Block Coordinate Regularization by Denoising

Yu Sun
Washington University in St. Louis
sun.yu@wustl.edu

Jiaming Liu
Washington University in St. Louis
jiaming.liu@wustl.edu

Ulugbek S. Kamilov
Washington University in St. Louis
kamilov@wustl.edu

Abstract

We consider the problem of estimating a vector from its noisy measurements using a prior specified only through a denoising function. Recent work on plug-and-play priors (PnP) and regularization-by-denoising (RED) has shown the state-of-the-art performance of estimators under such priors in a range of imaging tasks. In this work, we develop a new block coordinate RED algorithm that decomposes a large-scale estimation problem into a sequence of updates over a small subset of the unknown variables. We theoretically analyze the convergence of the algorithm and discuss its relationship to the traditional proximal optimization. Our analysis complements and extends recent theoretical results for RED-based estimation methods. We numerically validate our method using several denoiser priors, including those based on convolutional neural network (CNN) denoisers.

1 Introduction

Problems involving estimation of an unknown vector $\boldsymbol{x} \in \mathbb{R}^n$ from a set of noisy measurements $\boldsymbol{y} \in \mathbb{R}^m$ are important in many areas, including machine learning, image processing, and compressive sensing. Consider the scenario in Fig. 1, where a vector $\boldsymbol{x} \sim p_{\boldsymbol{x}}$ passes through the measurement channel $p_{\boldsymbol{y}|\boldsymbol{x}}$ to produce the measurement vector $\boldsymbol{y}$. When the estimation problem is ill-posed, it becomes essential to include the prior $p_{\boldsymbol{x}}$ in the estimation process. However, in high-dimensional settings, it is often difficult to directly obtain the true prior $p_{\boldsymbol{x}}$ and one is hence restricted to various indirect sources of prior information on $\boldsymbol{x}$. This paper considers the cases where the prior information on $\boldsymbol{x}$ is specified only via a denoising function, $\mathsf{D} : \mathbb{R}^n \to \mathbb{R}^n$, designed for the removal of additive white Gaussian noise (AWGN).

There has been considerable recent interest in leveraging denoisers as priors for the recovery of $\boldsymbol{x}$. One popular strategy, known as plug-and-play priors (PnP) [1], extends traditional proximal optimization [2] by replacing the proximal operator with a general off-the-shelf denoiser. It has been shown that the combination of proximal algorithms with advanced denoisers, such as BM3D [3] or DnCNN [4], leads to the state-of-the-art performance for various imaging problems [5–13]. A similar strategy has also been adopted in the context of a related class of algorithms known as approximate message passing (AMP) [14–17]. Regularization-by-denoising (RED) [18], and the closely related deep mean-shift priors [19], represent an alternative, in which the denoiser is used to specify an explicit regularizer that has a simple gradient. More recent work has clarified the existence of explicit RED regularizers [20], demonstrated its excellent performance on phase retrieval [21], and further boosted its performance in combination with a deep image prior [22]. In short, the use of advanced denoisers has proven to be essential for achieving the state-of-the-art results in many contexts. However, solving the corresponding estimation problem is still a significant computational challenge, especially in the context of high-dimensional vectors $\boldsymbol{x}$, typical in modern applications.

Preprint. Under review.

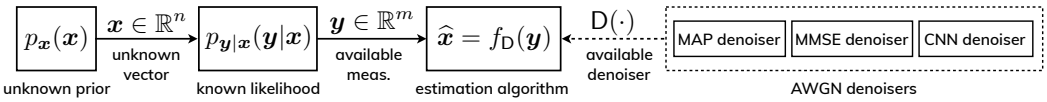

Figure 1: *The estimation problem considered in this work. The vector $\boldsymbol{x} \in \mathbb{R}^n$, with a prior $p_{\boldsymbol{x}}(\boldsymbol{x})$, passes through the measurement channel $p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x})$ to result in the measurements $\boldsymbol{y} \in \mathbb{R}^m$. The estimation algorithm $f_{\mathsf{D}}(\boldsymbol{y})$ does not have a direct access to the prior, but can rely on a denoising function $\mathsf{D} : \mathbb{R}^n \to \mathbb{R}^n$, specifically designed for the removal of additive white Gaussian noise (AWGN). We propose block coordinate RED as a scalable algorithm for obtaining $\boldsymbol{x}$ given $\boldsymbol{y}$ and D.*

In this work, we extend the current family of RED algorithms by introducing a new *block coordinate RED (BC-RED)* algorithm. The algorithm relies on random partial updates on $\boldsymbol{x}$, which makes it scalable to vectors that would otherwise be prohibitively large for direct processing. Additionally, as we shall see, the overall computational complexity of BC-RED can sometimes be lower than corresponding methods operating on the full vector. This behavior is consistent with the traditional coordinate descent methods that can outperform their full gradient counterparts by being able to better reuse local updates and take larger steps [23–26]. We present two theoretical results related to BC-RED. We first theoretically characterize the convergence of the algorithm under a set of transparent assumptions on the data-fidelity and the denoiser. Our analysis complements the recent theoretical analysis of full-gradient RED algorithms in [20] by considering block-coordinate updates and establishing the explicit worst-case convergence rate. Our second result establishes backward compatibility of BC-RED with the traditional proximal optimization. We show that when the denoiser corresponds to a proximal operator, BC-RED can be interpreted as an approximate MAP estimator, whose approximation error can be made arbitrarily small. To the best of our knowledge, this explicit link with proximal optimization is missing in the current literature on RED. BC-RED thus provides a flexible, scalable, and theoretically sound algorithm applicable to a wide variety of large-scale estimation problems. We demonstrate BC-RED on image recovery from linear measurements using several denoising priors, including those based on convolutional neural network (CNN) denoisers.

All proofs and some technical details have been omitted for space and included into the supplement that also provides more background and additional simulations.

2 Background

It is common to formulate the estimation in Figure 1 as an optimization problem

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathsf{arg\,min}}\, f(\boldsymbol{x}) \quad \text{with} \quad f(\boldsymbol{x}) = g(\boldsymbol{x}) + h(\boldsymbol{x}), \tag{1}$$

where g is the data-fidelity term and h is the regularizer. For example, the maximum a posteriori probability (MAP) estimator is obtained by setting

$$g(\boldsymbol{x}) = -\mathsf{log}(p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x})) \quad \text{and} \quad h(\boldsymbol{x}) = -\mathsf{log}(p_{\boldsymbol{x}}(\boldsymbol{x})),$$

where $p_{\boldsymbol{y}|\boldsymbol{x}}$ is the likelihood that depends on $\boldsymbol{y}$ and $p_{\boldsymbol{x}}$ is the prior. One of the most popular data-fidelity terms is least-squares $g(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$, which assumes a linear measurement model under AWGN. Similarly, one of the most popular regularizers is based on a sparsity-promoting penalty $h(\boldsymbol{x}) = \tau\|\boldsymbol{D}\boldsymbol{x}\|_1$, where $\boldsymbol{D}$ is a linear transform and $\tau > 0$ is the regularization parameter [27–30].

Many widely used regularizers, including the ones based on the ℓ_1-norm, are nondifferentiable. Proximal algorithms [2], such as the proximal-gradient method (PGM) [31–34] and alternating direction method of multipliers (ADMM) [35–38], are a class of optimization methods that can circumvent the need to differentiate nonsmooth regularizers by using the proximal operator

$$\mathsf{prox}_{\mu h}(\boldsymbol{z}) \;\coloneqq\; \underset{\boldsymbol{x}\in\mathbb{R}^n}{\mathsf{arg\,min}} \left\{ \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \mu h(\boldsymbol{x}) \right\}, \quad \mu > 0, \quad \boldsymbol{z} \in \mathbb{R}^n. \tag{2}$$

The observation that the proximal operator can be interpreted as the MAP denoiser for AWGN has prompted the development of PnP [1], where the proximal operator $\mathsf{prox}_{\mu h}(\cdot)$, within ADMM or PGM, is replaced with a more general denoising function $\mathsf{D}(\cdot)$.

Consider the following alternative to PnP that also relies on a denoising function [18, 19]

$$\boldsymbol{x}^t \leftarrow \boldsymbol{x}^{t-1} - \gamma \left(\nabla g(\boldsymbol{x}^{t-1}) + \mathsf{H}(\boldsymbol{x}^{t-1}) \right) \quad \text{where} \quad \mathsf{H}(\boldsymbol{x}) \coloneqq \tau(\boldsymbol{x} - \mathsf{D}(\boldsymbol{x})), \quad \tau > 0. \tag{3}$$

Under some conditions on the denoiser, it is possible to relate $\mathsf{H}(\cdot)$ in (3) to some explicit regularization function h. For example, when the denoiser is locally homogeneous and has a symmetric Jacobian [18, 20], the operator $\mathsf{H}(\cdot)$ corresponds to the gradient of the following function

$$h(\boldsymbol{x}) = \frac{\tau}{2} \boldsymbol{x}^{\mathsf{T}} (\boldsymbol{x} - \mathsf{D}(\boldsymbol{x})). \tag{4}$$

On the other hand, when the denoiser corresponds to the minimum mean squared error (MMSE) estimator $\mathsf{D}(\boldsymbol{z}) = \mathbb{E}[\boldsymbol{x}|\boldsymbol{z}]$ for the AWGN denoising problem [19, 20], $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{e}$, with $\boldsymbol{x} \sim p_{\boldsymbol{x}}(\boldsymbol{x})$ and $\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$, the operator $\mathsf{H}(\cdot)$ corresponds to the gradient of

$$h(\boldsymbol{x}) = -\tau\sigma^2 \mathsf{log}(p_{\boldsymbol{z}}(\boldsymbol{x})), \quad p_{\boldsymbol{z}}(\boldsymbol{x}) = (p_{\boldsymbol{x}} * p_{\boldsymbol{e}})(\boldsymbol{x}) = \int_{\mathbb{R}^n} p_{\boldsymbol{x}}(\boldsymbol{z}) \phi_\sigma(\boldsymbol{x} - \boldsymbol{z}) \, \mathsf{d}\boldsymbol{z}, \tag{5}$$

where ϕ_σ is the Gaussian probability density function of variance σ^2 and $*$ denotes convolution. In this paper, we will use the term RED to denote all methods seeking the fixed points of (3). The key benefits of the RED methods [18–22] are their explicit separation of the forward model from the prior, their ability to accommodate powerful denoisers (such as the ones based on CNNs) without differentiating them, and their state-of-the-art performance on a number of imaging tasks. The next section further extends the scalability of RED by designing a new block coordinate RED algorithm.

3 Block Coordinate RED

All the current RED algorithms operate on vectors in $\mathbb{R}^n$. We propose BC-RED, shown in Algorithm 1, to allow for partial randomized updates on $\boldsymbol{x}$. Consider the decomposition of $\mathbb{R}^n$ into $b \geq 1$ subspaces

$$\mathbb{R}^n = \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \cdots \times \mathbb{R}^{n_b} \quad \text{with} \quad n = n_1 + n_2 + \cdots + n_b.$$

For each $i \in \{1, \ldots, b\}$, we define the matrix $\mathsf{U}_i : \mathbb{R}^{n_i} \to \mathbb{R}^n$ that injects a vector in $\mathbb{R}^{n_i}$ into $\mathbb{R}^n$ and its transpose $\mathsf{U}_i^{\mathsf{T}}$ that extracts the ith block from a vector in $\mathbb{R}^n$. Then, for any $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_b) \in \mathbb{R}^n$

$$\boldsymbol{x} = \sum_{i=1}^{b} \mathsf{U}_i \boldsymbol{x}_i \quad \text{with} \quad \boldsymbol{x}_i = \mathsf{U}_i^{\mathsf{T}} \boldsymbol{x} \in \mathbb{R}^{n_i}, \quad i = 1, \ldots, b \quad \Leftrightarrow \quad \sum_{i=1}^{b} \mathsf{U}_i \mathsf{U}_i^{\mathsf{T}} = \mathsf{I}. \tag{6}$$

Note that (6) directly implies the norm preservation $\|\boldsymbol{x}\|_2^2 = \|\boldsymbol{x}_1\|_2^2 + \cdots + \|\boldsymbol{x}_b\|_2^2$ for any $\boldsymbol{x} \in \mathbb{R}^n$. We are interested in a block-coordinate algorithm that uses only a subset of operator outputs corresponding to coordinates in some block $i \in \{1, \ldots, b\}$. Hence, for an operator $\mathsf{G} : \mathbb{R}^n \to \mathbb{R}^n$, we define the block-coordinate operator $\mathsf{G}_i : \mathbb{R}^n \to \mathbb{R}^{n_i}$ as

$$\mathsf{G}_i(\boldsymbol{x}) \coloneqq [\mathsf{G}(\boldsymbol{x})]_i = \mathsf{U}_i^{\mathsf{T}} \mathsf{G}(\boldsymbol{x}) \in \mathbb{R}^{n_i}, \quad \boldsymbol{x} \in \mathbb{R}^n. \tag{7}$$

We introduce the following BC-RED algorithm.

Algorithm 1 Block Coordinate Regularization by Denoising (BC-RED)

1: **input:** initial value $\boldsymbol{x}^0 \in \mathbb{R}^n$, parameter $\tau > 0$, and step-size $\gamma > 0$.
2: **for** $k = 1, 2, 3, \ldots$ **do**
3: Choose an index $i_k \in \{1, \ldots, b\}$
4: $\boldsymbol{x}^k \leftarrow \boldsymbol{x}^{k-1} - \gamma \mathsf{U}_{i_k} \mathsf{G}_{i_k}(\boldsymbol{x}^{k-1})$
where $\mathsf{G}_i(\boldsymbol{x}) \coloneqq \mathsf{U}_i^{\mathsf{T}} \mathsf{G}(\boldsymbol{x})$ with $\mathsf{G}(\boldsymbol{x}) \coloneqq \nabla g(\boldsymbol{x}) + \tau(\boldsymbol{x} - \mathsf{D}(\boldsymbol{x}))$.
5: **end for**

Note that when $b = 1$, we have $n = n_1$ and $\mathsf{U}_1 = \mathsf{U}_1^{\mathsf{T}} = \mathsf{I}$. Hence, the theoretical analysis in this paper is also applicable to the full-gradient RED algorithm in (3).

As with traditional coordinate descent methods (see [25] for a review), BC-RED can be implemented using different block selection strategies. The strategy adopted for our theoretical analysis selects block indices i_k as i.i.d. random variables distributed uniformly over $\{1, \ldots, b\}$. An alternative is to

proceed in epochs of b consecutive iterations, where at the start of each epoch the set $\{1, \dots, b\}$ is reshuffled, and i_k is then selected consecutively from this ordered set. We numerically compare the convergence of both BC-RED variants in Section 5.

BC-RED updates its iterates one randomly picked block at a time using the output of G. When the algorithm converges, it converges to the vectors in the zero set of G

$$\mathsf{G}(\boldsymbol{x}^*) = \nabla g(\boldsymbol{x}^*) + \tau(\boldsymbol{x}^* - \mathsf{D}(\boldsymbol{x}^*)) = \boldsymbol{0} \quad \Leftrightarrow \quad \boldsymbol{x}^* \in \mathsf{zer}(\mathsf{G}) \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n : \mathsf{G}(\boldsymbol{x}) = \boldsymbol{0}\}.$$

Consider the following two sets

$$\mathsf{zer}(\nabla g) \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n : \nabla g(\boldsymbol{x}) = \boldsymbol{0}\} \quad \text{and} \quad \mathsf{fix}(\mathsf{D}) \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} = \mathsf{D}(\boldsymbol{x})\},$$

where $\mathsf{zer}(\nabla g)$ is the set of all critical points of the data-fidelity and $\mathsf{fix}(\mathsf{D})$ is the set of all fixed points of the denoiser. Intuitively, the fixed points of D correspond to all the vectors that are not denoised, and therefore can be interpreted as vectors that are *noise-free* according to the denoiser.

Note that if $\boldsymbol{x}^* \in \mathsf{zer}(\nabla g) \cap \mathsf{fix}(\mathsf{D})$, then $\mathsf{G}(\boldsymbol{x}^*) = \boldsymbol{0}$ and $\boldsymbol{x}^*$ is one of the solutions of BC-RED. Hence, any vector that is consistent with the data for a convex g and noiseless according to D is in the solution set. On the other hand, when $\mathsf{zer}(\nabla g) \cap \mathsf{fix}(\mathsf{D}) = \varnothing$, then $\boldsymbol{x}^* \in \mathsf{zer}(\mathsf{G})$ corresponds to a tradeoff between the two sets, explicitly controlled via $\tau > 0$ (see Fig. 8 in the supplement for an illustration). This explicit control is one of the key differences between RED and PnP.

BC-RED benefits from considerable *flexibility* compared to the full-gradient RED. Since each update is restricted to only one block of $\boldsymbol{x}$, the algorithm is suitable for parallel implementations and can deal with problems where the vector $\boldsymbol{x}$ is distributed in space and in time. However, the maximal benefit of BC-RED is achieved when G_i is efficient to evaluate. Fortunately, it was systematically shown in [39] that many operators—common in machine learning, image processing, and compressive sensing—admit *coordinate friendly* updates.

For a specific example, consider the least-squares data-fidelity g and a block-wise denoiser D. Define the residual vector $r(\boldsymbol{x}) \coloneqq \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}$ and consider a single iteration of BC-RED that produces $\boldsymbol{x}^+$ by updating the ith block of $\boldsymbol{x}$. Then, the update direction and the residual update can be computed as

$$\mathsf{G}_i(\boldsymbol{x}) = \boldsymbol{A}_i^{\mathsf{T}} r(\boldsymbol{x}) + \tau(\boldsymbol{x}_i - \mathsf{D}(\boldsymbol{x}_i)) \quad \text{and} \quad r(\boldsymbol{x}^+) = r(\boldsymbol{x}) - \gamma \boldsymbol{A}_i \mathsf{G}_i(\boldsymbol{x}), \tag{8}$$

where $\boldsymbol{A}_i \in \mathbb{R}^{m \times n_i}$ is a submatrix of $\boldsymbol{A}$ consisting of the columns corresponding to the ith block. In many problems of practical interest [39], the complexity of working with $\boldsymbol{A}_i$ is roughly b times lower than with $\boldsymbol{A}$. Also, many advanced denoisers can be effectively applied on image blocks rather than on the full image [40–42]. Therefore, the speed of b iterations of BC-RED is expected to be at least comparable to a single iteration of the full-gradient RED (see also Section E.1 in the supplement).

4 Convergence Analysis and Compatibility with Proximal Optimization

In this section, we present two theoretical results related to BC-RED. We first establish its convergence to an element of $\mathsf{zer}(\mathsf{G})$ and then discuss its compatibility with the theory of proximal optimization.

4.1 Fixed Point Convergence of BC-RED

Our analysis requires three assumptions that together serve as sufficient conditions for convergence.

Assumption 1. *The operator G is such that $\mathsf{zer}(\mathsf{G}) \neq \varnothing$. There is a finite number R_0 such that the distance of the initial $\boldsymbol{x}^0 \in \mathbb{R}^n$ to the farthest element of $\mathsf{zer}(\mathsf{G})$ is bounded, that is*

$$\max_{\boldsymbol{x}^* \in \mathsf{zer}(\mathsf{G})} \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|_2 \leq R_0.$$

This assumption is necessary to guarantee convergence and is related to the existence of the minimizers in the literature on traditional coordinate minimization [23–25].

The next two assumptions rely on Lipschitz constants along directions specified by specific blocks. We say that G_i is *block Lipschitz continuous* with constant $\lambda_i > 0$ if

$$\|\mathsf{G}_i(\boldsymbol{x}) - \mathsf{G}_i(\boldsymbol{y})\|_2 \leq \lambda_i \|\boldsymbol{h}_i\|_2, \quad \boldsymbol{x} = \boldsymbol{y} + \mathsf{U}_i \boldsymbol{h}_i, \quad \boldsymbol{y} \in \mathbb{R}^n, \boldsymbol{h}_i \in \mathbb{R}^{n_i}.$$

When $\lambda_i = 1$, we say that G_i is *block nonexpansive*. Note that if an operator G is globally λ-Lipschitz continuous, then it is straightforward to see that each $\mathsf{G}_i = \mathsf{U}_i^{\mathsf{T}} \mathsf{G}$ is also block λ-Lipschitz continuous.

Assumption 2. *The function g is continuously differentiable and convex. Additionally, for each $i \in \{1, \dots, b\}$ the block gradient $\nabla_i g$ is block Lipschitz continuous with constant $L_i > 0$. We define the largest block Lipschitz constant as $L_{\mathsf{max}} \coloneqq \mathsf{max}\{L_1, \dots, L_b\}$.*

Let $L > 0$ denote the global Lipschitz constant of ∇g. We always have $L_{\mathsf{max}} \leq L$ and, for some g, it may even happen that $L_{\mathsf{max}} = L/b$ [25]. As we shall see, the largest possible step-size γ of BC-RED depends on L_{max}, while that of the full-gradient RED on L. Hence, one natural advantage of BC-RED is that it can often take more aggressive steps compared to the full-gradient RED.

Assumption 3. *The denoiser D is such that each block denoiser D_i is block nonexpansive.*

Since the proximal operator is nonexpansive [2], it automatically satisfies this assumption. We revisit this scenario in a greater depth in Section 4.2. We can now establish the following result for BC-RED.

Theorem 1. *Run BC-RED for $t \geq 1$ iterations with random i.i.d. block selection under Assumptions 1-3 using a fixed step-size $0 < \gamma \leq 1/(L_{\mathsf{max}} + 2\tau)$. Then, we have*

$$\mathbb{E}\left[\min_{k \in \{1,\dots,t\}} \|\mathsf{G}(\boldsymbol{x}^{k-1})\|_2^2\right] \leq \mathbb{E}\left[\frac{1}{t}\sum_{k=1}^{t} \|\mathsf{G}(\boldsymbol{x}^{k-1})\|_2^2\right] \leq \frac{b(L_{\mathsf{max}} + 2\tau)}{\gamma t} R_0^2. \tag{9}$$

A proof of the theorem is provided in the supplement. Theorem 1 establishes that the iterates of BC-RED in expectation can get arbitrarily close to $\mathsf{zer}(\mathsf{G})$ with $O(1/t)$ rate. The proof relies on the monotone operator theory [43, 44], widely used in the context of convex optimization [2], including in the unified analysis of various traditional coordinate descent algorithms [45, 46].

Since $L_{\mathsf{max}} \leq L$, one important implication of Theorem 1, is that the worst-case convergence rate (in expectation) of b iterations of BC-RED is better than that of a single iteration of the full-gradient RED (to see this, note that the full-gradient rate is obtained by setting $b = 1$, $L_{\mathsf{max}} = L$, and removing the expectation in (9)). This implies that in *coordinate friendly settings* (as discussed at the end of Section 3), the overall computational complexity of BC-RED can be lower than that of the full-gradient RED. This gain is primarily due to two factors: (a) possibility to pick a larger step-size $\gamma = 1/(L_{\mathsf{max}} + 2\tau)$; (b) immediate reuse of each local block-update when computing the next iterate (the full-gradient RED updates the full vector before computing the next iterate).

In the special case of $\mathsf{D}(\boldsymbol{x}) = \boldsymbol{x} - (1/\tau)\nabla h(\boldsymbol{x})$, for some convex function h, BC-RED reduces to the traditional randomized coordinate descent method applied to (1). Hence, under the assumptions of Theorem 1, one can rely on the analysis of traditional coordinate descent methods in [25] to obtain

$$\mathbb{E}\left[f(\boldsymbol{x}^t)\right] - f^* \leq \frac{2b}{\gamma t} R_0^2 \tag{10}$$

where f^* is the minimum value in (1). A proof of (10) is provided in the supplement for completeness. Therefore, such denoisers lead to explicit convex RED regularizers and $O(1/t)$ convergence of BC-RED in terms of the objective. However, as discussed in Section 4.2, when the denoiser is a proximal operator of some convex h, BC-RED is *not* directly solving (1), but rather its approximation.

Finally, note that the analysis in Theorem 1 only provides *sufficient conditions* for the convergence of BC-RED. As corroborated by our numerical studies in Section 5, the actual convergence of BC-RED is more general and often holds beyond nonexpansive denoisers. One plausible explanation for this is that such denoisers are *locally nonexpansive* over the set of input vectors used in testing. On the other hand, the recent techniques for spectral-normalization of CNNs [47–49] provide a convenient tool for building *globally nonexpansive* neural denoisers that result in provable convergence of BC-RED.

4.2 Convergence for Proximal Operators

One of the limitations of the current RED theory is in its limited backward compatibility with the theory of proximal optimization. For example, as discussed in [18] (see section *"Can we mimic any prior?"*), the popular total variation (TV) denoiser [27] cannot be justified with the original RED regularization function (4). In this section, we show that BC-RED (and hence also the full-gradient RED) can be used to solve (1) for any convex, closed, and proper function h. We do this by establishing a formal link between RED and the concept of Moreau smoothing, widely used in nonsmooth optimization [50–52]. In particular, we consider to the following generic denoiser

$$\mathsf{D}(\boldsymbol{z}) = \mathsf{prox}_{(1/\tau)h}(\boldsymbol{z}) = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\mathsf{arg\,min}} \left\{\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + (1/\tau)h(\boldsymbol{x})\right\}, \quad \tau > 0, \quad \boldsymbol{z} \in \mathbb{R}^n, \tag{11}$$

where h is a closed, proper, and convex function [2]. Since the proximal operator is nonexpansive, it is also block nonexpansive, which means that Assumption 3 is automatically satisfied. Our analysis, however, requires an additional assumption using the constant R_0 defined in Assumption 1.

Assumption 4. *There is a finite number G_0 that bounds the largest subgradient of h, that is*

$$\max\{\|\boldsymbol{g}(\boldsymbol{x})\|_2 : \boldsymbol{g}(\boldsymbol{x}) \in \partial h(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{B}(\boldsymbol{x}^0, R_0)\} \leq G_0,$$

where $\mathcal{B}(\boldsymbol{x}^0, R_0) := \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x} - \boldsymbol{x}^0\|_2 \leq R_0\}$ denotes a ball of radius R_0, centered at $\boldsymbol{x}^0$.

This assumption on boundedness of the subgradients holds for a large number of regularizers used in practice, including both TV and the ℓ_1-norm penalties. We can now establish the following result.

Theorem 2. *Run BC-RED for $t \geq 1$ iterations with random i.i.d. block selection and the denoiser* (11) *under Assumptions 1-4 using a fixed step-size $0 < \gamma \leq 1/(L_{\max} + 2\tau)$. Then, we have*

$$\mathbb{E}\left[f(\boldsymbol{x}^t)\right] - f^* \leq \frac{2b}{\gamma t} R_0^2 + \frac{G_0^2}{2\tau}, \tag{12}$$

where the function f is defined in (1) *and f^* is its minimum.*

The theorem is proved in the supplement. It establishes that BC-RED in expectation *approximates* the solution of (1) with an error bounded by $(G_0^2/(2\tau))$. Hence, by setting $\tau = \sqrt{t}$ and $\gamma = 1/(L_{\max} + 2\sqrt{t})$, one can establish the following worst-case convergence rate

$$\mathbb{E}\left[f(\boldsymbol{x}^t)\right] - f^* \leq \frac{1}{\sqrt{t}}\left[2b(L_{\max} + 2)R_0^2 + G_0^2\right]. \tag{13}$$

When $h(\boldsymbol{x}) = -\log(p_{\boldsymbol{x}}(\boldsymbol{x}))$, the proximal operator corresponds to the MAP denoiser, and the solution of BC-RED corresponds to an approximate MAP estimator. This approximation can be made as precise as desired by considering larger values for the parameter $\tau > 0$. Note that this further justifies the RED framework by establishing that it can be used to compute a minimizer of any proper, closed, and convex (but not necessarily differentiable) h. Therefore, our analysis strengthens RED by showing that it can accommodate a much larger class of explicit regularization functions, beyond those characterized in (4) and (5).

5 Numerical Validation

There is a considerable recent interest in using advanced priors in the context of image recovery from underdetermined ($m < n$) and noisy measurements. Recent work [18–22] suggests significant performance improvements due to advanced denoisers (such as BM3D [3] or DnCNN [4]) over traditional sparsity-driven priors (such as TV [27]). Our goal is to complement these studies with several simulations validating our theoretical analysis and providing additional insights into BC-RED.

We consider inverse problems of form $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$, where $\boldsymbol{e} \in \mathbb{R}^m$ is an AWGN vector and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is a matrix corresponding to either a sparse-view Radon transform, i.i.d. zero-mean Gaussian random matrix of variance $1/m$, or radially subsampled two-dimensional Fourier transform. Such matrices are commonly used in the context of computerized tomography (CT) [53], compressive sensing [29, 30], and magnetic resonance imaging (MRI) [54], respectively. In all simulations, we set the measurement ratio to be approximately $m/n = 0.5$ with AWGN corresponding to input signal-to-noise ratio (SNR) of 30 dB and 40 dB. The images used correspond to 10 images randomly selected from the NYU fastMRI dataset [55], resized to be 160×160 pixels (see Fig. 5 in the supplement). BC-RED is set to work with 16 blocks, each of size 40×40 pixels. The reconstruction quality is quantified using SNR averaged over all ten test images.

In addition to well-studied denoisers, such as TV and BM3D, we design our own CNN denoiser denoted DnCNN*, which is a simplified version of the popular DnCNN denoiser (see Supplement E for details). This simplification reduces the computational complexity of denoising, which is important when running many iterations of BC-RED. Additionally, it makes it easier to control the global Lipschitz constant of the CNN via spectral-normalization [48]. We train DnCNN* for the removal of AWGN at four noise levels corresponding to $\sigma \in \{5, 10, 15, 20\}$. For each experiment, we select the denoiser achieving the highest SNR value. Note that the σ parameter of BM3D is also fine-tuned for each experiment from the same set $\{5, 10, 15, 20\}$.

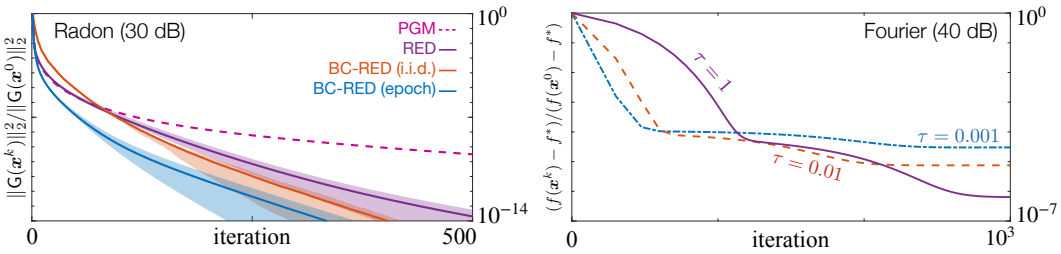

Figure 2: ***Left**: Illustration of the convergence of BC-RED under a nonexpansive DnCNN* prior. Average normalized distance to* $\mathsf{zer}(\mathsf{G})$ *is plotted against the iteration number with the shaded areas representing the range of values attained over all test images.* ***Right**: Illustration of the influence of the parameter* $\tau > 0$ *for solving TV regularized least-squares problem using BC-RED. As* τ *increases, BC-RED provides an increasingly accurate approximation to the TV optimization problem.*

Table 1: Average SNRs obtained for different measurement matrices and image priors.

Methods	**Radon**		**Random**		**Fourier**	
	30 dB	**40 dB**	**30 dB**	**40 dB**	**30 dB**	**40 dB**
PGM (TV)	20.66	24.40	26.07	***28.42***	28.74	29.99
U-Net	***21.90***	21.72	16.37	16.40	22.11	22.11
RED (TV)	20.79	24.46	25.64	28.30	28.67	29.97
BC-RED (TV)	20.78	24.42	25.70	28.40	28.71	29.99
RED (BM3D)	21.55	***25.24***	26.46	27.82	28.89	29.79
BC-RED (BM3D)	21.56	25.16	26.52	27.89	28.85	29.80
RED (DnCNN*)	20.89	24.38	26.53	28.05	29.33	*30.32*
BC-RED (DnCNN*)	20.88	24.42	***26.60***	28.12	***29.40***	***30.39***

Theorem 1 establishes that the sequence of iterates generated by BC-RED converges in expectation to an element of $\mathsf{zer}(\mathsf{G})$. This is illustrated in Fig. 2 (left) for the Radon matrix with 30 dB noise and a nonexpansive DnCNN* denoiser (see also Fig. 6 in the supplement). The average value of $\|\mathsf{G}(\boldsymbol{x}^k)\|_2^2/\|\mathsf{G}(\boldsymbol{x}^0)\|_2^2$ is plotted against the iteration number for the full-gradient RED and BC-RED, with b updates of BC-RED (each modifying a single block) represented as one iteration. We numerically tested two block selection rules for BC-RED (*i.i.d.* and *epoch*) and observed that processing in randomized epochs leads to a faster convergence. For reference, the figure also plots the normalized squared norm of the gradient mapping vectors produced by the traditional PGM with TV [56]. The shaded areas indicate the range of values taken over 10 runs corresponding to each test image. The results highlight the potential of BC-RED to enjoy a better convergence rate compared to the full-gradient RED, with BC-RED (epoch) achieving the accuracy of 10^{-10} in 104 iterations, while the full-gradient RED achieves the same accuracy in 190 iterations.

Theorem 2 establishes that for proximal-operator denoisers, BC-RED computes an approximate solution to (1) with an accuracy controlled by the parameter τ. This is illustrated in Fig. 2 (right) for the Fourier matrix with 40 dB noise and the TV regularized least-squares problem. The average value of $(f(\boldsymbol{x}^k) - f^*)/(f(\boldsymbol{x}^0) - f^*)$ is plotted against the iteration number for BC-RED with $\tau \in \{0.01, 0.1, 1\}$. The optimal value f^* is obtained by running the traditional PGM until convergence. As before, the figure groups b updates of BC-RED as a single iteration. The results are consistent with our theoretical analysis and show that as τ increases BC-RED provides an increasingly accurate solution to TV. On the other hand, since the range of possible values for the step-size γ depends on τ, the speed of convergence to f^* is also influenced by τ.

The benefits of the full-gradient RED algorithms have been well discussed in prior work [18–22]. Table 1 summarizes the average SNR performance of BC-RED in comparison to the full-gradient RED for all three matrix types and several priors. Unlike the full-gradient RED, BC-RED is implemented using block-wise denoisers that work on image patches rather than the full images. We empirically

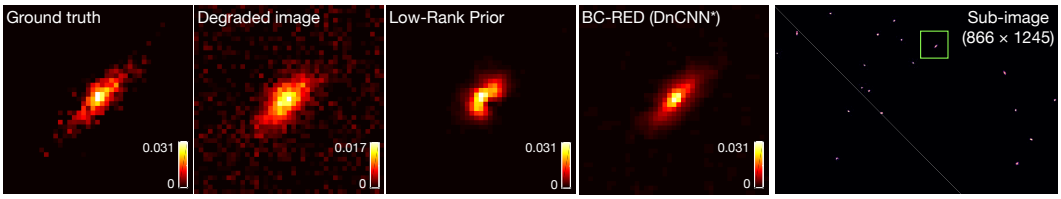

Figure 3: *Recovery of a* 8292×8364 *pixel galaxy image degraded by a spatially variant blur and a high-amount of AWGN. The efficacy of BC-RED on this problem is due to the natural sparsity in the latter, with all of the information contained in a small part of the full image.*

found that 40 pixel padding on the denoiser input is sufficient for BC-RED to match the performance of the full-gradient RED. The table also includes the results for the traditional PGM with TV [56] and the widely-used end-to-end U-Net approach [57, 58]. The latter first backprojects the measurements into the image domain and then denoises the result using U-Net [59]. The model was specifically trained end-to-end for the Radon matrix with 30 dB noise and applied as such to other measurement settings. All the algorithms were run until convergence with hyperparameters optimized for SNR. The DnCNN* denoiser in the table corresponds to the residual network with the Lipschitz constant of two (see Supplement E.2 for details). The overall best SNR in the table is highlighted in bold-italic, while the best RED prior is highlighted in light-green. First, note the excellent agreement between BC-RED and the full-gradient RED. This close agreement between two methods is encouraging as BC-RED relies on block-wise denoising and our analysis does not establish uniqueness of the solution, yet, in practice, both methods seem to yield solutions of nearly identical quality. Second, note that BC-RED and RED provide excellent approximations to PGM-TV solutions. Third, note how (unlike U-Net) BC-RED and RED with DnCNN* generalize to different measurement models. Finally. no prior seems to be universally good on all measurement settings, which indicates to the potential benefit of tailoring specific priors to specific measurement models.

Coordinate descent methods are known to be highly beneficial in problems where both m and n are very large, but each measurement depends only on a small subset of the unknowns [60]. Fig. 3 demonstrates BC-RED in such large-scale setting by adopting the experimental setup from a recent work [61] (see also Fig. 10 in the supplement). Specifically, we consider the recovery of a 8292×8364 pixel galaxy image degraded by 597 known point spread functions (PSFs) corresponding to different spatial locations (see Supplement F for details). The natural sparsity of the problem makes it ideal for BC-RED, which is implemented to update 41×41 pixel blocks in a randomized fashion by only picking areas containing galaxies. The computational complexity of BC-RED is further reduced by considering a simpler variant of DnCNN* that has only four convolutional layers (see Fig. 4 in the supplement). For comparison, we additionally show the result obtained by using the low-rank recovery method from [61] with all the parameters kept at the values set by the authors. Note that our intent here is not to justify DnCNN* as a prior for image deblurring, but to demonstrate that BC-RED can indeed be applied to a realistic, nontrivial image recovery task on a large image.

6 Conclusion and Future Work

Coordinate descent methods have become increasingly important in optimization for solving large-scale problems arising in data analysis. We have introduced BC-RED as a coordinate descent extension to the current family of RED algorithms and theoretically analyzed its convergence. Preliminary experiments suggest that BC-RED can be an effective tool in large-scale estimation problems arising in image recovery. More experiments are certainly needed to better asses the promise of this approach in various estimation tasks. For future work, we would like to explore accelerated and asynchronous variants of BC-RED to further enhance its performance in parallel settings.

Acknowledgments

This material is based upon work supported by NSF award CCF-1813910 and by NVIDIA Corporation with the donation of the Titan Xp GPU for research.

References

[1] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Global Conf. Signal Process. and INf. Process. (GlobalSIP)*, 2013.

[2] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2014.

[3] 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. 16, pp. 2080–2095, August 2007.

[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, July 2017.

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

[6] Sreehari *et al.*, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Trans. Comp. Imag.*, vol. 2, no. 4, pp. 408–423, December 2016.

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

[8] 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, December 2017.

[9] T. Meinhardt, M. Moeller, C. Hazirbas, and D. Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in *Proc. IEEE Int. Conf. Comp. Vis. (ICCV)*, 2017.

[10] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] 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, 2018.

[12] Y. Sun, B. Wohlberg, and U. S. Kamilov, "An online plug-and-play algorithm for regularized image reconstruction," *IEEE Trans. Comput. Imaging*, 2019.

[13] A. M. Teodoro, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "A convergent image fusion algorithm using scene-adapted Gaussian-mixture-based denoising," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 451–463, Jan. 2019.

[14] J. Tan, Y. Ma, and D. Baron, "Compressive imaging via approximate message passing with image denoising," *IEEE Trans. Signal Process.*, vol. 63, no. 8, pp. 2085–2092, Apr. 2015.

[15] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, September 2016.

[16] C. A. Metzler, A. Maleki, and R. Baraniuk, "BM3D-PRGAMP: Compressive phase retrieval based on BM3D denoising," in *Proc. IEEE Int. Conf. Image Proc.*, 2016.

[17] A. Fletcher, S. Rangan, S. Sarkar, and P. Schniter, "Plug-in estimation in high-dimensional linear inverse problems: A rigorous analysis," in *Proc. Advances in Neural Information Processing Systems 32*, 2018.

[18] 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, 2017.

[19] S. A. Bigdeli, M. Jin, P. Favaro, and M. Zwicker, "Deep mean-shift priors for image restoration," in *Proc. Advances in Neural Information Processing Systems 31*, Long Beach, CA, USA, Dec. 2017.

[20] 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.

[21] C. A. Metzler, P. Schniter, A. Veeraraghavan, and R. G. Baraniuk, "prDeep: Robust phase retrieval with a flexible deep network," in *Proc. 35th Int. Conf. Machine Learning (ICML)*, 2018.

[22] G. Mataev and P. Elad, M. Milanfar, "DeepRED: Deep image prior powered by RED," 2019, arXiv:1903.10176 [cs.CV].

[23] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM J. Optim.*, vol. 22, no. 2, pp. 341–362, 2012.

[24] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2037–2060, Oct. 2013.

[25] S. J. Wright, "Coordinate descent algorithms," *Math. Program.*, vol. 151, no. 1, pp. 3–34, June 2015.

[26] O. Fercoq and A. Gramfort, "Coordinate descent methods," Lecture notes *Optimization for Data Science*, École polytechnique, 2018.

[27] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1–4, pp. 259–268, November 1992.

[28] R. Tibshirani, "Regression and selection via the lasso," *J. R. Stat. Soc. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[29] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, February 2006.

[30] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[31] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, August 2003.

[32] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, November 2004.

[33] J. Bect, L. Blanc-Feraud, G. Aubert, and A. Chambolle, "A ℓ_1-unified variational framework for image restoration," in *Proc. ECCV*, Springer, Ed., New York, 2004, vol. 3024, pp. 1–13.

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

[35] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[36] M. V. Afonso, J. M.Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, September 2010.

[37] M. K. Ng, P. Weiss, and X. Yuan, "Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods," *SIAM J. Sci. Comput.*, vol. 32, no. 5, pp. 2710–2736, August 2010.

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

[39] Z. Peng, T. Wu, Y. Xu, M. Yan, and W. Yin, "Coordinate-friendly structures, algorithms and applications," *Adv. Math. Sci. Appl.*, vol. 1, no. 1, pp. 57–119, Apr. 2016.

[40] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, December 2006.

[41] A. Buades, B. Coll, and J. M. Morel, "Image denoising methods. A new nonlocal principle," *SIAM Rev*, vol. 52, no. 1, pp. 113–147, 2010.

[42] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. IEEE Int. Conf. Comp. Vis. (ICCV)*, 2011.

[43] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, 2 edition, 2017.

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

[45] Z. Peng, Y. Xu, M. Yan, and W. Yin, "ARock: An algorithmic framework for asynchronous parallel coordinate updates," *SIAM J. Sci. Comput.*, vol. 38, no. 5, pp. A2851–A2879, 2016.

[46] Y. T. Chow, T. Wu, and W. Yin, "Cyclic coordinate-update algorithms for fixed-point problems: Analysis and applications," *SIAM J. Sci. Comput.*, vol. 39, no. 4, pp. A1280–A1300, 2017.

[47] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[48] H. Sedghi, V. Gupta, and P. M. Long, "The singular values of convolutional layers," in *International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, May 2019.

[49] H. Gouk, E. Frank, B. Pfahringer, and M. Cree, "Regularisation of neural networks by enforcing Lipschitz continuity," 2018, arXiv:1804.04368.

[50] J. J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bull. Soc. Math. France*, vol. 93, pp. 273–299, 1965.

[51] R. T. Rockafellar and R. J-B Wets, *Variational Analysis*, Springer, 1998.

[52] Y.-L. Yu, "Better approximation and faster algorithm using the proximal average," in *Proc. Advances in Neural Information Processing Systems 26*, Lake Tahoe, CA, USA, December 5-10, 2013, pp. 458–466.

[53] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, IEEE, 1988.

[54] F. Knoll, K. Brendies, T. Pock, and R. Stollberger, "Second order total generalized variation (TGV) for MRI," *Magn. Reson. Med.*, vol. 65, no. 2, pp. 480–491, February 2011.

[55] Zbontar *et al.*, "fastMRI: An open dataset and benchmarks for accelerated MRI," 2018, arXiv:1811.08839.

[56] A. Beck and M. Teboulle, "Fast gradient-based algorithm for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, November 2009.

[57] 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, Sept. 2017.

[58] Y. S. Han, J. Yoo, and J. C. Ye, "Deep learning with domain adaptation for accelerated projection reconstruction MR," *Magn. Reson. Med.*, vol. 80, no. 3, pp. 1189–1205, Sept. 2017.

[59] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[60] F. Niu, B. Recht, C. Ré, and S. J. Wright, "Hogwild!: A lock-free approach to parallelizing stochastic gradient descent," in *Proc. Advances in Neural Information Processing Systems 24*, Granada, Spain, December 12-15, 2011, pp. 693–701.

[61] S. Farrens, F. M. Ngolè Mboula, and J.-L. Starck, "Space variant deconvolution of galaxy survey images," *A&A*, vol. 601, pp. A66, 2017.

[62] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge Univ. Press, 2004.

[63] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, 2004.

[64] Mandelbaum *et al.*, "The third gravitational lensing accuracy testing (GREAT3) challenge handbook," *Astrophys. J. Suppl. S.*, vol. 212, no. 1, pp. 5, Aug. 2014.

[65] Cropper *et al.*, "VIS: The visible imager for Euclid," in *Proc. SPIE*, 2012, vol. 8442.

[66] T. Kuntzer, M. Tewes, and F. Courbin, "Stellar classification from single-band imaging using machine learning," *A&A*, vol. 591, pp. A54, 2016.

Supplementary Material

We adopt the monotone operator theory [43, 44] for a unified analysis of BC-RED. In Supplement A, we prove the convergence of BC-RED to an element of $\mathsf{zer}(\mathsf{G})$. In Supplement B, we prove that for proximal-operator denoisers, BC-RED converges to an approximate solution of (1). For completeness, in Supplement C, we discuss the well-known convergence results for traditional coordinate descent [23–26]. In Supplement D, we provide the background material used in Supplement A and Supplement B, expressed in a form convenient for block-coordinate analysis. In Supplement E, we provide additional technical details omitted from the main paper due to space, such as the details on computational complexity and CNN architectures. In Supplement F, we present additional simulations that were also omitted from the main paper due to space.

A Proof of Theorem 1

A fixed-point convergence of averaged operators is well-known under the name of Krasnosel'skii-Mann theorem (see Section 5.2 in [43]) and was recently applied to the analysis of PnP [12] and several full-gradient RED algorithms in [20]. Our analysis here extends these results to the block-coordinate setting and provides explicit worst-case convergence rates for BC-RED.

We consider the following operators

$$\mathsf{G}_i = \nabla_i g + \mathsf{H}_i \quad \text{with} \quad \mathsf{H}_i = \tau \mathsf{U}_i^{\mathsf{T}}(\mathsf{I} - \mathsf{D}).$$

and proceed in several steps.

(a) Since $\nabla_i g$ is block L_i-Lipschitz continuous, it is also block L_{max}-Lipschitz continuous. Hence, we know from Proposition 7 in Supplement D.3 that it is block $(1/L_{\mathsf{max}})$-cocoercive. Then from Proposition 4 in Supplement D.2, we know that the operator $(\mathsf{U}_i^{\mathsf{T}} - (2/L_{\mathsf{max}})\nabla_i g)$ is block nonexpansive.

(b) From the definition of H_i and the fact that D_i is block nonexpansive, we know that $(\mathsf{U}_i^{\mathsf{T}} - (1/\tau)\mathsf{H}_i) = \mathsf{D}_i$ is block nonexpansive.

(c) From Proposition 1 in Supplement D.1, we know that a convex combination of block nonexpansive operators is also block nonexpansive, hence we conclude that

$$\begin{aligned}&\mathsf{U}_i^{\mathsf{T}} - \frac{2}{L_{\mathsf{max}} + 2\tau}\mathsf{G}_i \\ &= \left(\frac{2}{L_{\mathsf{max}} + 2\tau} \cdot \frac{L_{\mathsf{max}}}{2}\right)\left[\mathsf{U}_i^{\mathsf{T}} - \frac{2}{L_{\mathsf{max}}}\nabla_i g\right] + \left(\frac{2}{L_{\mathsf{max}} + 2\tau} \cdot \frac{2\tau}{2}\right)\left[\mathsf{U}_i^{\mathsf{T}} - \frac{1}{\tau}\mathsf{H}_i\right],\end{aligned}$$

is block nonexpansive. Then from Proposition 4 in Supplement D.2, we know that G_i is block $1/(L_{\mathsf{max}} + 2\tau)$-cocoercive.

(d) Consider any $\boldsymbol{x}^* \in \mathsf{zer}(\mathsf{G})$, an index $i \in \{1, \dots, b\}$ picked uniformly at random, and a single iteration of BC-RED $\boldsymbol{x}^+ = \boldsymbol{x} - \gamma \mathsf{U}_i \mathsf{G}_i \boldsymbol{x}$. Define a vector $\boldsymbol{h}_i \coloneqq \mathsf{U}_i^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{x}^*) \in \mathbb{R}^{n_i}$. We then have

$$\begin{aligned}\|\boldsymbol{x}^+ - \boldsymbol{x}^*\|^2 &= \|\boldsymbol{x} - \boldsymbol{x}^* - \gamma \mathsf{U}_i \mathsf{G}_i \boldsymbol{x}\|^2 \\ &= \|\boldsymbol{x} - \boldsymbol{x}^*\| - 2\gamma(\mathsf{U}_i \mathsf{G}_i \boldsymbol{x})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{x}^*) + \gamma^2 \|\mathsf{G}_i \boldsymbol{x}\|^2 \\ &= \|\boldsymbol{x} - \boldsymbol{x}^*\| - 2\gamma(\mathsf{G}_i \boldsymbol{x} - \mathsf{G}_i \boldsymbol{x}^*)^{\mathsf{T}} \boldsymbol{h}_i + \gamma^2 \|\mathsf{G}_i \boldsymbol{x}\|^2 \\ &\leq \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 - \frac{1}{L_{\mathsf{max}} + 2\tau}\left(2\gamma - (L_{\mathsf{max}} + 2\tau)\gamma^2\right)\|\mathsf{G}_i \boldsymbol{x}\|^2 \\ &\leq \|\boldsymbol{x} - \boldsymbol{x}^*\|^2 - \frac{\gamma}{L_{\mathsf{max}} + 2\tau}\|\mathsf{G}_i \boldsymbol{x}\|^2, \end{aligned} \tag{14}$$

where in the third line we used $\mathsf{G}_i \boldsymbol{x}^* = \mathsf{U}_i^{\mathsf{T}} \mathsf{G} \boldsymbol{x}^* = \boldsymbol{0}$, in the fourth line the block cocoercivity of G_i, and in the last line the fact that $0 < \gamma \leq 1/(L_{\mathsf{max}} + 2\tau)$.

(e) By taking a conditional expectation on both sides and rearranging the terms, we obtain

$$\begin{aligned}\frac{\gamma}{L_{\mathsf{max}} + 2\tau}\mathbb{E}\left[\|\mathsf{G}_i \boldsymbol{x}\|^2 | \boldsymbol{x}\right] = \frac{\gamma}{b(L_{\mathsf{max}} + 2\tau)}\sum_{i=1}^{b}\|\mathsf{G}_i \boldsymbol{x}\|^2 &= \frac{\gamma}{b(L_{\mathsf{max}} + 2\tau)}\|\mathsf{G}\boldsymbol{x}\|^2 \\ &\leq \mathbb{E}\left[\|\boldsymbol{x} - \boldsymbol{x}^*\|^2 - \|\boldsymbol{x}^+ - \boldsymbol{x}^*\|^2 | \boldsymbol{x}\right]\end{aligned}$$

(f) Hence by averaging over $t \geq 1$ iterations and taking the total expectation

$$\mathbb{E}\left[\frac{1}{t}\sum_{k=1}^{t}\|\mathsf{G}\boldsymbol{x}^{k-1}\|^2\right] \leq \frac{1}{t}\left[\frac{b(L_{\mathsf{max}}+2\tau)}{\gamma}\|\boldsymbol{x}^0-\boldsymbol{x}^*\|^2\right] \leq \frac{1}{t}\left[\frac{b(L_{\mathsf{max}}+2\tau)}{\gamma}R_0^2\right].$$

The last inequality directly leads to the result.

Remark. Eq. (14) implies that, under Assumptions 1-3, the iterates of BC-RED satisfy

$$\|\boldsymbol{x}^t-\boldsymbol{x}^*\| \leq \|\boldsymbol{x}^{t-1}-\boldsymbol{x}^*\| \leq \cdots \leq \|\boldsymbol{x}^0-\boldsymbol{x}^*\| \leq R_0, \tag{15}$$

which means that the distance of the iterates of BC-RED to $\mathsf{zer}(\mathsf{G})$ is nonincreasing.

Remark. Suppose we are solving a *coordinate friendly problem* [39], in which the cost of the full gradient update is b times the cost of block update. Consider the step-size $\gamma = 1/(L+2\tau)$ where L is the global Lipschitz constant of the gradient method. A similar analysis as above would yield the following convergence rate for the gradient method

$$\frac{1}{t}\sum_{k=1}^{t}\|\mathsf{G}\boldsymbol{x}^{k-1}\|^2 \leq \frac{(L+2\tau)^2R_0^2}{t}$$

Now, consider the step-size $\gamma = 1/(L_{\mathsf{max}}+2\tau)$ and suppose that we run $(t\cdot b)$ updates of BC-RED with $t \geq 1$. Then, we have that

$$\mathbb{E}\left[\frac{1}{tb}\sum_{k=1}^{tb}\|\mathsf{G}\boldsymbol{x}^{k-1}\|^2\right] \leq \frac{(L_{\mathsf{max}}+2\tau)^2R_0^2}{t}.$$

Since $L_{\mathsf{max}} \leq L \leq bL_{\mathsf{max}}$, where the upper bound can sometimes be tight, we conclude that the expected complexity of the block-coordinate algorithm is lower compared to the full algorithm.

B Proof of Theorem 2

The concept of Moreau smoothing is well-known and has been extensively used in other contexts (see for example [52]). Our contribution is to formally connect the concept to RED-based algorithms, which leads to its novel justification as an approximate MAP estimator. The basic review of relevant concepts from proximal optimization is given in Supplement D.4.

For $\tau > 0$, we consider the Moreau envelope of h

$$h_{(1/\tau)}(\boldsymbol{x}) \coloneqq \min_{\boldsymbol{z}\in\mathbb{R}^n}\left\{\frac{1}{2}\|\boldsymbol{z}-\boldsymbol{x}\|^2 + (1/\tau)h(\boldsymbol{z})\right\}.$$

From Proposition 9 in Supplement D.4 we know that

$$0 \leq h(\boldsymbol{x}) - \tau h_{(1/\tau)}(\boldsymbol{x}) \leq \frac{G_0}{2\tau} \tag{16}$$

and from Proposition 8 in Supplement D.4, we know that

$$\tau\nabla h_{(1/\tau)}(\boldsymbol{x}) = \tau(\boldsymbol{x} - \mathsf{prox}_{(1/\tau)h}(\boldsymbol{x})). \tag{17}$$

Hence, we can express the function f as follows

$$\begin{aligned} f(\boldsymbol{x}) &= g(\boldsymbol{x}) + h(\boldsymbol{x}) \\ &= (g(\boldsymbol{x}) + \tau h_{(1/\tau)}(\boldsymbol{x})) + (h(\boldsymbol{x}) - \tau h_{(1/\tau)}(\boldsymbol{x})) \\ &= f_{(1/\tau)}(\boldsymbol{x}) + (h(\boldsymbol{x}) - \tau h_{(1/\tau)}(\boldsymbol{x})), \end{aligned}$$

where $f_{(1/\tau)} \coloneqq g + \tau h_{(1/\tau)}$. From eq. (17), we conclude that a single iteration of BC-RED

$$\boldsymbol{x}^+ = \boldsymbol{x} - \gamma\mathsf{U}_i\mathsf{G}_i\boldsymbol{x} \quad \text{with} \quad \mathsf{G}_i = \mathsf{U}_i^{\mathsf{T}}(\nabla g(\boldsymbol{x}) + \tau\nabla h_{(1/\tau)}(\boldsymbol{x}))$$

is performing a block-coordinate descent on the function $f_{(1/\tau)}$. From eq. (16) and the convexity of the Moreau envelope, we have

$$f^*_{(1/\tau)} = f_{(1/\tau)}(\boldsymbol{x}^*) \leq f_{(1/\tau)}(\boldsymbol{x}) \leq f(\boldsymbol{x}), \quad \boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{x}^*\in\mathsf{zer}(\mathsf{G}).$$

Hence, there exists a finite f^* such that $f(\boldsymbol{x}) \geq f^*$ with $f^*_{(1/\tau)} \leq f^*$. Consider the iteration $t \geq 1$ of BC-RED, then we have that

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{x}^t)] - f^* &\leq \mathbb{E}[f(\boldsymbol{x}^t)] - f^*_{(1/\tau)} \\ &= (\mathbb{E}[f_{(1/\tau)}(\boldsymbol{x}^t)] - f^*_{(1/\tau)}) + \mathbb{E}[(h(\boldsymbol{x}^t) - \tau h_{(1/\tau)}(\boldsymbol{x}^t)]) \\ &\leq \frac{2b}{\gamma t} R_0^2 + \frac{G_0^2}{2\tau},\end{aligned}$$

where we applied (10), which is further discussed in Supplement C.

The proof of eq. (13) is directly obtained by setting $\tau = \sqrt{t}$, $\gamma = L_{\mathsf{max}} + 2\sqrt{t}$, and noting that $t \geq \sqrt{t}$, for all $t \geq 1$.

C Convergence of the Traditional Coordinate Descent

The following analysis has been adopted from [25]. We include it here for completeness.

Consider the following denoiser

$$\mathsf{D}(\boldsymbol{x}) = \boldsymbol{x} - \frac{1}{\tau}\nabla h(\boldsymbol{x}), \quad \tau > 0, \quad \boldsymbol{x} \in \mathbb{R}^n,$$

and the following function

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + h(\boldsymbol{x})$$

where g and h are both convex and continuously differentiable. For this denoiser, we have that

$$\mathsf{G}(\boldsymbol{x}) = \nabla g(\boldsymbol{x}) + \tau(\boldsymbol{x} - \mathsf{D}(\boldsymbol{x})) = \nabla g(\boldsymbol{x}) + \nabla h(\boldsymbol{x}) = \nabla f(\boldsymbol{x}).$$

Therefore, in this case, BC-RED is minimizing a convex and smooth function f, which means that any $\boldsymbol{x}^* \in \mathsf{zer}(\mathsf{G})$ is a global minimizer of f. Additionally, due to Proposition 2 in Supplement D.1 and Proposition 7 in Supplement D.3, we have

$$\mathsf{D}_i \text{ is block nonexpansive} \quad \Leftrightarrow \quad \nabla_i h \text{ is block } 2\tau\text{-Lipschitz continuous.}$$

Hence, for such denoisers, Assumption 3 is equivalent to the 2τ-Lipschitz smoothness of block gradients $\nabla_i h$.

To prove eq. 10, we consider the following iteration

$$\boldsymbol{x}^+ = \boldsymbol{x} - \mathsf{U}_i\mathsf{G}_i\boldsymbol{x} \quad \text{with} \quad \mathsf{G}_i = \nabla_i f = \nabla_i g + \nabla_i h,$$

which under our assumptions is a special case of the setting for Theorem 1.

(a) From the block Lipscthiz continuity of f, we conclude that

$$\begin{aligned}f(\boldsymbol{x}^+) &\leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^{\mathsf{T}}(\boldsymbol{x}^+ - \boldsymbol{x}) + \frac{(L_{\mathsf{max}} + 2\tau)}{2}\|\boldsymbol{x}^+ - \boldsymbol{x}\|^2 \\ &= f(\boldsymbol{x}) - \gamma\|\nabla_i f(\boldsymbol{x})\|^2 + \frac{\gamma^2(L_{\mathsf{max}} + 2\tau)}{2}\|\nabla_i f(\boldsymbol{x})\|^2 \\ &\leq f(\boldsymbol{x}) - \frac{\gamma}{2}\|\nabla_i f(\boldsymbol{x})\|^2,\end{aligned}$$

where the last inequality comes from the fact that $\gamma \leq 1/(L_{\mathsf{max}} + 2\tau)$.

(b) For all $t \geq 1$, define

$$\varphi_t \coloneqq \mathbb{E}\left[f(\boldsymbol{x}^t)\right] - f(\boldsymbol{x}^*).$$

Then from (a), we can conclude that

$$\varphi_t \leq \varphi_{t-1} - \frac{\gamma}{2b}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}^{t-1})\|^2\right] \leq \varphi_{t-1} - \frac{\gamma}{2b}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}^{t-1})\|\right]^2,$$

where in the last inequality we used the Jensen's inequality, and the fact that

$$\mathbb{E}\left[\|\nabla_i f(\boldsymbol{x}^{t-1})\|^2\right] = \mathbb{E}\left[\mathbb{E}\left[\|\nabla_i f(\boldsymbol{x}^{t-1})\|^2 | \boldsymbol{x}^{t-1}\right]\right] = \mathbb{E}\left[\frac{1}{b}\sum_{i=1}^{b}\|\nabla_i f(\boldsymbol{x}^t)\|^2\right] = \frac{1}{b}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}^{t-1})\|^2\right].$$

(c) From convexity, we know that

$$\varphi_t = \mathbb{E}\left[f(\boldsymbol{x}^t)\right] - f(\boldsymbol{x}^*) \leq \mathbb{E}\left[\nabla f(\boldsymbol{x}^t)^{\mathsf{T}}(\boldsymbol{x}^t - \boldsymbol{x}^*)\right] \leq \mathbb{E}\left[\|\nabla f(\boldsymbol{x}^t)\|\|\boldsymbol{x}^t - \boldsymbol{x}^*\|\right] \leq R_0 \cdot \mathbb{E}\left[\|\nabla f(\boldsymbol{x}^t)\|\right],$$

where in the last inequality, we used eq. (15). This combined with the result of (b) implies that

$$\varphi_t \leq \varphi_{t-1} - \frac{\gamma}{2b}\frac{\varphi_{t-1}^2}{R_0^2}.$$

(d) Note that from (c), we can obtain

$$\frac{1}{\varphi_t} - \frac{1}{\varphi_{t-1}} = \frac{\varphi_{t-1} - \varphi_t}{\varphi_t \varphi_{t-1}} \geq \frac{\varphi_{t-1} - \varphi_t}{\varphi_{t-1}^2} \geq \frac{\gamma}{2bR_0^2}.$$

By iterating this inequality, we get the final result

$$\frac{1}{\varphi_t} \geq \frac{1}{\varphi_0} + \frac{\gamma t}{2b\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2} \geq \frac{\gamma t}{2bR_0^2} \quad \Rightarrow \quad \varphi_t \leq \frac{2b}{\gamma t}R_0^2.$$

D Background Material

The results in this section are well-known in the optimization literature and can be found in different forms in standard textbooks [43, 51, 62, 63]. For completeness, we summarize the key results useful for our analysis by restating them in a block-coordinate form.

D.1 Properties of Block-Coordinate Operators

Most of the concepts in this part come from the traditional monotone operator theory [43, 44] adapted for block-coordinate operators.

Definition 1. *We define the* block-coordinate operator $\mathsf{T}_i : \mathbb{R}^n \to \mathbb{R}^{n_i}$ *of* $\mathsf{T} : \mathbb{R}^n \to \mathbb{R}^n$ *as*

$$\mathsf{T}_i\boldsymbol{x} \coloneqq [\mathsf{T}\boldsymbol{x}]_i = \mathsf{U}_i^{\mathsf{T}}\mathsf{T}\boldsymbol{x} \in \mathbb{R}^{n_i}, \quad \boldsymbol{x} \in \mathbb{R}^n.$$

The operator T_i *applies* T *to its input vector and then extracts the subset of outputs corresponding to the coordinates in the block* $i \in \{1, \ldots, b\}$.

Remark. When $b = 1$, we have that $n = n_1$ and $\mathsf{U}_1 = \mathsf{U}_1^{\mathsf{T}} = \mathsf{I}$. Then, all the properties in this section reduce to their standard counterparts from the monotone operator theory in $\mathbb{R}^n$. In such settings, we simply drop the word *block* from the name of the property.

Definition 2. T_i *is* block Lipschitz continuous with constant $\lambda_i > 0$ *if*

$$\|\mathsf{T}_i\boldsymbol{x} - \mathsf{T}_i\boldsymbol{y}\| \leq \lambda_i\|\boldsymbol{h}_i\|, \quad \boldsymbol{x} = \boldsymbol{y} + \mathsf{U}_i\boldsymbol{h}_i, \quad \boldsymbol{y} \in \mathbb{R}^n, \boldsymbol{h}_i \in \mathbb{R}^{n_i}.$$

When $\lambda_i = 1$*, we say that* T_i *is* block nonexpansive.

Definition 3. *An operator* T_i *is* block cocoercive with constant $\beta_i > 0$ *if*

$$(\mathsf{T}_i\boldsymbol{x} - \mathsf{T}_i\boldsymbol{y})^{\mathsf{T}}\boldsymbol{h}_i \geq \beta_i\|\mathsf{T}_i\boldsymbol{x} - \mathsf{T}_i\boldsymbol{y}\|^2, \qquad \boldsymbol{x} = \boldsymbol{y} + \mathsf{U}_i\boldsymbol{h}_i, \quad \boldsymbol{y} \in \mathbb{R}^n, \boldsymbol{h}_i \in \mathbb{R}^{n_i}.$$

When $\beta_i = 1$*, we say that* T_i *is* block firmly nonexpansive.

The following propositions are conclusions derived from the definition of above.

Proposition 1. *Let* $\mathsf{T}_{ij} : \mathbb{R}^n \to \mathbb{R}^{n_i}$ *for* $j \in J$ *be a set of block nonexpansive operators. Then, their convex combination*

$$\mathsf{T}_i \coloneqq \sum_{j \in J} \theta_j \mathsf{T}_{ij}, \quad \textit{with} \quad \theta_j > 0 \textit{ and } \sum_{j \in J} \theta_j = 1,$$

is nonexpansive.

Proof. By using the triangular inequality and the definition of block nonexpansiveness, we obtain

$$\|\mathsf{T}_i\boldsymbol{x} - \mathsf{T}_i\boldsymbol{y}\| \leq \sum_{j\in J}\theta_j\|\mathsf{T}_{ij}\boldsymbol{x} - \mathsf{T}_{ij}\boldsymbol{y}\| \leq \left(\sum_{j\in J}\theta_j\right)\|\boldsymbol{h}_i\| = \|\boldsymbol{h}_i\|,$$

for all $\boldsymbol{y} \in \mathbb{R}^n$ and $\boldsymbol{h}_i \in \mathbb{R}^{n_i}$ where $\boldsymbol{x} = \boldsymbol{y} + \mathsf{U}_i\boldsymbol{h}_i$. □

Proposition 2. *Consider* $\mathsf{R}_i = \mathsf{U}_i^\mathsf{T} - \mathsf{T}_i$ *where* $\mathsf{T}_i : \mathbb{R}^n \to \mathbb{R}^{n_i}$.

$$\mathsf{T}_i \textit{ is block nonexpansive} \quad \Leftrightarrow \quad \mathsf{R}_i \textit{ is } (1/2)\textit{-block cocoercive.}$$

Proof. First suppose that R_i is $1/2$ block cocoercive. Let $\boldsymbol{x} = \boldsymbol{y} + \mathsf{U}_i\boldsymbol{h}_i$ for all $\boldsymbol{y} \in \mathbb{R}^n$ and $\boldsymbol{h}_i \in \mathbb{R}^{n_i}$. We then have

$$\frac{1}{2}\|\mathsf{R}_i\boldsymbol{x} - \mathsf{R}_i\boldsymbol{y}\|^2 \leq (\mathsf{R}_i\boldsymbol{x} - \mathsf{R}_i\boldsymbol{y})^\mathsf{T}\boldsymbol{h}_i = \|\boldsymbol{h}_i\|^2 - (\mathsf{T}_i\boldsymbol{x} - \mathsf{T}_i\boldsymbol{y})^\mathsf{T}\boldsymbol{h}_i.$$

We also have that

$$\frac{1}{2}\|\mathsf{R}_i\boldsymbol{x} - \mathsf{R}_i\boldsymbol{y}\|^2 = \frac{1}{2}\|\boldsymbol{h}_i\|^2 - (\mathsf{T}_i\boldsymbol{x} - \mathsf{T}_i\boldsymbol{y})^\mathsf{T}\boldsymbol{h}_i + \frac{1}{2}\|\mathsf{T}_i\boldsymbol{x} - \mathsf{T}_i\boldsymbol{y}\|^2.$$

By combining these two and simplifying the expression, we obtain that

$$\|\mathsf{T}_i\boldsymbol{x} - \mathsf{T}_i\boldsymbol{y}\| \leq \|\boldsymbol{h}_i\|.$$

The converse can be proved by following this logic in reverse. □

D.2 Block Averaged Operators

It is well known that the iteration of a nonexpansive operator does not necessarily converge. To see this consider a nonexpansive operator $\mathsf{T} = -\mathsf{I}$, where I is identity. However, it is also well known that the convergence can be established for averaged operators.

Definition 4. *For a constant* $\alpha \in (0,1)$*, we say that the operator* T *is* α-averaged*, if there exists a nonexpansive operator* N *such that* $\mathsf{T} = (1-\alpha)\mathsf{I} + \alpha\mathsf{N}$.

Definition 5. *For a constant* $\alpha \in (0,1)$*, we say that* $\mathsf{T}_i : \mathbb{R}^n \to \mathbb{R}^{n_i}$ *is* block α-averaged*, if there exists a block nonexpansive operator* N_i *such that* $\mathsf{T}_i = (1-\alpha)\mathsf{U}_i^\mathsf{T} + \alpha\mathsf{N}_i$.

Remark. It is clear that if T is α-averaged, then $\mathsf{T}_i = \mathsf{U}_i^\mathsf{T}\mathsf{T}$ is block α-averaged.

The following characterization is often convenient.

Proposition 3. *For a block nonexpansive operator* T_i*, a constant* $\alpha \in (0,1)$*, and the operator* $\mathsf{R}_i := \mathsf{U}_i^\mathsf{T} - \mathsf{T}_i$*, the following are equivalent*

(a) T_i *is block* α*-averaged*

(b) $(1-1/\alpha)\mathsf{U}_i^\mathsf{T} + (1/\alpha)\mathsf{T}_i$ *is block nonexpansive*

(c) $\|\mathsf{T}_i\boldsymbol{x} - \mathsf{T}_i\boldsymbol{y}\|^2 \leq \|\boldsymbol{h}_i\|^2 - \left(\frac{1-\alpha}{\alpha}\right)\|\mathsf{R}_i\boldsymbol{x} - \mathsf{R}_i\boldsymbol{y}\|^2, \quad \boldsymbol{x} = \boldsymbol{y} + \mathsf{U}_i\boldsymbol{h}_i, \quad \boldsymbol{y} \in \mathbb{R}^n, \boldsymbol{h}_i \in \mathbb{R}^{n_i}$

Proof. The equivalence of (a) and (b) is clear from the definition. To establish the equivalence with (c), consider an operator N_i and $\mathsf{T}_i = (1-\alpha)\mathsf{U}_i^\mathsf{T} + \alpha\mathsf{N}_i$. Note that

$$\mathsf{R}_i = \mathsf{U}_i^\mathsf{T} - \mathsf{T}_i = \alpha(\mathsf{U}_i^\mathsf{T} - \mathsf{N}_i).$$

Then, for all $\boldsymbol{y} \in \mathbb{R}^n$ and $\boldsymbol{h}_i \in \mathbb{R}^{n_i}$, with $\boldsymbol{x} = \boldsymbol{y} + \mathsf{U}_i\boldsymbol{h}_i$, we have that

$$\begin{aligned}
\|\mathsf{T}_i\boldsymbol{x} - \mathsf{T}_i\boldsymbol{y}\|^2 &= \|(1-\alpha)\boldsymbol{h}_i + \alpha(\mathsf{N}_i\boldsymbol{x} - \mathsf{N}_i\boldsymbol{y})\|^2 \\
&= (1-\alpha)\|\boldsymbol{h}_i\|^2 + \alpha\|\mathsf{N}_i\boldsymbol{x} - \mathsf{N}_i\boldsymbol{y}\|^2 - \alpha(1-\alpha)\|\boldsymbol{h}_i - (\mathsf{N}_i\boldsymbol{x} - \mathsf{N}_i\boldsymbol{y})\|^2 \\
&= (1-\alpha)\|\boldsymbol{h}_i\|^2 + \alpha\|\mathsf{N}_i\boldsymbol{x} - \mathsf{N}_i\boldsymbol{y}\|^2 - \left(\frac{1-\alpha}{\alpha}\right)\|\mathsf{R}_i\boldsymbol{x} - \mathsf{R}_i\boldsymbol{y}\|^2, \qquad (18)
\end{aligned}$$

where we used the fact that

$$\|(1-\alpha)\boldsymbol{x}+\alpha\boldsymbol{y}\|^2=(1-\alpha)\|\boldsymbol{x}\|^2+\alpha\|\boldsymbol{y}\|^2-\alpha(1-\alpha)\|\boldsymbol{x}-\boldsymbol{y}\|^2,\quad \theta\in\mathbb{R},\,\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^n.$$

Consider also

$$\|\boldsymbol{h}_i\|^2-\left(\frac{1-\alpha}{\alpha}\right)\|\mathsf{R}_i\boldsymbol{x}-\mathsf{R}_i\boldsymbol{y}\|^2=(1-\alpha)\|\boldsymbol{h}_i\|^2+\alpha\|\boldsymbol{h}_i\|^2-\left(\frac{1-\alpha}{\alpha}\right)\|\mathsf{R}_i\boldsymbol{x}-\mathsf{R}_i\boldsymbol{y}\|^2. \quad (19)$$

It is clear that we have

$$(18)\leq(19)\quad\Leftrightarrow\quad \mathsf{N}_i \text{ is block nonexpansive}\quad\Leftrightarrow\quad \mathsf{T}_i \text{ is block } \alpha\text{-averaged},$$

where for the last equivalence, we used the definition of block averagedness. □

Proposition 4. *Consider a block-coordinate operator $\mathsf{T}_i=\mathsf{U}_i^\mathsf{T}\mathsf{T}$ with $\mathsf{T}:\mathbb{R}^n\to\mathbb{R}^n$. Let $\boldsymbol{x}=\boldsymbol{y}+\mathsf{U}_i\boldsymbol{h}$ with $\boldsymbol{x}\in\mathbb{R}^n$, $\boldsymbol{h}_i\in\mathbb{R}^{n_i}$ and consider $\beta_i>0$. Then, the following are equivalent*

(a) T_i is block β_i-cocoercive

(b) $\beta_i\mathsf{T}_i$ is block firmly nonexpansive

(c) $\mathsf{U}_i^\mathsf{T}-\beta_i\mathsf{T}_i$ is block firmly nonexpansive.

(d) $\beta_i\mathsf{T}_i$ is block $(1/2)$-averaged.

(e) $\mathsf{U}_i^\mathsf{T}-2\beta_i\mathsf{T}_i$ is block nonexpansive.

Proof. The equivalence between (a) and (b) is readily observed by defining $\mathsf{P}_i\coloneqq\beta_i\mathsf{T}_i$ and noting that

$$(\mathsf{P}_i\boldsymbol{x}-\mathsf{P}_i\boldsymbol{y})^\mathsf{T}\boldsymbol{h}_i=\beta_i(\mathsf{T}_i\boldsymbol{x}-\mathsf{T}_i\boldsymbol{y})^\mathsf{T}\boldsymbol{h}_i\quad\text{and}\quad\|\mathsf{P}_i\boldsymbol{x}-\mathsf{P}_i\boldsymbol{y}\|^2=\beta_i^2\|\mathsf{T}_i\boldsymbol{x}-\mathsf{T}_i\boldsymbol{y}\|.$$

Define $\mathsf{R}_i\coloneqq\mathsf{U}_i^\mathsf{T}-\mathsf{P}_i$ and suppose (b) is true, then

$$\begin{aligned}(\mathsf{R}_i\boldsymbol{x}-\mathsf{R}_i\boldsymbol{y})^\mathsf{T}\boldsymbol{h}_i&=\|\boldsymbol{h}_i\|^2-(\mathsf{P}_i\boldsymbol{x}-\mathsf{P}_i\boldsymbol{y})^\mathsf{T}\boldsymbol{h}_i\\&=\|\mathsf{R}_i\boldsymbol{x}-\mathsf{R}_i\boldsymbol{y}\|^2+(\mathsf{P}_i\boldsymbol{x}-\mathsf{P}_i\boldsymbol{y})^\mathsf{T}\boldsymbol{h}_i-\|\mathsf{P}_i\boldsymbol{x}-\mathsf{P}_i\boldsymbol{y}\|^2\\&\geq\|\mathsf{R}_i\boldsymbol{x}-\mathsf{R}_i\boldsymbol{y}\|^2.\end{aligned}$$

By repeating the same argument for $\mathsf{P}_i=\mathsf{U}_i^\mathsf{T}-\mathsf{R}_i$, we establish the full equivalence between (b) and (c).

The full equivalence of (b) and (d) can be established by observing that

$$\begin{aligned}&2\|\mathsf{P}_i\boldsymbol{x}-\mathsf{P}_i\boldsymbol{y}\|^2\leq2(\mathsf{P}_i\boldsymbol{x}-\mathsf{P}_i\boldsymbol{y})^\mathsf{T}\boldsymbol{h}_i\\&\Leftrightarrow\quad\|\mathsf{P}_i\boldsymbol{x}-\mathsf{P}_i\boldsymbol{y}\|^2\leq2(\mathsf{P}_i\boldsymbol{x}-\mathsf{P}_i\boldsymbol{y})^\mathsf{T}\boldsymbol{h}_i-\|\mathsf{P}_i\boldsymbol{x}-\mathsf{P}_i\boldsymbol{y}\|^2\\&\qquad\qquad=\|\boldsymbol{h}_i\|^2-\big(\|\boldsymbol{h}_i\|^2-2(\mathsf{P}_i\boldsymbol{x}-\mathsf{P}_i\boldsymbol{y})^\mathsf{T}\boldsymbol{h}_i+\|\mathsf{P}_i\boldsymbol{x}-\mathsf{P}_i\boldsymbol{y}\|^2\big)\\&\qquad\qquad=\|\boldsymbol{h}_i\|^2-\|\mathsf{R}_i\boldsymbol{x}-\mathsf{R}_i\boldsymbol{y}\|^2.\end{aligned}$$

To show the equivalence with (e), first suppose that $\mathsf{N}_i\coloneqq\mathsf{U}_i^\mathsf{T}-2\mathsf{P}_i$ is block nonexpansive, then $\mathsf{P}_i=\frac{1}{2}(\mathsf{U}_i^\mathsf{T}+(-\mathsf{N}_i))$ is block $1/2$-averaged, which means that it is block firmly nonexpansive. On the other hand, if P_i is block firmly nonexpansive, then it is block $1/2$-averaged, which means that from Proposition 3(b) we have that $(1-2)\mathsf{U}_i^\mathsf{T}+2\mathsf{P}_i=2\mathsf{P}_i-\mathsf{U}_i^\mathsf{T}=-\mathsf{N}_i$ is block nonexpansive. This directly means that N_i is block nonexpansive. □

D.3 Operator Properties for Convex Function

It is convenient to link properties of a function $f : \mathbb{R}^n \to \mathbb{R}$, $\boldsymbol{x} \mapsto y = f(\boldsymbol{x})$, to the properties of operators derived from it. The key properties for our analysis are related to continuity and convexity.

Proposition 5. *Let f be continuously differentiable function with $\nabla_i f$ that is block L_i-Lipschitz continuous. Then,*

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^{\mathsf{T}}(\boldsymbol{y} - \boldsymbol{x}) + \frac{L_i}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 = f(\boldsymbol{x}) + \nabla_i f(\boldsymbol{x})^{\mathsf{T}}\boldsymbol{h}_i + \frac{L_i}{2}\|\boldsymbol{h}_i\|^2$$

for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{h}_i \in \mathbb{R}^{n_i}$, where $\boldsymbol{y} = \boldsymbol{x} + \mathsf{U}_i\boldsymbol{h}_i$.

Proof. The proof is a minor variation of the one presented in Section 2.1 of [63]. □

Proposition 6. *Consider a continuously differentiable f such that $\nabla_i f$ is block L_i-Lipschitz continuous. Let $\boldsymbol{x}^* \in \mathbb{R}^n$ denote the global minimizer of f. Then, we have that*

$$\frac{1}{2L_i}\|\nabla_i f(\boldsymbol{x})\|^2 \leq (f(\boldsymbol{x}) - f(\boldsymbol{x}^*)) \leq \frac{L_i}{2}\|\boldsymbol{x} - \boldsymbol{x}^*\|^2, \quad \boldsymbol{x} = \boldsymbol{x}^* + \mathsf{U}_i\boldsymbol{h}_i, \quad \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{h}_i \in \mathbb{R}^{n_i}.$$

Proof. The proof is a minor variation of the discussion in Section 9.1.2 of [62]. □

Proposition 7. *For a convex and continuously differentiable function f, we have*

$$\nabla_i f \text{ is block } L_i\text{-Lipschitz continuous} \quad \Leftrightarrow \quad \nabla_i f \text{ is block } (1/L_i)\text{-cocoercive}.$$

Proof. The proof is a minor variation of the one presented as Theorem 2.1.5 in Section 2.1 of [63]. □

D.4 Moreau smoothing and proximal operators

In this section, we consider a class of functions that are proper, closed, and convex, but are not necessarily differentiable. The proximal operator is a widely-used concept in such nonsmooth optimization problems [50, 51].

Definition 6. *Consider a proper, closed, and convex h and a constant $\mu > 0$. We define the* proximal operator

$$\mathsf{prox}_{\mu h}(\boldsymbol{x}) \coloneqq \operatorname*{arg\,min}_{\boldsymbol{z} \in \mathbb{R}^n} \left\{ \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{x}\|^2 + \mu h(\boldsymbol{z}) \right\}$$

and the Moreau envelope

$$h_\mu(\boldsymbol{x}) \coloneqq \min_{\boldsymbol{z} \in \mathbb{R}^n} \left\{ \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{x}\|^2 + \mu h(\boldsymbol{z}) \right\}.$$

Proposition 8. *The function h_μ is convex and continuously differentiable with a 1-Lipschitz gradient*

$$\nabla h_\mu(\boldsymbol{x}) = \boldsymbol{x} - \mathsf{prox}_{\mu h}(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^n.$$

Proof. We first show that h_μ is convex. Consider

$$q(\boldsymbol{x}, \boldsymbol{z}) \coloneqq \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{x}\|^2 + \mu h(\boldsymbol{z}),$$

which is convex $(\boldsymbol{x}, \boldsymbol{z})$. Then, for any $0 \leq \theta \leq 1$ and $(\boldsymbol{x}_1, \boldsymbol{z}_1), (\boldsymbol{x}_2, \boldsymbol{z}_2) \in \mathbb{R}^{2n}$, we have

$$h_\mu(\theta\boldsymbol{x}_1 + (1-\theta)\boldsymbol{x}_2) \leq q(\theta\boldsymbol{x}_1 + (1-\theta)\boldsymbol{x}_2, \theta\boldsymbol{z}_1 + (1-\theta)\boldsymbol{z}_2) \leq \theta q(\boldsymbol{x}_1, \boldsymbol{z}_1) + (1-\theta)q(\boldsymbol{x}_2, \boldsymbol{z}_2),$$

where we used the convexity of q. Since this inequality holds everywhere, we have

$$h_\mu(\theta\boldsymbol{x}_1 + (1-\theta)\boldsymbol{x}_2) \leq \theta h_\mu(\boldsymbol{x}_1) + (1-\theta)h_\mu(\boldsymbol{x}_2),$$

with $h_\mu(\boldsymbol{x}_1) = \min_{\boldsymbol{z}_1} q(\boldsymbol{x}_1, \boldsymbol{z}_1)$ and $h_\mu(\boldsymbol{x}_2) = \min_{\boldsymbol{z}_2} q(\boldsymbol{x}_2, \boldsymbol{z}_2)$.

To show the differentiability, note that

$$\begin{aligned} h_\mu(\boldsymbol{x}) &= \frac{1}{2}\|\boldsymbol{x}\|^2 - \max_{\boldsymbol{z}\in\mathbb{R}^n}\left\{\boldsymbol{x}^\mathsf{T}\boldsymbol{z} - \mu h(\boldsymbol{z}) - \frac{1}{2}\|\boldsymbol{z}\|^2\right\} \\ &= \frac{1}{2}\|\boldsymbol{x}\|^2 - \phi^\star(\boldsymbol{x}) \quad \text{with} \quad \phi(\boldsymbol{z}) \coloneqq \frac{1}{2}\|\boldsymbol{z}\|^2 + \mu h(\boldsymbol{z}), \end{aligned}$$

where $\phi^\star$ denotes the conjugate of ϕ. The function ϕ is closed and 1-strongly convex. Hence, we know that $\phi^\star$ is defined for all $\boldsymbol{x} \in \mathbb{R}^n$ and is differentiable with gradient [62]

$$\nabla\phi^\star(\boldsymbol{x}) = \operatorname*{arg\,max}_{\boldsymbol{z}\in\mathbb{R}^n}\left\{\boldsymbol{x}^\mathsf{T}\boldsymbol{z} - \mu h(\boldsymbol{z}) - \frac{1}{2}\|\boldsymbol{z}\|^2\right\} = \mathsf{prox}_{\mu h}(\boldsymbol{x}).$$

Hence, we conclude that

$$\nabla h_\mu(\boldsymbol{x}) = \boldsymbol{x} - \nabla\phi^\star(\boldsymbol{x}) = \boldsymbol{x} - \mathsf{prox}_{\mu h}(\boldsymbol{x}).$$

Note that since the proximal operator is firmly nonexpansive, ∇h_μ is also firmly nonexpansive, which means that it is 1-Lipschitz. □

The next result shows that the Moreau envelope can serve as a smooth approximation to a nonsmooth function.

Proposition 9. *Consider $h \in \mathbb{R}^n$ and its Moreau envelope $h_\mu(\boldsymbol{x})$ for $\mu > 0$. Then,*

$$0 \le h(\boldsymbol{x}) - \frac{1}{\mu}h_\mu(\boldsymbol{x}) \le \frac{\mu}{2}G_{\boldsymbol{x}}^2 \quad \textit{with} \quad G_{\boldsymbol{x}}^2 \coloneqq \min_{\boldsymbol{g}\in\partial h(\boldsymbol{x})}\|\boldsymbol{g}\|^2, \quad \boldsymbol{x}\in\mathbb{R}^n.$$

Proof. First note that

$$\frac{1}{\mu}h_\mu(\boldsymbol{x}) = \min_{\boldsymbol{z}\in\mathbb{R}^n}\left\{\frac{1}{2\mu}\|\boldsymbol{z}-\boldsymbol{x}\|^2 + h(\boldsymbol{z})\right\} \le h(\boldsymbol{x}), \quad \boldsymbol{x}\in\mathbb{R}^n,$$

which is due to the fact that $\boldsymbol{z} = \boldsymbol{x}$ is potentially suboptimal. We additionally have for any $\boldsymbol{g} \in \partial h(\boldsymbol{x})$

$$\begin{aligned} h_\mu(\boldsymbol{x}) - \mu h(\boldsymbol{x}) &= \min_{\boldsymbol{z}\in\mathbb{R}^n}\left\{\mu h(\boldsymbol{z}) - \mu h(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{z}-\boldsymbol{x}\|^2\right\} \\ &\ge \min_{\boldsymbol{z}\in\mathbb{R}^n}\left\{\mu\boldsymbol{g}^\mathsf{T}(\boldsymbol{z}-\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{z}-\boldsymbol{x}\|^2\right\} \\ &= \min_{\boldsymbol{z}\in\mathbb{R}^n}\left\{\frac{1}{2}\|\boldsymbol{z}-(\boldsymbol{x}-\mu\boldsymbol{g})\|^2 - \frac{\mu^2}{2}\|\boldsymbol{g}\|^2\right\} \\ &= -\frac{\mu^2}{2}\|\boldsymbol{g}\|^2. \end{aligned}$$

This directly leads to the conclusion. □

E Additional Technical Details

In this section, we discuss several technical details that we omitted from the main paper for space. Section E.1 discusses issues related to implementation and computational complexity of BC-RED. Section E.2 discusses the architecture of our own CNN denoiser DnCNN* and provides details on its training. Section E.3 discusses the influence of the Lipschitz constant of the CNN denoiser on its performance as a denoising prior.

E.1 Computational Complexity and a Coordinate-Friendly Implementation

Theoretical analysis in Section 4 of the main paper suggests that, if b updates of BC-RED (each modifying a single block) are counted as a single iteration, the worst-case convergence rate of BC-RED is expected to be better than that of the full-gradient RED. This fact was empirically validated in Section 5, where we showed that in practice BC-RED needs much fewer iterations to converge. However, the overall computational complexity of two methods depends on their

per-iteration complexities. In particular, the overall complexity of BC-RED is favorable when its total number of iterations required for convergence offsets the cost of solving the problem in a block-coordinate fashion. As for traditional coordinate descent methods [39, 60], in many problems of interest, the computational complexity of a single update of BC-RED will be roughly b times lower than that of the full-gradient method.

The computational complexity of each block-update will depend on the specifics of the data-fidelity term g and the denoiser D used in the estimation problem. For example, consider the problem where $g(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$. Additionally, suppose that $\boldsymbol{x}$ is such that it is sufficient represent its prior with a block-wise denoiser on each $\boldsymbol{x}_i$, rather than on the full $\boldsymbol{x}$. This situation is very common in image processing, where many popular denoisers are applied block-wise [42]. Then, one can obtain a very efficient implementation of BC-RED, illustrated in Algorithm 2.

The worst-case complexity of applying $\boldsymbol{A}_i$ and $\boldsymbol{A}_i^{\mathsf{T}}$ is $O(mn_i)$, which means that the cost of b updates such updates for $i \in \{1, \ldots, b\}$ is $O(mn)$. Additionally, if the complexity of b block-wise denoising operations is equivalent or less than the complexity of denoising the full vector (which is generally true for advanced denoisers), then the complexity of b updates of BC-RED will be equivalent or better than a single iteration of the full-gradient RED.

Some of our simulations were conducted using denoisers applied on the full-image and others using block-wise denoisers. In particular, the convergence simulations in Fig. 2 and Fig. 6 relied on the full-image denoisers, in order to use identical denoisers for both RED and BC-RED and be fully compatible with the theoretical analysis. On the other hand, the SNR results in Table 1, Table 2, Fig. 7, and Fig. 8 rely on block-wise denoisers, where the denoiser input includes an additional 40 pixel padding around the block and the output has the exact size of the block. The padding size was determined empirically in order to have a close match between BC-RED and RED. We have observed that having even larger paddings does not influence the results of BC-RED. Finally, the size of the denoiser input and output for the galaxy simulations in Fig. 3 and Fig. 10 exactly matches the block size, with no additional padding.

Algorithm 2 BC-RED for the least-squares data-fidelity and a block-wise denoiser

input: initial value $\boldsymbol{x}^0 \in \mathbb{R}^n$, parameter $\tau > 0$, and step-size $\gamma > 0$.
initialize: $\boldsymbol{r}^0 \leftarrow \boldsymbol{A}\boldsymbol{x}^0 - \boldsymbol{y}$
for $k = 1, 2, 3, \ldots$ **do**
  Choose an index $i_k \in \{1, \ldots, b\}$
  $\boldsymbol{x}^k \leftarrow \boldsymbol{x}^{k-1} - \gamma \mathsf{U}_{i_k} \mathsf{G}_{i_k}(\boldsymbol{x}^{k-1})$ with $\mathsf{G}_{i_k}(\boldsymbol{x}^{k-1}) = \boldsymbol{A}_{i_k}^{\mathsf{T}} \boldsymbol{r}^{k-1} + \tau(\boldsymbol{x}_{i_k} - \mathsf{D}(\boldsymbol{x}_{i_k}))$.
  $\boldsymbol{r}^k \leftarrow \boldsymbol{r}^{k-1} - \gamma \boldsymbol{A}_{i_k} \mathsf{G}_{i_k}(\boldsymbol{x}^{k-1})$
end for

E.2 Architecture and Training of DnCNN*

We designed DnCNN* fully based on DnCNN architecture. The network contains three parts. The first part is a composite convolutional layer, consisting of a normal convolutional layer and a rectified linear units (ReLU) layer. It convolves the $n_1 \times n_2$ input to $n_1 \times n_2 \times 64$ features maps by using 64 filters of size 3×3. The second part is a sequence of 5 composite convolutional layers, each having 64 filters of size $3 \times 3 \times 64$. Those composite layers further processes the feature maps generated by the first part. The third part of the network, a single convolutional layer, generates the final output image by convolving the feature maps with a $3 \times 3 \times 64$ filter. Every convolution is performed with a stride $= 1$, so that the intermediate feature maps share the same spatial size of the input image. Fig. 4 visualizes the architectural details. We generated 52000 training examples by adding AWGN to 13000 images (320×320) from the NYU fastMRI dataset [55] and cropping them into 4 sub-images of size 160×160 pixels. We trained DnCNN* to optimize the *mean squared error* by using the Adam optimizer.

E.3 Influence of the Lipschitz Constant on Performance

Our theoretical analysis in Theorem 1 assumes that the denoiser each block denoiser D_i of D is block-nonexpansive. It is relatively straightforward to control the global Lipscthiz constants of

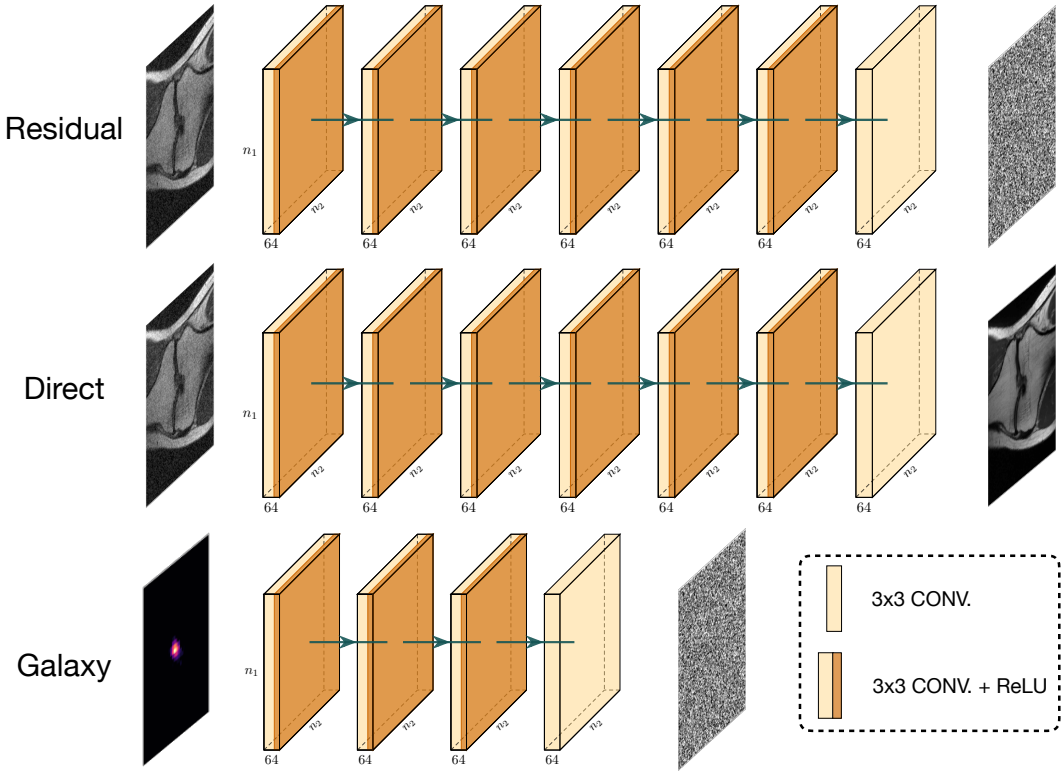

Figure 4: *The architecture of three variants of DnCNN* used in our simulations. Each neural net is trained to remove AWGN from noisy input images.* ***Residual DnCNN**** *is trained to predict the noise from the input. The final desired denoiser* D *is obtained by simply subtracting the predicted noise from the input* $\mathsf{D}(\boldsymbol{z}) = \boldsymbol{z} - \mathsf{DnCNN}^*(\boldsymbol{z})$. ***Direct DnCNN**** *is trained to directly output a clean image from a noisy input* $\mathsf{D}(\boldsymbol{z}) = \mathsf{DnCNN}^*(\boldsymbol{z})$. ***Galaxy DnCNN**** *is a further simplification of the Residual DnCNN to only 4 convolutional layers specifically designed for large-scale image recovery. In most experiments, we further constrain the Lipschitz constant (LC) of the direct denoiser to be LC = 1 and of the residual denoiser to LC = 2 by using spectral normalization [48]. LC = 1 means that* D *is a nonexpansive denoiser. A residual* $\mathsf{R} = \mathsf{I} - \mathsf{D}$ *with LC = 2 provides a necessary (but not sufficient) condition for* D *to be a nonexpansive denoiser.*

CNN denoisers via spectral normalization [47–49] and we have empirically tested the influence of nonexpansiveness to the quality of final image recovery.

Table 2 summarizes the SNR performance of BC-RED for two common variants of DnCNN*. The first variant is trained to learn the *direct* mapping from a noisy input to a clean image, while the second variant relies on *residual learning* to map its input to noise (shown in Fig. 4). To gain insight into the influence of the *Lipschitz constant (LC)* of a denoiser to its performance as a prior, we trained denoisers with both globally constrained and nonconstrained LCs via the spectral-normalization technique from [48]. For the direct network, we trained DnCNN* with LC $= 1$, which corresponds to a nonexpansive denoiser. For the residual network, we considered LC $= 2$, which is a necessary (but not sufficient) condition for the nonexpansiveness. In our simulations, BC-RED converged for all the variants of DnCNN*, except for the direct and unconstrained DnCNN*, which confirms that our theoretical analysis provides only sufficient conditions for convergence. Nonetheless, our simulations reveal the performance loss of the algorithm for the direct and nonexpansive (LC = 1) DnCNN*. On the other hand, the performance of the residual DnCNN* with LC $= 2$ nearly matches the performance of fully unconstrained networks in all experiments.

Table 2: *Average SNR achieved by BC-RED for two variants of DnCNN* at different Lipschitz constant (LC) values. Note how the stability of nonexpansive (LC = 1) direct DnCNN* comes with a suboptimal SNR performance. On the other hand, the excellent SNR performance of unconstrained direct DnCNN* comes with algorithmic instability. Finally, the residual DnCNN* with LC = 2 leads to both stable convergence and nearly SNR optimal results in all our simulations.*

Variants of DnCNN*		**Radon**		**Random**		**Fourier**	
		30 dB	**40 dB**	**30 dB**	**40 dB**	**30 dB**	**40 dB**
Direct	Unconstrained	21.67	24.74	Diverges	Diverges	29.40	30.35
	LC = 1	19.33	22.98	19.89	20.26	25.06	25.40
Residual	Unconstrained	20.88	24.68	26.49	27.60	29.39	30.31
	LC = 2	20.88	24.42	26.60	28.12	29.40	30.39

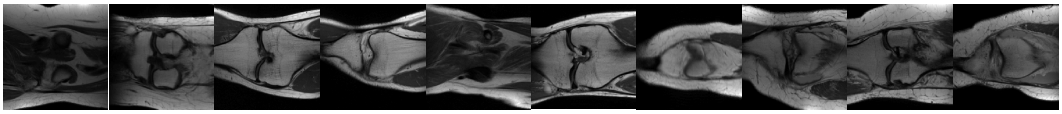

Figure 5: *Ten randomly selected test images from the fastMRI knee dataset [55].*

F Additional Numerical Validation

Fig. 5 shows ten randomly selected test images used for numerical validation. The simulations in this paper were performed on a machine equipped with an Intel Xeon Gold 6130 Processor that has 16 cores of 2.1 GHz and 192 GBs of DDR memory. We trained all neural nets using NVIDIA RTX 2080 GPUs.

Fig. 6 presents the convergence plots for *direct* and *residual* DnCNN* with Radon matrix. In order to ensure nonexpansivenss, the LC of direct DnCNN* is constrained to 1. On the other hand, the LC of the residual DnCNN* is constrained to 2, which is a necessary condition for ensuring its nonexpansiveness. We compare two variants of BC-RED, one with *i.i.d.* block selection and an alternative that proceeds in *epochs* of b consecutive iterations, where at the start of each epoch the set $\{1, \dots, b\}$ is reshuffled, and i_k is then selected consecutively from this ordered set. The figure first confirms our observation of the convergence of BC-RED under different DnCNN*, and further highlights the faster convergence speed of BC-RED due to its ability to select larger step-size and immediately reuse each block update. Among two block selection rules, *BC-RED (epoch)* clearly outperforms *BC-RED (i.i.d.)* in all our simulations, which has also been observed in traditional coordinate descent methods [25]. However, the theoretical understanding of this gap in performance between *epoch* and *i.i.d.* block selection remains elusive.

Fig. 7 visually compares the images recovered by BC-RED and RED and two baseline methods. First, the images visually illustrate the excellent agreement between BC-RED and RED. Second, leveraging advanced denoisers in BC-RED largely improves the reconstruction quality over PGM with the traditional TV prior. For instance, BC-RED under DnCNN* outperforms PGM under TV by 1 dB for Fourier matrix. Finally, we note the stability of BC-RED using the CNN denoiser versus the deteriorating performance of U-Net, which is trained end-to-end for Radon matrix with 30 dB noise. This fact highlights one key merit of the RED framework, that the CNN denoiser, only trained once, can be directly applied in different scenarios for different tasks with no degradation.

In BC-RED, the parameter τ controls the tradeoff between $\mathsf{zer}(\nabla g)$ and $\mathsf{fix}(\mathsf{D})$. Fig. 8 illustrates evolution of images reconstructed by BC-RED for different τ. The first row corresponds to the reconstruction from the Fourier measurements with 30 dB noise, while the second row corresponds to the Radon measurements with 40 dB noise. The figure clearly shows how τ explicitly adjusts the balance between the data-fit and the denoiser. In particular, small τ, corresponding to weak denoising, results in unwanted artifacts in the reconstructed images, while large τ promotes denoising strength but smooths out desired features and details. The leftmost images in Fig. 8 shows the optimal balance introduced by τ^*.

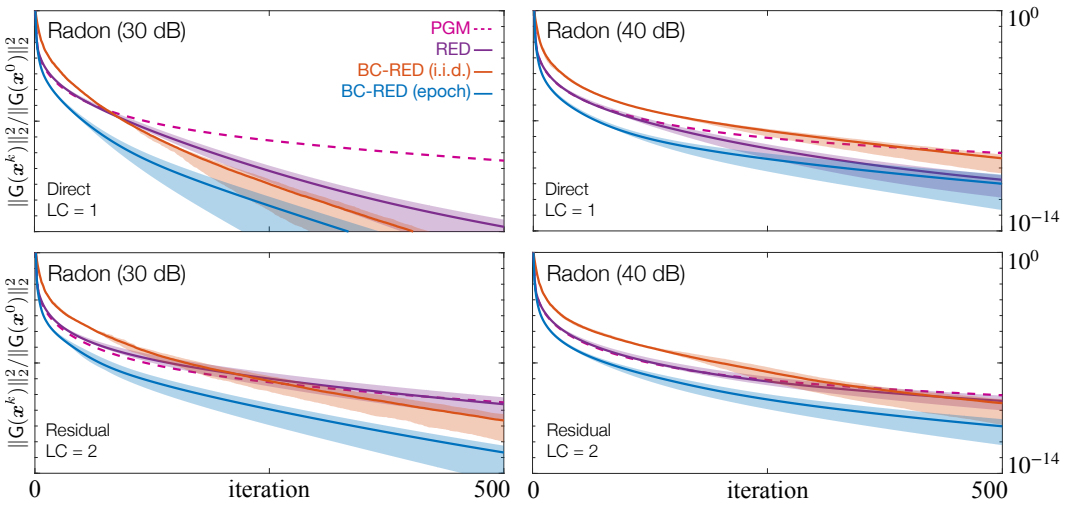

Figure 6: ***Left column*** *shows the convergence of BC-RED under different DnCNN* priors for Radon matrix with 30 dB noise. The top figure corresponds to the nonexpansive, direct DnCNN*, while the bottom figure corresponds to the residual DnCNN* with Lipschitz constant of two.* ***Right column*** *shows the convergence of BC-RED under the same set of DnCNN* priors for Radon matrix with 40 dB noise. Average normalized distance to* $\mathsf{zer}(\mathsf{G})$ *is plotted against the iteration number with the shaded area representing the range of values taken over all test images.* ***We observed general stability of BC-RED across all simulations for direct DnCNN* with LC = 1 and residual DnCNN* with LC = 2.***

To conclude, we present the experimental details of the galaxy image recovery task. In the simulation, we inherited the dataset used in [61]. The dataset[1] contains 10'000 galaxy survey images from the GREAT3 Challenge [64], and each image is cropped to 41×41 pixel size. The dataset also includes 597 simulated space variant point spread functions (PSF) corresponding to 597 physical position across 4 4096×4132 pixel CCDs [65, 66]. In order to synthesize the 8292×8364 pixel image, we first selected 597 galaxy images from the dataset and degraded each of them by a different PSF, and then locate the degraded images back to the corresponding positions in the full image. Note that we also contaminated each degraded image with AWGN of 5 dB. Figure 4 shows the architecture of the 4-layer DnCNN* used as denoiser for the galaxy image recovery. We generated 72000 training examples by rotating and flipping the rest 9000 images, and trained the neural network to learn the noise residual with LC$= 2$.

Since the locations of galaxies were known in this case, we optimized the speed of BC-RED by only updating the blocks containing galaxies. In practice, such block selection strategies can be efficiently implemented by applying a threshold on image intensities to separate blocks with galaxies from the ones that have only noise. As illustrated in Fig. 9, BC-RED converged to about 4.78×10^{-5}, in relative accuracy within 120 seconds, which corresponds to 100 iterations of the algorithm, with b BC-RED updates grouped as a single iteration. Fig. 10 illustrates the performance of BC-RED under DnCNN* for 4 example galaxies selected from the 1316×1245 pixel sub-image. The first row on the left shows the same galaxy in Fig. 3 in the main paper. We obtained the reconstructed image of the low-rank matrix prior by running the algorithm with default parameter values. This experiment demonstrates that BC-RED can indeed be applied to a realistic, nontrivial image recovery task on a large image.

[1] http://www.cosmostat.org/deconvolution

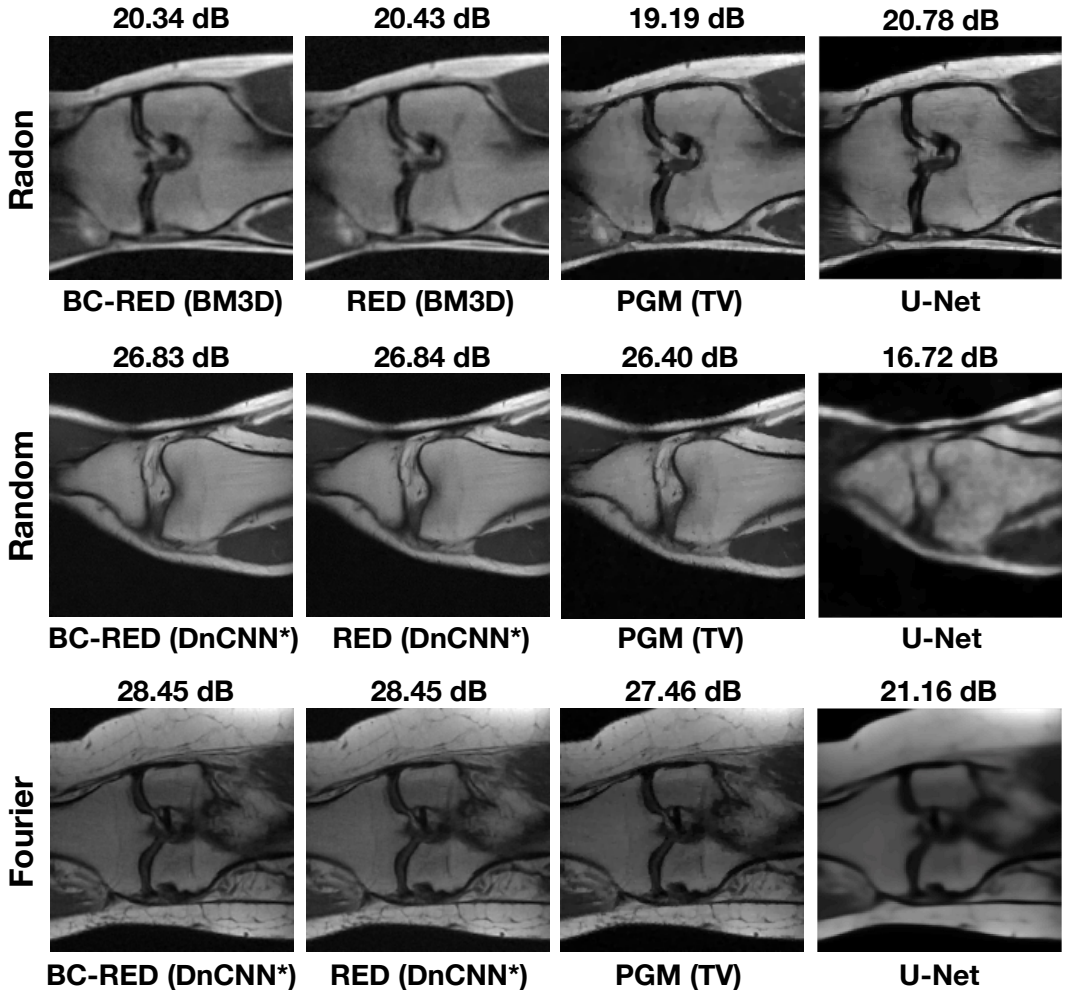

Figure 7: *Visual comparison between BC-RED and RED against PGM (TV) and U-Net for all three matrices with 30 dB noise. For BC-RED and RED, we selected the denoiser resulting in the best reconstruction performance. Every image is marked by its SNR value with respect to the ground truth. We highlight the excellent agreement between BC-RED and RED in all experiments. Note the strong degradation in the image quality for U-Net, due to the mismatch between the training and testing.*

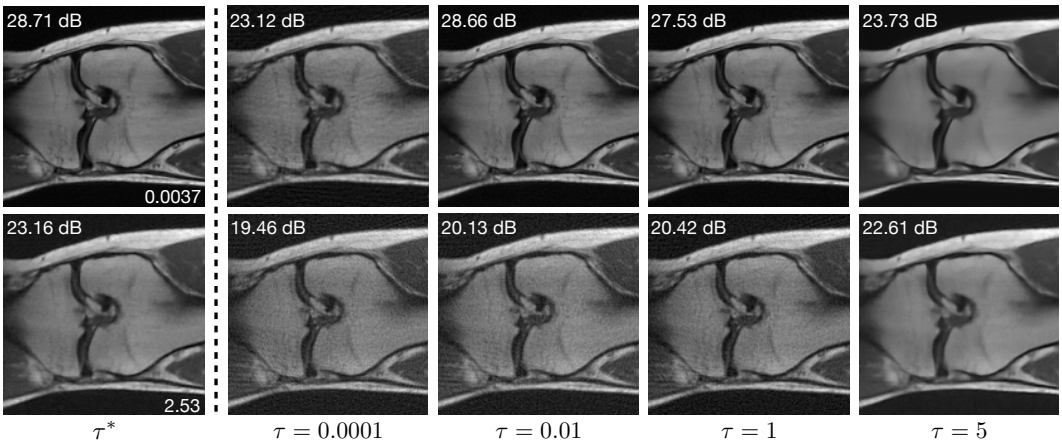

Figure 8: *Evolution of the images reconstructed by BC-RED using the DnCNN* denoiser for different values of τ. The first row corresponds to Fourier matrix with 30 dB noise, while the second row corresponds to the Radon matrix with 40 dB noise. Each reconstructed image is marked with its SNR value with respect to the ground truth image. The optimal parameters τ^* for the two problems are 0.0037 and 2.35, respectively. The denoiser used in this simulation is the residual DnCNN* with a Lipschitz constant LC = 2. This figure illustrates how τ enables an explicit tradeoff between the data-fit and the regularization.*

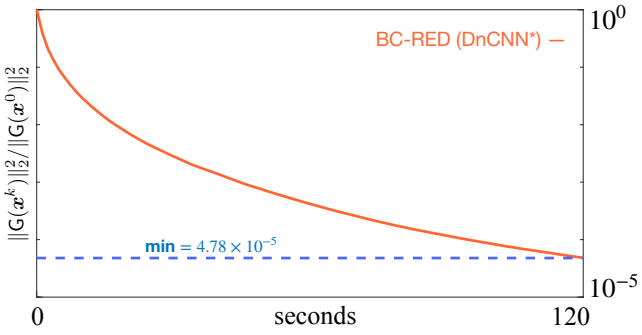

Figure 9: *Illustration of the convergence of BC-RED under DnCNN* in the realistic, large-scale image recovery task. BC-RED is run for 100 iterations, which leads to the accuracy of 4.78×10^{-5} within 120 seconds. The efficiency of the algorithm is due to the sparsity of the recovery problem.*

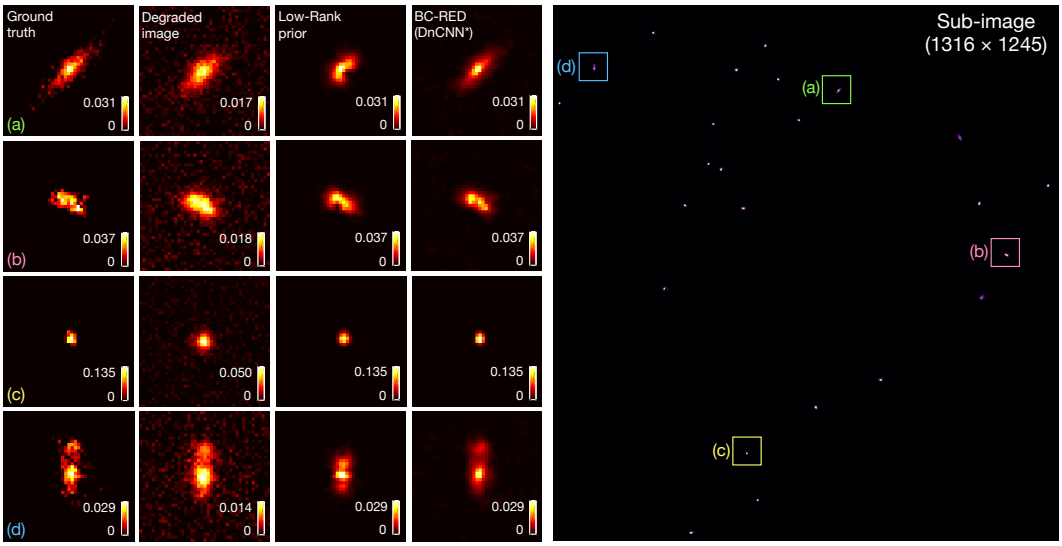

Figure 10: *Illustration of performance of BC-RED under residual DnCNN* denoiser with LC = 2. The first and the second columns show the ground truth images and the blocks from the measurement, respectively. The third and the forth columns are the reconstructed results obtained by BC-RED and the low-rank matrix prior [61], respectively. The rightmost image is a* 1316×1245 *pixel sub-image of the full-sized* 8292×8364 *pixel reconstructed image obtained by BC-RED. Note that the intent of this figure is not to justify DnCNN* as a prior for image recovery, but to demonstrate that BC-RED can indeed be applied to a realistic, nontrivial image recovery task on a large image.*