}(x^{(i)})\big) \\ &=: \mathcal{A}^{(i+1)}(x^{(i)}). \end{aligned}$$

We first show that the paired operators $\{\mathcal{A}^{(i+1)}, \mathcal{A}^{(i)}\}$ is asymptotically contractive:

$$\begin{aligned} &\left\|\mathcal{A}^{(i+1)}(u) - \mathcal{A}^{(i)}(v)\right\|_2 \\ &= \left\|\big(\mathrm{Id} + \gamma^{-1}\hat{\partial}(f(x;y) + \mathbb{I}_{\mathcal{X}})\big)^{-1}\big(\mathcal{R}_{\theta^{(i+1)}}(u)\big) - \big(\mathrm{Id} + \gamma^{-1}\hat{\partial}(f(x;y) + \mathbb{I}_{\mathcal{X}})\big)^{-1}\big(\mathcal{R}_{\theta^{(i)}}(v)\big)\right\|_2 \\ &\leq \|\mathcal{R}_{\theta^{(i+1)}}(u) - \mathcal{R}_{\theta^{(i)}}(v)\|_2 \qquad \text{(A.29)} \\ &\leq L'\|u - v\|_2 + \epsilon^{(i+1)}\|u - v\|_2, \qquad \text{(A.30)} \end{aligned}$$

$\forall u, v$, where the inequality (A.29) holds because the subdifferential of the convex extended-valued function $f(x;y) + \mathbb{I}_{\mathcal{X}}$ (the characteristic function of a convex set $\mathcal{X}$, $\mathbb{I}_{\mathcal{X}}$, is convex, and the sum of the two convex functions, $f(x;y) + \mathbb{I}_{\mathcal{X}}$, is convex) is a monotone operator [13, §4.2], and the resolvent of a monotone relation with a positive parameter, i.e., $(\mathrm{Id} + \gamma^{-1}\hat{\partial}(f(x;y) + \mathbb{I}_{\mathcal{X}}))^{-1}$ with $\gamma^{-1} > 0$, is nonexpansive [13, §6], and the inequality (A.30) holds by $L' < 1$ via the contractiveness of the paired operators $(\mathcal{R}_{\theta^{(i+1)}}, \mathcal{R}_{\theta^{(i)}})$, $\forall i$. Note that the inequality (A.29) does not hold for nonconvex $f(x;y)$ and/or $\mathcal{X}$. Considering that $L' < 1$, we show that the sequence $\{x^{(i+1)} : i \geq 0\}$ is Cauchy sequence:

$$\begin{aligned} \left\|x^{(i+l)} - x^{(i)}\right\|_2 &= \left\|(x^{(i+l)} - x^{(i+l-1)}) + \ldots + (x^{(i+1)} - x^{(i)})\right\|_2 \\ &\leq \left\|x^{(i+l)} - x^{(i+l-1)}\right\| + \ldots + \left\|x^{(i+1)} - x^{(i)}\right\|_2 \\ &\leq \big({L'}^{l-1} + \ldots + 1\big)\left\|x^{(i+1)} - x^{(i)}\right\|_2 + \big(\epsilon^{(i+l)} + \ldots + \epsilon^{(i+1)}\big) \\ &\leq \frac{1}{1-L'}\left\|x^{(i+1)} - x^{(i)}\right\|_2 + \sum_{i'=1}^{l} \epsilon^{(i+i')} \end{aligned}$$

where the second inequality uses the result in (A.30). Since the sequence $\{x^{(i+1)} : i \geq 0\}$ is Cauchy sequence, $\{x^{(i+1)} : i \geq 0\}$ is convergent, and this completes the proofs. □

In terms of guaranteeing convergence, BCD-Net has three theoretical or practical limitations compared to Momentum-Net:

- Different from Momentum-Net, BCD-Net assumes the asymptotic contractive condition for the paired operators $\{\mathcal{R}_{\theta^{(i+1)}}, \mathcal{R}_{\theta^{(i)}}\}$. When image mapping operators in (Alg.2.1) are identical across iterations, i.e., $\{\mathcal{R}_\theta = \mathcal{R}_{\theta^{(i+1)}} : i \geq 0\}$, then $\mathcal{R}_\theta$ is assumed to be contractive. On the other hand, a mapping operator (identical across iterations) of Momentum-Net only needs to be nonexpansive. Note, however, that when $f(x;y) = \frac{1}{2}\|y - Ax\|_W^2$ with $A^H W A \succ 0$ (e.g., Example 5), BCD-Net can guarantee the sequence convergence with the asymptotically nonexpansive paired operators $(\mathcal{R}_{\theta^{(i+1)}}, \mathcal{R}_{\theta^{(i)}})$ (see Definition 7) [14].
- When one applies an iterative solver to (Alg.2.2), there always exist some numerical errors and these obstruct the sequence convergence guarantee in Proposition A.5. To guarantee sufficiently small numerical errors from iterative methods solving (Alg.2.2) (so that one can find a critical point solution for the MBIR problem (Alg.2.2)), one needs to use sufficiently many inner iterations that can substantially slow down entire MBIR.
- BCD-Net does not guarantee the sequence convergence for nonconvex data-fit $f(x;y)$, whereas Momentum-Net guarantees convergence to a fixed-point for both convex $f(x;y)$ and nonconvex $f(x;y)$.

A.8 For the sCNN architecture (18), connection between convolutional training loss (P2) and its patch-based training loss

This section shows that given the sCNN architecture (18), the convolutional training loss in (P2) has three advantages over the patch-based training loss in [14], [15] that may use all the extracted overlapping patches of size R:

- The corresponding patch-based loss does not model the patch aggregation process that is inherently modeled in (18).
- It is an upper bound of the convolutional loss (P2).
- It requires about R times more memory than (P2).

We prove the benefits of (P2) using the following lemma.

Lemma A.6. *The loss function (P2) for training the residual convolutional autoencoder in (18) is bounded by the patch-based loss function:*

$$\frac{1}{2L}\sum_{s=1}^{S}\Big\|\hat{x}_s^{(i)}-\frac{1}{R}\sum_{k=1}^{K}d_k\circledast\mathcal{T}_{\alpha_k}(e_k\circledast x_s^{(i)})\Big\|_2^2\leq\frac{1}{2LR}\sum_{s=1}^{S}\|\widehat{X}_s^{(i)}-D\mathcal{T}_{\tilde{\alpha}}(EX_s^{(i)})\|_{\mathrm{F}}^2, \tag{A.31}$$

where the residual is defined by $\hat{x}_s^{(i)}\triangleq x_s-x_s^{(i)}$, $\{x_s,x_s^{(i)}\}$ are given as in (P2), $\widehat{X}_s\in\mathbb{R}^{R\times V_s}$ and $X_s\in\mathbb{R}^{R\times V_s}$ are the lth training data matrices whose columns are V_s vectorized patches extracted from the images $\hat{x}_s$ and x_s (with the circulant boundary condition and the "stride" parameter 1), respectively, $D\triangleq[d_1,\ldots,d_K]\in\mathbb{C}^{R\times K}$ is a decoding filter matrix, and $E\triangleq[e_1^*,\ldots,e_K^*]^H\in\mathbb{C}^{K\times R}$ is an encoding filter matrix. Here, the definition of soft-thresholding operator in (6) is generalized by

$$(\mathcal{T}_{\tilde{\alpha}}(u))_k\triangleq\begin{cases}u_k-\alpha_k\cdot\mathrm{sign}(u_k), & |u_k|>\alpha_k,\\ 0, & \textit{otherwise},\end{cases} \tag{A.32}$$

for $K=1,\ldots,K$, where $\tilde{\alpha}=[\alpha_1,\ldots,\alpha_K]^T$. See other related notations in (18).

Proof. First, we have the following reformulation [3, §S.1]:

$$\begin{bmatrix}e_1*u\\ \vdots\\ e_K*u\end{bmatrix}=\underbrace{P'\begin{bmatrix}EP_1\\ \vdots\\ EP_N\end{bmatrix}}_{\triangleq\widetilde{E}}u,\quad\forall u, \tag{A.33}$$

where $P'\in\mathbb{C}^{KN\times KN}$ is a permutation matrix, E is defined in Lemma (A.6), and $P_n\in\mathbb{C}^{R\times N}$ is the nth patch extraction operator for $n=1,\ldots,N$. Considering that $\frac{1}{R}\sum_{k=1}^{K}\mathsf{flip}(e_k^*)\circledast(e_k\circledast u)=\widetilde{E}^H\widetilde{E}u$ via the definition of $\widetilde{E}$ in (A.33) (see also the reformulation technique in [3, §S.1]), we obtain the following reformulation result:

$$\frac{1}{R}\sum_{k=1}^{K}\mathsf{flip}(e_k^*)\circledast\mathcal{T}_{\alpha_k}(e_k\circledast x_s^{(i)})=\frac{1}{R}\sum_{n=1}^{N}P_n^HE^H\mathcal{T}_{\tilde{\alpha}}\big(EP_nx_s^{(i)}\big) \tag{A.34}$$

where the soft-thresholding operators $\{\mathcal{T}_{\alpha_k}(\cdot):\forall k\}$ and $\mathcal{T}_{\tilde{\alpha}}(\cdot)$ are defined in (A.32) and we use the permutation invariance of the thresholding operator $\mathcal{T}_\alpha(\cdot)$, i.e., $\mathcal{T}_\alpha(P(\cdot))=P\cdot\mathcal{T}_\alpha(\cdot)$ for any α. Finally, we obtain the result in (A.31) as follows:

$$\begin{aligned}\frac{1}{2L}\sum_{s=1}^{S}\Big\|\hat{x}_s^{(i)}-\frac{1}{R}\sum_{k=1}^{K}d_k\circledast\mathcal{T}_{\alpha_k}(e_k\circledast x_s^{(i)})\Big\|_2^2&=\frac{1}{2L}\sum_{s=1}^{S}\Big\|\hat{x}_s^{(i)}-\frac{1}{R}\sum_{n=1}^{N}P_n^HD\mathcal{T}_{\tilde{\alpha}}\big(EP_nx_s^{(i)}\big)\Big\|_2^2 &\text{(A.35)}\\ &=\frac{1}{2L}\sum_{s=1}^{S}\Big\|\frac{1}{R}\sum_{n=1}^{N}P_n^HP_n\hat{x}_s^{(i)}-\frac{1}{R}\sum_{n=1}^{N}P_n^HD\mathcal{T}_{\tilde{\alpha}}\big(EP_nx_s^{(i)}\big)\Big\|_2^2 &\text{(A.36)}\\ &=\frac{1}{2LR^2}\sum_{s=1}^{S}\Big\|\sum_{n=1}^{N}P_n^H\Big(\hat{x}_{l,n}^{(i)}-D\mathcal{T}_{\tilde{\alpha}}\big(Ex_{l,n}^{(i)}\big)\Big)\Big\|_2^2\\ &\leq\frac{1}{2LR}\sum_{s=1}^{S}\sum_{n=1}^{N}\Big\|\hat{x}_{l,n}^{(i)}-D\mathcal{T}_{\tilde{\alpha}}\big(Ex_{l,n}^{(i)}\big)\Big\|_2^2 &\text{(A.37)}\\ &=\frac{1}{2LR}\sum_{s=1}^{S}\Big\|\widehat{X}_s^{(i)}-D\mathcal{T}_{\tilde{\alpha}}\Big(EX_s^{(i)}\Big)\Big\|_{\mathrm{F}}^2,\end{aligned}$$

where D is defined in Lemma A.6, $\{\hat{x}_{l,n}^{(i)}=P_n\hat{x}_s^{(i)}\in\mathbb{C}^R,x_{l,n}^{(i)}=P_nx_s^{(i)}\in\mathbb{C}^R:n=1,\ldots,N\}$ is a set of extracted patches, the training matrices $\{\widehat{X}_s^{(i)},X_s^{(i)}\}$ are defined by $\widehat{X}_s^{(i)}\triangleq[\hat{x}_{l,n}^{(i)},\ldots,\hat{x}_{l,N}^{(i)}]$ and $X_s^{(i)}\triangleq[x_{l,1}^{(i)},\ldots,x_{l,N}^{(i)}]$. Here, the equality (A.35) uses the result in (A.34), the equality (A.36) holds by $\sum_{n=1}^{N}P_n^HP_n=R\cdot I$ (for the circulant boundary condition in Lemma A.6), and the inequality (A.37) holds by $\widetilde{P}\widetilde{P}^H\preceq R\cdot I$ with $\widetilde{P}\triangleq[P_1^H\cdots P_N^H]^H$. □

Lemma A.6 reveals that when the patch-based training approach extract all the R-size overlapping patches, *1)* the corresponding patch-based loss is an upper bound of the convolutional loss (P2); *2)* it requires about R-times larger memory than (P2) because $V_s \approx RN_s$ for $x \in \mathbb{R}^{N_s}$ and the boundary condition described in Lemma A.6, $\forall l$; and *3)* it misses modeling the patch aggregation process that is inherently modeled in (18) – see that the patch aggregation operator $\sum_{n=1}^{N} P_n^H(\cdot)_n$ is removed in the inequality (A.37) in the proof of Lemma A.6. In addition, different from the patch-based training approach [14], [15], i.e., training with the function on the right-hand side in (A.31), one can use different sizes of filters $\{e_k, d_k : \forall k\}$ in the convolutional training loss, i.e., the function on the left-hand side in (A.31).

A.9 Details of experimental setup

A.9.1 Majorization matrix designs for quadratic data-fit

For (real-valued) quadratic data-fit $f(x;y)$ in the form of $\frac{1}{2}\|y - Ax\|_W^2$, if a majorization matrix M exists such that $A^H W A \preceq M$, it is straightforward to verify that the gradient of quadratic data-fit $f(x;y)$ satisfies the M-Lipchitz continuity in Definition 1, i.e.,

$$\|\nabla f(u;y) - \nabla f(v;y)\|_{M^{-1}} = \|A^H W A u - A^H W A v\|_{M^{-1}} \leq \|u - v\|_M^2, \quad \forall u, v \in \mathbb{R}^N.$$

because the assumption $A^T W A \preceq M \Leftrightarrow M^{-1/2} A^T W A M^{-1/2} \preceq I$ implies that the eigenspectrum of $M^{-1/2} A^T W A M^{-1/2}$ lies in the interval $[0, 1]$, and gives the following result:

$$\left(M^{-1/2} A^T W A M^{-1/2}\right)^2 \preceq I \Leftrightarrow (A^T W A) M^{-1} (A^T W A) \preceq M.$$

Next, we review a useful lemma in designing majorization matrices for a wide class of quadratic data-fit $f(x;y)$:

Lemma A.7 ([2, Lem. S.3])**.** *For a (possibly complex-valued) matrix A and a diagonal matrix W with non-negative entries, $A^H W A \preceq \mathrm{diag}(|A^H|W|A|1)$, where $|A|$ denotes the matrix consisting of the absolute values of the elements of A.*

A.9.2 Parameters for MBIR optimization models: Sparse-view CT reconstruction

For MBIR model using EP regularization, we combined a EP regularizer $\sum_{n=1}^{N} \sum_{n' \in \mathcal{N}_n} \iota_n \iota_{n'} \varphi(x_n - x_{n'})$ and the data-fit $f(x;y)$ in §4.1.1, where $\mathcal{N}_n$ is the set of indices of the neighborhood, ι_n and $\iota_{n'}$ are parameters that encourage uniform noise [16], and $\varphi(\cdot)$ is the Lange penalty function, i.e., $\varphi(t) = \delta^2(|t/\delta| - \log(1 + |t/\delta|))$, with $\delta = 10$ in HU. We chose the regularization parameter (e.g., γ in (P0)) as $2^{15.5}$. We ran the relaxed linearized augmented Lagrangian method [17] with 100 iterations and 12 ordered-subsets, and initialized the EP MBIR algorithms with a conventional FBP method using a Hanning window.

For MBIR model using a learned convolutional regularizer [6, (P2)], we trained convolutional regularizer with filters of $\{h_k \in \mathbb{R}^R : R = K = 7^2\}$ via CAOL [3] in an unsupervised training manner; see training details in [3]. The regularization parameters (e.g., γ in (1)) were selected by applying the "spectral spread" based selection scheme in §3.2 with the tuned factor $\chi^\star = 167.64$. We selected the spatial-strength-controlling hard-thresholding parameter (i.e., α' in [6, (P2)]) as follows: for Test samples #1–2, we chose it is as 10^{-10} and 6^{-11}, respectively. We initialized the MBIR model using a learned regularizer with the EP MBIR results obtained above. We terminated the iterations if the relative error stopping criterion (e.g., [2, (44)]) is met before reaching the maximum number of iterations. We set the tolerance value as 10^{-13} and the maximum number of iterations to 4×10^3.

A.9.3 Parameters for MBIR optimization models: LF photography using a focal stack

For MBIR model using 4D EP regularization [18], we combined a 4D EP regularizer $\sum_{n=1}^{N} \sum_{n' \in \mathcal{N}_n} \varphi(x_n - x_{n'})$ and the data-fit $f(x;y)$ in §4.1.2, where $\mathcal{N}_n$ is the set of indices of the 4D neighborhood, and $\varphi(\cdot)$ is the hyperbola penalty function, i.e., $\varphi(t) = \delta^2(\sqrt{1 + |t/\delta|^2} - 1)$. We selected the hyperbola function parameter δ and regularization parameter (e.g., γ in (P0)) as follows: for Test samples #1–3, we chose them as $\{\delta = 10^{-4}, \gamma = 10^3\}$, $\{\delta = 10^{-1}, \gamma = 10^7\}$, and $\{\delta = 10^{-1}, \gamma = 5 \times 10^3\}$, respectively. We ran the conjugate gradient method with 100 iterations, and initialized the 4D EP MBIR algorithms with $A^T y$ rescaled in the interval $[0, 1]$.

A.9.4 Reconstruction accuracy and depth estimation accuracy of different MBIR methods

Tables A.1–A.3 below provide reconstruction accuracy numerics of different MBIR methods in sparse-view CT reconstruction and LF photography using a focal stack, and reports the SPO depth estimation [19] accuracy numerics on reconstructed LFs from different MBIR methods.

A.9.5 Reconstructed images and estimated depths with noniterative analytical methods

This section provides reconstructed images by an analytical back-projection method in sparse-view CT reconstruction and LF photography using a focal stack (see the first two columns in Fig. A.2), and estimated depths from reconstructed LFs via the SPO depth estimation method [19] (see the third column in Fig. A.2(c)). Results in Fig. A.2 below are supplementary to Fig. 8, Fig. 9, and Fig. 10, and the first two columns visualize initial input images to INN methods.

TABLE A.1
RMSE (HU) of different CT MBIR methods
(fan-beam geometry with 12.5% projections views and 10^5 incident photons)

	(a) FBP	(b) EP reg.	(c) Learned convolutional reg. [3], [6]	(d) Momentum-Net-sCNN	(e) **Momentum-Net-sCNN** w/ larger width	(f) Momentum-Net-dCNN
Test #1	82.8	40.8	35.2	19.9	**19.5**	19.8
Test #2	74.9	38.5	34.5	18.4	**17.7**	17.8

(c)'s convolutional regularizer uses $\{R=K=7^2\}$
(d)'s refining sCNNs are in the form of residual single-hidden layer convolutional autoencoder (18) with $\{R=K=7^2\}$.
(e)'s refining sCNNs are in the form of residual single-hidden layer convolutional autoencoder (18) with $\{R=7^2, K=9^2\}$.
This setup gives results in Fig. 8(d), as described in §4.2.1.
(f)'s refining dCNNs are in the form of residual multi-hidden layer CNN (19) with $\{L=4, R=3^2, K=64\}$.

TABLE A.2
PSNR (dB) of different LF MBIR methods
(LF photography systems with $C=5$ detectors obtain a focal stack of LFs consisting of $S=81$ sub-aperture images)

	(a) A^Ty	(b) 4D EP reg. [18]	(c) Momentum-Net-sCNN	(d) **Momentum-Net-dCNN**
Test #1	16.4	32.0	35.8	**37.1**
Test #2	21.1	28.1	30.7	**32.0**
Test #3	21.6	28.1	30.9	**31.7**

(c)'s refining sCNNs are in the form of residual single-hidden layer convolutional autoencoder with $\{R=5^2, K=32\}$.
(d)'s refining dCNNs are in the form of residual multi-hidden layer CNN (19) with $\{L=6, R=3^2, K=16\}$.
Momentum-Nets use refining CNNs in an epipolar-domain; see details in §4.2.1.

TABLE A.3
RMSE (in 10^{-2}, m) of estimated depth from reconstructed LFs with different LF MBIR methods
(LF photography systems with $C=5$ detectors obtain a focal stack of LFs consisting of $S=81$ sub-aperture images)

	(a) Ground truth LF	(b) Reconstructed LF by A^Ty	(c) Reconstructed LF by 4D EP reg. [18]	(d) Reconstructed LF by Momentum-Net-sCNN	(e) Reconstructed LF by **Momentum-Net-dCNN**
Test #1	4.7	41.0	13.8	8.0	**5.7**
Test #2	30.5	117.6	39.5	34.6	**31.9**
Test #3	n/a†	n/a†	n/a†	n/a†	n/a†

SPO depth estimation [19] was applied to reconstructed LFs.
(d)'s refining sCNNs are in the form of residual single-hidden layer convolutional autoencoder with $\{R=5^2, K=32\}$.
(e)'s refining dCNNs are in the form of residual multi-hidden layer CNN (19) with $\{L=6, R=3^2, K=16\}$.
Momentum-Nets use refining CNNs in an epipolar-domain; see details in §4.2.1.
†The ground truth depth map for Test sample #3 does not exist in the LF dataset [20].

A.10 How to choose parameters of image refining modules in soft-refining INNs?

In soft-refining INNs using iterative-wise refining NNs, one does not need to greatly increase parameter dimensions of refining NNs [14], [21]. The natural question then arises, "How one can choose between sCNN (18) and dCNN (19) refiners, and select their parameters (R, K, and L)?" The first answer to this question depends on some understanding of data-fit $f(x;y)$ in MBIR problem (P1), e.g., the regularization strength γ and the condition number variations across training data-fit majorizers. (An additional criteria could be general understandings between sample size/diversity and parameter dimension of NNs.)

For example, the sparse-view CT system in §4.1.1 needs moderate regularization strength ($\chi^\star = 167.64$) and the majorization matrices of its training data-fits have mild condition number variations (the standard deviation is 1.1). training data-fits have mild parameter variations across samples. Comparing results between Momentum-Net-sCNN and -dCNN in Fig. 5 and Table A.1 demonstrates that sCNN (18) seems suffice. Table A.1(d)–(e) shows that one can further improve the refining accuracy of sCNN (18) by increasing its width, i.e., K. The LF photography system using a limited focal stack in §4.1.2 needs a large γ value ($\chi^\star = 1.5$), and the majorization matrices of its training data-fits have large condition number variations (the standard deviation is 2245.5). Comparing results between Momentum-Net-sCNN and -dCNN in Fig. 7 and Table A.2 demonstrates that dCNN (19) yields higher PSNR than sCNN (18). For dCNN (19), we observed increasing its depth, i.e., L, up to a certain number is more effective than increasing its width, i.e., K, as briefly discussed in §4.2.1.

For choosing the relaxation parameter ρ in (Alg.1.1), we also suggest considering the regularization strength in (Alg.1.3). For an application that needs moderate regularization strength, e.g., sparse-view CT in §4.1.1, we suggest setting ρ to 0.5 so as to mix information between input and output of refining NNs, rather than $1-\varepsilon$ that does not mix input and output. For an application that needs strong regularization, e.g., LF photography using a limited focal stack in §4.1.2, we suggest using $\rho = 1-\varepsilon$ than $\rho = 0.5$. Results in the next section validate this suggestion.

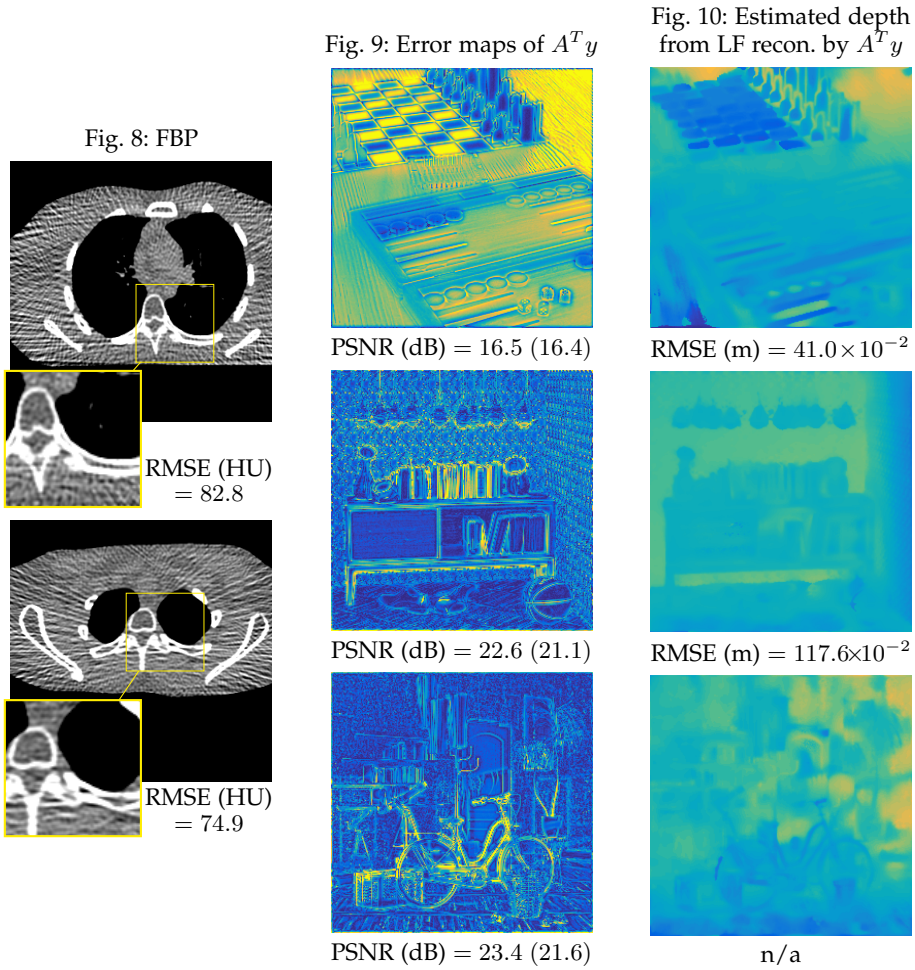


Fig. A.2. Reconstructed images from analytical back-projection methods. We used such results in the first two columns to initialize INN methods.

A.10.1 Performance of Momentum-Net with different relaxation parameters ρ in (Alg.1.1)

Fig. A.3 below compares the performances of Momentum-Net-sCNN with different ρ values. The results in Fig. A.3 support the ρ selection guideline in §4.2.3. One can maximize the MBIR accuracy of Momentum-Net by properly selecting ρ.

Note that $\rho \in (0, 1)$ controls strength of inference from refining NNs in (Alg.1.1), but does not affect the convergence guarantee of Momentum-Net. Fig. A.3 illustrates that Momentum-Net appears to converge regardless of ρ values.

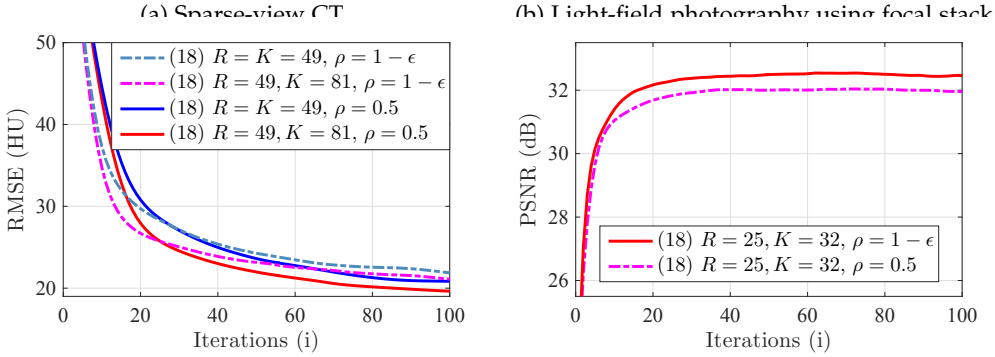


Fig. A.3. Convergence behavior of Momentum-Net-sCNN with different relaxation parameters, $\rho = 0.5$ and $\rho = 1 - \varepsilon$. For both applications (see their imaging setups in §4.1), PyTorch ver. 0.3.1 was used.

A.11 Parameters of Momentum-Net

Table A.4 below lists parameters of Momentum-Net, and summarizes selection guidelines or default values. Similar to BCD-Net/ADMM-Net, the main tuning jobs to maximize the performance of Momentum-Net include selecting architectures of refining NNs $\{\mathcal{R}_{\theta^{(i)}} : \forall i\}$ in (Alg.1.1), and choosing a regularization parameter γ in (Alg.1.3) by tuning χ in §3.2. One can simplify the tuning process by using the selection guidelines in §A.10 for selecting architectures of $\{\mathcal{R}_{\theta^{(i)}} : \forall i\}$, and training

χ in §3.2. Note that one designs majorization matrices $\{M^{(i)} : \forall i\}$ rather than tuning them: majorization matrices can be analytically designed, e.g., Lemma A.7 as used in §4.2.1; one can algorithmically design them [22]. Tighter majorization matrices are expected to further accelerate the convergence of Momentum-Net [2], [3].

TABLE A.4
Guidelines for choosing parameters of Momentum-Net

Param.	Module	Guidelines or default values
$\{\mathcal{R}_{\theta^{(i)}} : \forall i\}$	(Alg.1.1)	Trainable by §3.1. For selecting their architecture/param., see guideline §A.10.
$\rho \in (0, 1)$	(Alg.1.1)	Use regularization strength γ; see guideline in §A.10.
$\delta < 1$ in (8)–(9)	(Alg.1.2)	$1-\varepsilon$
$\{M^{(i)} : \forall i\}$	(Alg.1.3)	Designed off-line. For large-scale inverse problems with quadratic data-fit, use Lemma A.7.
$\lambda \geq 1$ in (7)	(Alg.1.3)	For convex $F(x; y, z^{(i+1)})$, $\lambda = 1$; for nonconvex $F(x; y, z^{(i+1)})$, $\lambda = 1+\varepsilon$.
$\gamma > 0$	(Alg.1.3)	Chosen by tuning/training χ in §3.2

All INN methods also must select a number of INN iterations, N_{iter}. One could determine it by using the convergence behavior of iteration-wise refiners in Fig. 2.

REFERENCES

[1] I. Y. Chun, Z. Huang, H. Lim, and J. A. Fessler, "Momentum-Net: Fast and convergent iterative neural network for inverse problems," submitted, Jul. 2019.
[2] 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.
[3] ——, "Convolutional analysis operator learning: Acceleration and convergence," *IEEE Trans. Image Process.*, vol. 29, pp. 2108–2122, 2020.
[4] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imaging Sci.*, vol. 6, no. 3, pp. 1758–1789, Sep. 2013.
[5] ——, "A globally convergent algorithm for nonconvex optimization based on block coordinate update," *J. Sci. Comput.*, vol. 72, no. 2, pp. 700–734, Aug. 2017.
[6] I. Y. Chun and J. A. Fessler, "Convolutional analysis operator learning: Application to sparse-view CT," in *Proc. Asilomar Conf. on Signals, Syst., and Comput.*, Pacific Grove, CA, Oct. 2018, pp. 1631–1635.
[7] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.*, vol. 140, no. 1, pp. 125–161, Aug. 2013.
[8] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.
[9] S. Foucart and H. Rauhut, *A mathematical introduction to compressive sensing*. New York, NY: Springer, 2013.
[10] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Berlin: Springer Verlag, 2009, vol. 317.
[11] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM J. Control Optm.*, vol. 14, no. 5, pp. 877–898, Aug. 1976.
[12] B. S. Mordukhovich and N. M. Nam, "Geometric approach to convex subdifferential calculus," *Optimization*, vol. 66, no. 6, pp. 839–873, 2017.
[13] E. K. Ryu and S. Boyd, "Primer on monotone operator methods," *Appl. Comput. Math*, vol. 15, no. 1, pp. 3–43, Jan. 2016.
[14] I. Y. Chun, X. Zheng, Y. Long, and J. A. Fessler, "BCD-Net for low- dose CT reconstruction: Acceleration, convergence, and generalization," in *Proc. Med. Image Computing and Computer Assist. Interven.*, Shenzhen, China, Oct. 2019, pp. 31–40.
[15] I. Y. Chun and J. A. Fessler, "Deep BCD-net using identical encoding-decoding CNN structures for iterative image recovery," in *Proc. IEEE IVMSP Workshop*, Zagori, Greece, Jun. 2018, pp. 1–5.
[16] J. H. Cho and J. A. Fessler, "Regularization designs for uniform spatial resolution and noise properties in statistical image reconstruction for 3-D X-ray CT," *IEEE Trans. Med. Imag.*, vol. 34, no. 2, pp. 678–689, Feb. 2015.
[17] H. Nien and J. A. Fessler, "Relaxed linearized algorithms for faster X-ray CT image reconstruction," *IEEE Trans. Med. Imag.*, vol. 35, no. 4, pp. 1090–1098, Apr. 2016.
[18] M.-B. Lien, C.-H. Liu, I. Y. Chun, S. Ravishankar, H. Nien, M. Zhou, J. A. Fessler, Z. Zhong, and T. B. Norris, "Ranging and light field imaging with transparent photodetectors," *Nature Photonics*, vol. 14, no. 3, pp. 143–148, Jan. 2020.
[19] S. Zhang, H. Sheng, C. Li, J. Zhang, and Z. Xiong, "Robust depth estimation for light field via spinning parallelogram operator," *Comput. Vis. Image Und.*, vol. 145, pp. 148–159, Apr. 2016.
[20] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4D light fields," in *Proc. ACCV*, Taipei, Taiwan, Nov. 2016, pp. 19–34.
[21] H. Lim, I. Y. Chun, Y. K. Dewaraja, and J. A. Fessler, "Improved low-count quantitative PET reconstruction with an iterative neural network," *IEEE Trans. Med. Imag.* (to appear), May 2020. [Online]. Available: http://arxiv.org/abs/1906.02327
[22] M. G. McGaffin and J. A. Fessler, "Algorithmic design of majorizers for large-scale inverse problems," *ArXiv preprint stat.CO/1508.02958*, Oct. 2015.