Scalable Plug-and-Play ADMM With Convergence Guarantees

Yu Sun, *Graduate Student Member, IEEE*, Zihui Wu, Xiaojian Xu, *Graduate Student Member, IEEE*, Brendt Wohlberg, *Senior Member, IEEE*, and Ulugbek S. Kamilov, *Senior Member, IEEE*

***Abstract*—Plug-and-play priors (PnP) is a broadly applicable methodology for solving inverse problems by exploiting statistical priors specified as denoisers. Recent work has reported the state-of-the-art performance of PnP algorithms using pre-trained deep neural nets as denoisers in a number of imaging applications. However, current PnP algorithms are impractical in large-scale settings due to their heavy computational and memory requirements. This work addresses this issue by proposing an *incremental* variant of the widely used PnP-ADMM algorithm, making it scalable to problems involving a large number measurements. We theoretically analyze the convergence of the algorithm under a set of explicit assumptions, extending recent theoretical results in the area. Additionally, we show the effectiveness of our algorithm with nonsmooth data-fidelity terms and deep neural net priors, its fast convergence compared to existing PnP algorithms, and its scalability in terms of speed and memory.**

***Index Terms*—Regularized image reconstruction, plug-and-play priors, deep learning, regularization parameter.**

I. Introduction

PLUG-AND-PLAY priors (PnP) is a simple yet flexible methodology for imposing statistical priors without explicitly forming an objective function [1], [2]. PnP algorithms alternate between imposing data consistency by minimizing a data-fidelity term and imposing a statistical prior by applying an additive white Gaussian noise (AWGN) denoiser. PnP draws its inspiration from the *proximal algorithms* extensively used in nonsmooth composite optimization [3], such as the proximal-gradient method (PGM) [4]–[7] and alternating direction method of multipliers (ADMM) [8]–[11]. The popularity of deep learning has led to a wide adoption of PnP for exploiting *learned* priors specified through pre-trained deep neural nets, leading to its state-of-the-art performance in a variety of applications [12]–[16]. Its empirical success has spurred a follow-up work that provided theoretical justifications to PnP in various settings [17]–[25]. Despite this progress, the computation and memory requirements of current PnP algorithms makes them impractical in problems with a large number of measurements. To the best of our knowledge, the only prior work on developing PnP algorithms for processing large-scale measurements is the *stochastic gradient descent variant of PnP (PnP-SGD)*, whose fixed-point convergence was recently analyzed for smooth data-fidelity terms [20].

In this work, we present a new *incremental PnP-ADMM (IPA)* algorithm for dealing with large-scale measurements. As an extensions of the widely used PnP-ADMM [1], [2], IPA can integrate statistical information from a data-fidelity term and a pre-trained deep neural net. However, unlike PnP-ADMM, IPA can effectively scale to datasets that are too large for traditional batch processing by using a single element or a small subset of the dataset at a time. The memory and per-iteration complexity of IPA is independent of the number of measurements, thus allowing it to deal with very large datasets. Additionally, unlike PnP-SGD [20], IPA can effectively address problems with *nonsmooth* data-fidelity terms, and generally has faster convergence. We present a detailed convergence analysis of IPA under a set of explicit assumptions on the data-fidelity term and the denoiser. Our analysis extends the recent fixed-point analysis of PnP-ADMM in [23] to partial randomized processing of data. To the best of our knowledge, the proposed scalable PnP algorithm and corresponding convergence analysis are absent from the current literature in this area. Our numerical validation demonstrates the practical effectiveness of IPA for integrating nonsmooth data-fidelity terms and deep neural net priors, its fast convergence compared to PnP-SGD, and its scalability in terms of both speed and memory. In summary, we establish IPA as a flexible, scalable, and theoretically sound PnP algorithm applicable to a wide variety of large-scale problems.

II. Background

Consider the problem of estimating an unknown vector $\boldsymbol{x} \in \mathbb{R}^n$ from a set of noisy measurements $\boldsymbol{y} \in \mathbb{R}^m$. It is standard

Manuscript received January 14, 2021; revised April 27, 2021 and June 27, 2021; accepted June 27, 2021. Date of publication July 2, 2021; date of current version August 14, 2021. This work supported in part by the National Science Foundation award CCF-1813910, by the the National Science Foundation CAREER award under Grant CCF-2043134, and by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20200061DR. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stanley H. Chan. *(Yu Sun, Zihui Wu, and Xiaojian Xu contributed equally to this work.) (Corresponding author: Ulugbek S. Kamilov.)*

Yu Sun and Xiaojian Xu are with the Department of Computer Science and Enginnering, Washington University in St. Louis, St. Louis, MO 63130 USA (e-mail: sun.yu@wustl.edu; xiaojianxu@wustl.edu).

Zihui Wu is with the Department of Computer Science, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: ray.wu@wustl.edu).

Brendt Wohlberg is with Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545 USA (e-mail: brendt@ieee.org).

Ulugbek S. Kamilov is with the Department of Computer Science and Engineering and the Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA (e-mail: kamilov@ieee.org).

This article has supplementary downloadable material available at https://doi.org/10.1109/TCI.2021.3094062, provided by the authors.

Digital Object Identifier 10.1109/TCI.2021.3094062

practice to formulate the solution as an optimization

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

where g is a data-fidelity term that quantifies consistency with the observed data $\boldsymbol{y}$ and h is a regularizer that encodes prior knowledge on $\boldsymbol{x}$. As an example, consider the nonsmooth ℓ_1-norm data-fidelity term $g(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_1$, which assumes a linear observation model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$, and the TV regularizer $h(\boldsymbol{x}) = \tau\|\boldsymbol{D}\boldsymbol{x}\|_1$, where $\boldsymbol{D}$ is the gradient operator and $\tau > 0$ is the regularization parameter. Common applications of (1) include sparse vector recovery in compressive sensing [26], [27], image restoration using total variation (TV) [28], and low-rank matrix completion [29].

Proximal algorithms are often used for solving problems of form (1) when g or h are nonsmooth [3]. For example, one such standard algorithm, ADMM, can be summarized as

$$\boldsymbol{z}^k = \text{prox}_{\gamma g}(\boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1}) \tag{2a}$$

$$\boldsymbol{x}^k = \text{prox}_{\gamma h}(\boldsymbol{z}^k - \boldsymbol{s}^{k-1}) \tag{2b}$$

$$\boldsymbol{s}^k = \boldsymbol{s}^{k-1} + \boldsymbol{x}^k - \boldsymbol{z}^k \,, \tag{2c}$$

where $\gamma > 0$ is the penalty parameter [11] and *proximal operator* is defined as

$$\text{prox}_{\tau h}(\boldsymbol{z}) := \underset{\boldsymbol{x}\in\mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \tau h(\boldsymbol{x}) \right\} \tag{3}$$

for any proper, closed, and convex function h [3]. The proximal operator can be interpreted as a *maximum a posteriori probability (MAP)* estimator for the AWGN denoising problem

$$\boldsymbol{z} = \boldsymbol{x}_0 + \boldsymbol{n} \quad \text{where} \quad \boldsymbol{x}_0 \sim p_{\boldsymbol{x}_0}, \quad \boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \tau\mathbf{I}) \,, \tag{4}$$

by setting $h(\boldsymbol{x}) = -\log(p_{\boldsymbol{x}_0}(\boldsymbol{x}))$. This perspective inspired the development of PnP [1], [2], where the proximal operator is simply replaced by a more general denoiser $\mathsf{D} : \mathbb{R}^n \to \mathbb{R}^n$ such as BM3D [30] or DnCNN [31]. For example, the widely used PnP-ADMM can be summarized as

$$\boldsymbol{z}^k = \text{prox}_{\gamma g}(\boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1}) \tag{5a}$$

$$\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k - \boldsymbol{s}^{k-1}) \tag{5b}$$

$$\boldsymbol{s}^k = \boldsymbol{s}^{k-1} + \boldsymbol{x}^k - \boldsymbol{z}^k \,, \tag{5c}$$

where, in analogy with $\tau > 0$ in (3), we introduce the parameter $\sigma > 0$ controlling the relative strength of the denoiser. Remarkably, this heuristic of using denoisers not associated with any h within an iterative algorithm exhibited great empirical success [12]–[15], [25] and spurred a great deal of theoretical work on PnP algorithms [17]–[24].

A elegant fixed-point convergence analysis of PnP-ADMM was presented in [23]. By substituting $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ into PnP-ADMM, the algorithm is expressed in terms of an operator

$$\mathsf{P} := \frac{1}{2}\mathsf{I} + \frac{1}{2}(2\mathsf{G} - \mathsf{I})(2\mathsf{D}_\sigma - \mathsf{I}) \quad \text{with}$$

$$\mathsf{G} := \text{prox}_{\gamma g} \,, \tag{6}$$

where I denotes the identity operator. The convergence of PnP-ADMM is then established through its equivalence to the fixed-point convergence of the sequence $\boldsymbol{v}^k = \mathsf{P}(\boldsymbol{v}^{k-1})$. The equivalence of PnP-ADMM to the iterations of the operator (6) originates from the well-known relationship between ADMM and the Douglas-Rachford splitting [3], [8], [19], [23].

Scalable optimization algorithms have become increasingly important in the context of large-scale problems arising in machine learning and data science [32]. Stochastic and online optimization techniques have been investigated for traditional ADMM [33]–[37], where $\text{prox}_{\gamma g}$ is approximated using a subset of observations (with or without subsequent linearization). Our work contributes to this area by investigating the scalability of PnP-ADMM that is *not* minimizing any explicit objective function. Since PnP-ADMM can integrate powerful deep neural net denoisers, there is a need to understand its theoretical properties and ability to process a large number of measurements.

Before introducing our algorithm, it is worth briefly mentioning an emerging paradigm of using deep neural nets for solving ill-posed imaging inverse problems (see, reviews [38]–[41]). This work is most related to techniques that explicitly decouple the measurement model from the learned prior. For example, learned denoisers have been adopted for a class of algorithms in compressive sensing known as *approximate message passing (AMP)* [42]–[45]. The key difference of PnP from AMP is that it does not assume random measurement operators. *Regularization by denoising (RED)* is a closely related method that specifies an explicit regularizers that has a simple gradient [46]–[48]. PnP does not seek the existence of such an objective, instead interpreting solutions as equilibrum points balancing the data-fit and the prior [19]. By focusing on the partial processing of $\boldsymbol{y}$, this work is complementary to the recent approaches that perform block-coordinate processing of $\boldsymbol{x}$ either in the context of deep unrolling [49], [50] or RED [51], [52]. Finally, a recent line of work has investigated the recovery and convergence guarantees for priors specified by *generative adversarial networks (GANs)* [53]–[57]. PnP does not seek to project its iterates to the range of a GAN, instead it directly uses the output of a simple AWGN denoiser to improve the estimation quality. This simplifies the training and application of learned priors within the PnP methodology. Our work contributes to this broad area by providing new conceptual, theoretical, and empirical insights into incremental ADMM optimization under statistical priors specified as deep neural net denoisers.

III. Incremental PnP-ADMM

Batch PnP algorithms operate on the whole observation vector $\boldsymbol{y} \in \mathbb{R}^m$. We are interested in partial randomized processing of observations by considering the decomposition of $\mathbb{R}^m$ into $b \geq 1$ blocks

$$\mathbb{R}^m = \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \cdots \times \mathbb{R}^{m_b} \quad \text{with}$$

$$m = m_1 + m_2 + \cdots + m_b \,.$$

We thus consider data-fidelity terms of the form

$$g(\boldsymbol{x}) = \frac{1}{b}\sum_{i=1}^{b} g_i(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^n \,, \tag{7}$$

Algorithm 1: Incremental Plug-and-Play ADMM (IPA)

input: initial values $\boldsymbol{x}^0, \boldsymbol{s}^0 \in \mathbb{R}^n$, parameters $\gamma, \sigma > 0$.
for $k = 1, 2, 3, \ldots$ **do**
  Choose an index $i_k \in \{1, \ldots, b\}$
  $\boldsymbol{z}^k \leftarrow \mathsf{G}_{i_k}(\boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1})$ where $\mathsf{G}_{i_k} := \operatorname{prox}_{\gamma g_{i_k}}$
  $\boldsymbol{x}^k \leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k - \boldsymbol{s}^{k-1})$
  $\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + \boldsymbol{x}^k - \boldsymbol{z}^k$
end for

where each g_i is evaluated only on the subset $\boldsymbol{y}_i \in \mathbb{R}^{m_i}$ of the full data $\boldsymbol{y}$.

The proposed IPA algorithm seeks to avoid the direct computation of $\operatorname{prox}_{\gamma g}$ in PnP-ADMM. As shown in Algorithm 1, it extends stochastic variants of traditional ADMM [33]–[37] by integrating denoisers D_σ that are *not* associated with any h. Its per-iteration complexity is independent of the number of data blocks b, since it processes only a single component function g_i at every iteration.

It is important to note that in some applications [58]–[60], the $\operatorname{prox}_{\gamma g}$ step of PnP-ADMM can be efficiently evaluated by leveraging the structure of the measurement operator (such as diagonalization by Fourier transform). Nonetheless, IPA provides flexibility for controlling the number of measurements $1 \leq m_i \leq m$ used in every iteration, which makes it a useful alternative to PnP-ADMM, when the memory/computational benefits for evaluating $\operatorname{prox}_{\gamma g_i}$ (which uses only $\boldsymbol{y}_i \in \mathbb{R}^{m_i}$ and $\boldsymbol{A}_i \in \mathbb{R}^{m_i \times n}$) outweigh those of $\operatorname{prox}_{\gamma g}$ (which uses $\boldsymbol{y} \in \mathbb{R}^m$ and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$).

In principle, IPA can be implemented using different block selection rules. The strategy adopted for our theoretical analysis focuses on the usual strategy of selecting indices i_k as independent and identically distributed (i.i.d.) random variables distributed uniformly over $\{1, \ldots, b\}$. An alternative would be to proceed in epochs of b consecutive iterations, where at the start of each epoch the set $\{1, \ldots, b\}$ is reshuffled, and i_k is selected from this ordered set [61]. In some applications, it might also be beneficial to select indices i_k in an online data-adaptive fashion by taking into account the statistical relationships among observations [62], [63].

Unlike PnP-SGD, IPA does not require smoothness of the functions g_i. Instead of computing the partial gradient ∇g_i, as is done in PnP-SGD, IPA evaluates the partial proximal operator G_i. Nonsmooth data-fidelity terms have been extensively used in many applications, including wavelet inpainting, tensor factorization, feature selection, dictionary learning, and phase unwrapping [64]–[70]. The maximal benefit of IPA over PnP-SGD is expected for problems in which G_i is efficient to evaluate. This is a case for a number of functions commonly used in many applications (see the extensive discussion on proximal operators in [71]). For example, the proximal operator of the ℓ_2-norm data-fidelity term $g_i(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{A}_i\boldsymbol{x}\|_2^2$ has a closed-form solution

$$\mathsf{G}_i(\boldsymbol{z}) = \operatorname{prox}_{\gamma g_i}(\boldsymbol{z}) = \left(\mathbf{I} + \gamma \boldsymbol{A}_i^\mathsf{T}\boldsymbol{A}_i\right)^{-1}\left(\boldsymbol{z} + \gamma \boldsymbol{A}_i^\mathsf{T}\boldsymbol{y}_i\right) \quad (8)$$

Algorithm 2: Minibatch IPA

input: initial values $\boldsymbol{x}^0, \boldsymbol{s}^0 \in \mathbb{R}^n$, parameters $\gamma, \sigma > 0$, minibatch size $p \geq 1$.
for $k = 1, 2, 3, \ldots$ **do**
  Choose indices $i_1, \ldots, i_p$ from the set $\{1, \ldots, b\}$.
  $\boldsymbol{z}^k \leftarrow \widehat{\mathsf{G}}(\boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1})$ where $\widehat{\mathsf{G}} := \frac{1}{p}\sum_{j=1}^p \operatorname{prox}_{\gamma g_{i_j}}$
  $\boldsymbol{x}^k \leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k - \boldsymbol{s}^{k-1})$
  $\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + \boldsymbol{x}^k - \boldsymbol{z}^k$
end for

for $\gamma > 0$ and $\boldsymbol{z} \in \mathbb{R}^n$. Prior work has extensively discussed efficient strategies for evaluating (8) for a variety of linear operators, including convolutions, partial Fourier transforms, and subsampling masks [9], [58]–[60]. As a second example, consider the ℓ_1-data fidelity term $g_i(\boldsymbol{x}) = \|\boldsymbol{y}_i - \boldsymbol{A}_i\boldsymbol{x}\|_1$, which is nonsmooth. The corresponding proximal operator has a closed form solution for any orthogonal operator $\boldsymbol{A}_i$ and can also be efficiently computed in many other settings [71].

IPA can also be implemented as a *minibatch* algorithm, processing several blocks in parallel at every iteration, thus improving its efficiency on multi-processor hardware architectures. Algorithm 2 presents the minibatch version of IPA that averages several proximal operators evaluated over different data blocks. When the minibatch size $p = 1$, Algorithm 2 reverts to Algorithm 1. The main benefit of minibatch IPA is its suitability for parallel computation of $\widehat{\mathsf{G}}$, which can take advantage of multi-processor architectures.

Minibatch IPA is related to the *proximal average* approximation of $\mathsf{G} = \operatorname{prox}_{\gamma g}$ [72], [73]

$$\overline{\mathsf{G}}(\boldsymbol{x}) = \frac{1}{b}\sum_{i=1}^b \operatorname{prox}_{\gamma g_i}(\boldsymbol{x}) \quad \boldsymbol{x} \in \mathbb{R}^n \ .$$

When Assumption 1, introduced in Section IV, is satisfied, then the approximation error is bounded for any $\boldsymbol{x} \in \mathbb{R}^n$ as

$$\|\mathsf{G}(\boldsymbol{x}) - \overline{\mathsf{G}}(\boldsymbol{x})\| \leq 2\gamma L \ .$$

Minibatch IPA thus simply uses a minibatch approximation $\widehat{\mathsf{G}}$ of the proximal average $\overline{\mathsf{G}}$. One implication of this is that even when the minibatch is *exactly* equal to the full measurement vector, minibatch IPA is not exact due to the approximation error introduced by the proximal average. However, the resulting approximation error can be made as small as desired by controlling the penalty parameter $\gamma > 0$.

IV. Theoretical Analysis

We now present a theoretical analysis of IPA. We fist present an intuitive interpretation of its solutions, and then present our convergence analysis under a set of explicit assumptions.

A. Fixed Point Interpretation

PnP cannot be interpreted using the standard tools from convex optimization, since its solution is generally not a minimizer

of an objective function. Nonetheless, we develop an intuitive operator based interpretation.

Consider the following set-valued operator

$$\mathsf{T} := \gamma \partial g + (\mathsf{D}_\sigma^{-1} - \mathsf{I}) \quad \gamma > 0 \,, \tag{9}$$

where ∂g is the subdifferential of the data-fidelity term and $\mathsf{D}_\sigma^{-1}(\boldsymbol{x}) := \{\boldsymbol{z} \in \mathbb{R}^n : \boldsymbol{x} = \mathsf{D}_\sigma(\boldsymbol{z})\}$ is the inverse operator of the denoiser D_σ. The details for obtaining (9) from (2) are provided in Appendix C.1. Note that this inverse operator exists even when D_σ is not one-to-one [8], [74]. By characterizing the fixed points of PnP algorithms, it can be shown that their solutions can be interpreted as vectors in the zero set of T

$$\begin{aligned} &\mathbf{0} \in \mathsf{T}(\boldsymbol{x}^*) = \gamma \partial g(\boldsymbol{x}^*) + (\mathsf{D}_\sigma^{-1}(\boldsymbol{x}^*) - \boldsymbol{x}^*) \\ &\Leftrightarrow \quad \boldsymbol{x}^* \in \operatorname{zer}(\mathsf{T}) := \{\boldsymbol{x} \in \mathbb{R}^n : \mathbf{0} \in \mathsf{T}(\boldsymbol{x})\} \,. \end{aligned}$$

Consider the following two sets

$$\begin{aligned} \operatorname{zer}(\partial g) &:= \{\boldsymbol{x} \in \mathbb{R}^n : \mathbf{0} \in \partial g(\boldsymbol{x})\} \quad \text{and} \\ \operatorname{fix}(\mathsf{D}_\sigma) &:= \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x} = \mathsf{D}_\sigma(\boldsymbol{x})\} \,, \end{aligned}$$

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

If $\boldsymbol{x}^* \in \operatorname{zer}(\partial g) \cap \operatorname{fix}(\mathsf{D}_\sigma)$, then $\boldsymbol{x}^* \in \operatorname{zer}(\mathsf{T})$, which implies that $\boldsymbol{x}^*$ is one of the solutions. Hence, any vector that minimizes a convex data-fidelity term g and noiseless according to D_σ is in the solution set. On the other hand, when $\operatorname{zer}(\partial g) \cap \operatorname{fix}(\mathsf{D}_\sigma) = \varnothing$, then $\boldsymbol{x}^* \in \operatorname{zer}(\mathsf{T})$ corresponds to an equilibrium point between two sets.

This interpretation of PnP highlights one important aspect that is often overlooked in the literature, namely that, unlike in the traditional formulation (1), the regularization in PnP depends on both the denoiser parameter $\sigma > 0$ and the penalty parameter $\gamma > 0$, with both influencing the solution. Hence, the best performance is obtained by jointly tuning both parameters for a given experimental setting. In the special case of $\mathsf{D}_\sigma = \operatorname{prox}_{\gamma h}$ with $\gamma = \sigma^2$, we have

$$\begin{aligned} \operatorname{fix}(\mathsf{D}_\sigma) &= \{\boldsymbol{x} \in \mathbb{R}^n : \mathbf{0} \in \partial h(\boldsymbol{x})\} \quad \text{and} \\ \operatorname{zer}(\mathsf{T}) &:= \{\boldsymbol{x} \in \mathbb{R}^n : \mathbf{0} \in \partial g(\boldsymbol{x}) + \partial h(\boldsymbol{x})\} \,, \end{aligned}$$

which corresponds to the optimization formulation (1) whose solutions are independent of γ.

B. Convergence Analysis

Our analysis requires three assumptions that jointly serve as sufficient conditions.

Assumption 1: Each g_i is proper, closed, convex, and Lipschitz continuous with constant $L_i > 0$. We define the largest Lipschitz constant as $L = \max\{L_1, \ldots, L_b\}$.

This assumption is commonly adopted in nonsmooth optimization and is equivalent to existence of a global upper bound on subgradients [34], [73], [75]. It is satisfied by a large number of functions, such as the ℓ_1-norm. The ℓ_2-norm also satisfies Assumption 1 when it is evaluated over a bounded subset of $\mathbb{R}^n$. We next state our assumption on D_σ.

Assumption 2: The residual $\mathsf{R}_\sigma := \mathsf{I} - \mathsf{D}_\sigma$ of the denoiser D_σ is firmly nonexpansive.

We review firm nonexpansiveness and other related concepts in the Appendix C. Firmly nonexpansive operators are a subset of *nonexpansive* operators (those that are Lipschitz continuous with constant one). A simple strategy to obtain a firmly nonexpansive operator is to create a $(1/2)$-averaged operator from a nonexpansive operator [3]. The residual R_σ is firmly nonexpansive *if and only if* D_σ is firmly nonexpansive. It is worth noting that (a) any explicit or implicit proximal operator is firmly nonexpansive, and (b) any symmetric matrix with eigenvalues in $[0, 1]$ is firmly nonexpansive. This implies that many recently designed denoisers for PnP, such as those discussed in [2], [22], [76]–[78] automatically satisfy Assumption 2.

The rationale for stating Assumption 2 for R_σ is based on our interest in *residual* deep neural nets. The success of residual learning in the context of image restoration is well known [31]. Prior work has also shown that Lipschitz constrained residual networks yield excellent performance without sacrificing stable convergence [23], [51]. Additionally, there has recently been an explosion of techniques for training Lipschitz constrained and firmly nonexpansive deep neural nets [23], [79]–[81].

Assumption 3: The operator T in (9) is such that $\operatorname{zer}(\mathsf{T}) \neq \varnothing$. There also exists $R < \infty$ such that

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2 \leq R \quad \text{for all} \quad \boldsymbol{x}^* \in \operatorname{zer}(\mathsf{T}) \,.$$

The first part of the assumptions simply ensures the existence of a solution. The existence of the bound R often holds in practice, as many denoisers have bounded range spaces. In particular, this is true for a number of image denoisers whose outputs live within the bounded subset $[0255]^n \subset \mathbb{R}^n$.

We will state our convergence results in terms of the operator $\mathsf{S} : \mathbb{R}^n \to \mathbb{R}^n$ defined as

$$\mathsf{S} := \mathsf{D}_\sigma - \mathsf{G}(2\mathsf{D}_\sigma - \mathsf{I}) \,. \tag{10}$$

Both IPA and PnP-ADMM can be interpreted as algorithms for computing an element in $\operatorname{zer}(\mathsf{S})$, which is equivalent to finding an element of $\operatorname{zer}(\mathsf{T})$ (see details in Appendix C).

We are now ready to state our main result on IPA.

Theorem 1: Run IPA for $t \geq 1$ iterations with random i.i.d. block selection under Assumptions 1-3 using a penalty parameter $\gamma > 0$. Then, the sequence $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ satisfies

$$\mathbb{E}\left[\frac{1}{t}\sum_{k=1}^{t} \|\mathsf{S}(\boldsymbol{v}^k)\|_2^2\right] \leq \frac{(R + 2\gamma L)^2}{t} + \max\{\gamma, \gamma^2\} C \,, \tag{11}$$

where $C := 4LR + 12L^2$ is a positive constant.

In order to contextualize this result, we also review the convergence of the traditional PnP-ADMM.

Theorem 2: Run PnP-ADMM for $t \geq 1$ iterations under Assumptions 1-3 using a penalty parameter $\gamma > 0$. Then, the

sequence $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ satisfies

$$\frac{1}{t}\sum_{k=1}^{t} \|\mathsf{S}(\boldsymbol{v}^k)\|_2^2 \leq \frac{(R+2\gamma L)^2}{t} \,. \tag{12}$$

Both proofs are provided in the Appendix A. The proof of Theorem 2 is a modification of the analysis in [23], obtained by relaxing the *strong convexity* assumption in [23] by Assumption 1 and replacing the assumption that R_σ is a *contraction* in [23] by Assumption 2. Theorem 2 establishes that the iterates of PnP-ADMM satisfy $\|\mathsf{S}(\boldsymbol{v}^t)\| \to 0$ as $t \to \infty$. Since S is firmly nonexpansive (see Appendix C.3) and D_σ is nonexpansive, the Krasnosel'skii-Mann theorem (see Section 5.2 in [82]) directly implies that $\boldsymbol{v}^t \to \mathrm{zer}(\mathsf{S})$ and $\boldsymbol{x}^t = \mathsf{D}_\sigma(\boldsymbol{v}^t) \to \mathrm{zer}(\mathsf{T})$.

Theorem 1 establishes that *in expectation*, IPA has a similar convergence behavior to PnP-ADMM up to an error term that depends on the penalty parameter γ. One can precisely control the accuracy of IPA by setting γ to a desired level. In practice, γ can be treated as a hyperparameter and tuned to maximize performance for a suitable image quality metric, such as SNR or SSIM. Our numerical results in Section V corroborate that excellent SNR performance of IPA can be achieved without taking $\|\mathsf{S}(\boldsymbol{v}^t)\|_2$ to zero, which simplifies practical applicability of IPA. (Note that the convergence analysis for IPA in Theorem 1 can be easily extended to minibatch IPA with a straightforward extension of Lemma 1 in Appendix A.2 to several indices, and by following the steps of the main proof in Appendix A.1.)

Finally, note that the convergence of the IPA iterates can also be analyzed under assumptions adopted in [23], namely that g_i are strongly convex and R_σ is a contraction. Such an analysis leads to the statement

$$\mathbb{E}\left[\|\boldsymbol{x}^t - \boldsymbol{x}^*\|_2\right] \leq \eta^t(2R + 4\gamma L) + (4\gamma L)/(1-\eta) \,, \tag{13}$$

where $0 < \eta < 1$. Equation (13) establishes a linear convergence to $\mathrm{zer}(\mathsf{T})$ up to an error term. A proof of (13) is provided in the Appendix B. As corroborated by our simulations in Section V, the actual convergence of IPA holds even more broadly than suggested by both sets of sufficient conditions. This suggests a possibility of future analysis of IPA under more relaxed assumptions.

V. Numerical Validation

Recent work has shown the excellent performance of PnP for smooth data-fidelity terms using advanced denoising priors. Our goal in this section is to extend these studies with simulations validating the effectiveness of IPA for nonsmooth data-fidelity terms and deep neural net priors, as well as demonstrating its scalability to large-scale inverse problems. We consider two applications of the form $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$, where $\boldsymbol{e} \in \mathbb{R}^m$ denotes the noise and $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ denotes either a random Gaussian matrix in *compressive sensing (CS)* or the transfer function in *intensity diffraction tomography (IDT)* [83].

Our deep neural net prior is based on the DnCNN architecture [31], with its batch normalization layers removed for controlling the Lipschitz constant of the network via spectral normalization [84] (see details in Appendix F.1). We train a nonexpansive residual network R_σ by predicting the noise residual from its noisy input. While this means that R_σ is not trained to be firmly nonexpansive, we observed that nonexpansiveness was sufficient for empirical convergence. Note also that a nonexpansive R_σ satisfies the necessary (but not sufficient) condition for firm nonexpansiveness of D_σ. It is also worth mentioning that denoiser design, which is not our main focus, is an active area of research in the context of PnP. The training data is generated by adding AWGN to the BSD400 images [85]. The reconstruction quality is quantified using the signal-to-noise ratio (SNR) in dB. We pre-train several deep neural net models as denoisers for $\sigma \in [1, 10]$, using σ intervals of 0.5, and use the denoiser achieving the best SNR.

A. Integration of Nonsmooth Data-Fidelity Terms and Pretrained Deep Priors

We first test IPA on non-smooth data-fidelity terms. The matrix $\boldsymbol{A}$ is generated with i.i.d. zero-mean Gaussian random elements of variance $1/m$, and $\boldsymbol{e}$ as a sparse Bernoulli-Gaussian vector with the sparsity ratio of 0.1. This means that, in expectation, ten percent of the elements of $\boldsymbol{y}$ are contaminated by AWGN. The sparse nature of the noise motivates the usage of the ℓ_1-norm $g(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_1$, since it is less sensitive to extreme values. The nonsmoothness of ℓ_1-norm prevents the usage of gradient-based algorithms such as PnP-SGD. On the other hand, the application IPA is facilitated by efficient strategies for computing the proximal operator [28], [86].

Note that the focus of this section is on using CS as a convenient application for demonstrating some of the key properties of IPA, and is not on achieving the state-of-the-art subsampling in CS [44], [87]–[90]. For any subsampling rate, the reconstruction quality of IPA is expected to match that of PnP-ADMM, which has been extensively studied in prior work. In particular, a recent work [91] has extensively compared the recovery performance of PnP relative to several widely-used algorithms in CS.

We set the measurement ratio to be approximately $m/n = 0.7$ with AWGN of standard deviation 5. Twelve standard images from *Set12* [31] are used in testing, each resized to 64×64 pixels for rapid parameter tuning and testing. We quantify the convergence accuracy using the normalized distance $\|\mathsf{S}(\boldsymbol{v}^k)\|_2^2/\|\boldsymbol{v}^k\|_2^2$, which is expected to approach zero as IPA converges to a fixed point.

Theorem 1 characterizes the convergence of IPA in terms of $\|\mathsf{S}(\boldsymbol{v}^k)\|_2$ up to a constant error term that depends on γ. This is illustrated in Fig. 1 for three values of the penalty parameter $\gamma \in \{\gamma_0, \gamma_0/2, \gamma_0/4\}$ with $\gamma_0 = 0.02$. The average normalized distance $\|\mathsf{S}(\boldsymbol{v}^k)\|_2^2/\|\boldsymbol{v}^k\|_2^2$ and SNR are plotted against the iteration number and labeled with their respective final values. The shaded areas represent the range of values attained across all test images. IPA is implemented to use a random half of the elements in $\boldsymbol{y}$ in every iteration to impose the data-consistency. Fig. 1 shows the improved convergence of IPA to $\mathrm{zer}(\mathsf{S})$ for smaller values of γ, which is consistent with our theoretical analysis. Specifically, the final accuracy improves approximately $3\times$ (from 1.07×10^{-5} to 3.59×10^{-6}) when γ is reduced from γ_0 to $\gamma_0/4$. On the other hand, the SNR values are nearly identical for all three experiments, indicating that in

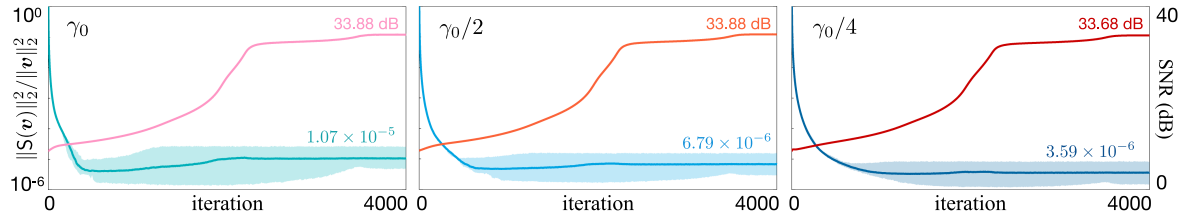


Fig. 1. Illustration of the influence of the penalty parameter $\gamma > 0$ on the convergence of IPA for a DnCNN prior. The average normalized distance to $\mathrm{zer}(\mathsf{S})$ and SNR (dB) are plotted against the iteration number with the shaded areas representing the range of values attained over 12 test images. The accuracy of IPA improves for smaller values of γ. However, the SNR performance is nearly identical, indicating that in practice IPA can achieve excellent results for a range of fixed γ values.

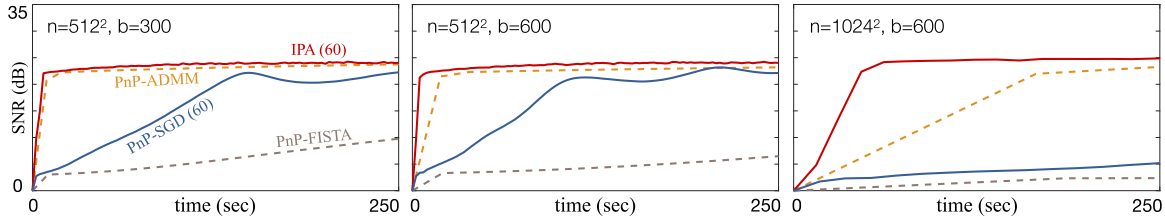


Fig. 2. Illustration of scalability of IPA and several widely used PnP algorithms on problems of different sizes. The parameters n and b denote the image size and the number of acquired intensity images, respectively. The average SNR is plotted against time in seconds. Both IPA and PnP-SGD use random minibatches of 60 measurements at every iteration, while PnP-ADMM and PnP-FISTA use all the measurements. The figure highlights the fast empirical convergence of IPA compared to PnP-SGD as well as its ability to address larger problems compared to PnP-ADMM and PnP-FISTA.

TABLE I
FINAL AVERAGE SNR (DB) AND RUNTIME OBTAINED BY SEVERAL PNP ALGORITHMS ON ALL TEST IMAGES

Simulations	**Parameters**		$n = 512^2$ (b = 300)	$n = 512^2$ (b = 600)	$n = 1024^2$ (b = 600)
Algorithms	σ	γ		SNR in dB (Runtime)	
PnP-FISTA	1	5×10^{-4}	22.60 (19.4 min)	22.79 (42.6 min)	23.56 (8.1 hr)
PnP-SGD (60)	1	5×10^{-4}	22.31 (7.1 min)	22.74 (5.2 min)	23.42 (44.3 min)
PnP-ADMM	2.5	1	**24.23** (7.4 min)	**24.40** (14.7 min)	**25.50** (1.4 hr)
IPA (60)	2.5	1	23.65 (**1.7 min**)	23.88 (**2 min**)	24.95 (**11 min**)

practice different γ values lead to fixed points of similar quality. This indicates that IPA can achieve high-quality result without taking $\|\mathsf{S}(\boldsymbol{v}^k)\|_2$ to zero.

B. Scalability in Large-Scale Optical Tomography

We now discuss the scalability of IPA on intensity diffraction tomography (IDT), which is a data intensive computational imaging modality [83]. The goal is to recover the spatial distribution of the *complex* permittivity contrast of an object given a set of its intensity-only measurements. In this problem, $\boldsymbol{A}$ consists of a set of b complex matrices $[\boldsymbol{A}_1, \ldots, \boldsymbol{A}_b]^{\mathsf{T}}$, where each $\boldsymbol{A}_i$ is a convolution corresponding to the ith measurement $\boldsymbol{y}_i$. We adopt the ℓ_2-norm loss $g(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$ as the data-fidelity term to empirically compare the performance of IPA and PnP-SGD on the same problem. PnP-SGD has been implemented with Nesterov acceleration, as in [20].

In the simulation, we follow the experimental setup in [83] under AWGN corresponding to an input SNR of 20 dB. We select six images from the CAT2000 dataset [92] as our test examples, each cropped to n pixels. We assume real permittivity functions,

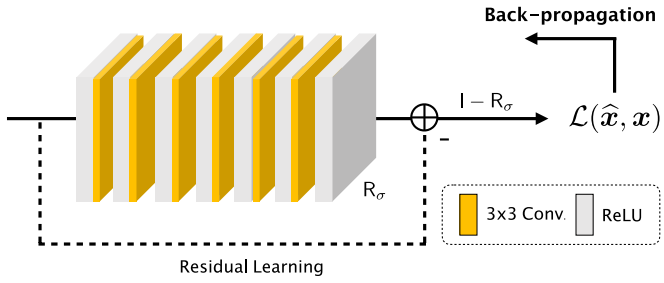


Fig. 3. Illustration of the architecture of DnCNN used in all experiments. Vectors $\widehat{\boldsymbol{x}}$ and $\boldsymbol{x}$ denote the denoised image and ground truth, respectively. The neural net is trained to remove the AWGN from its noisy input image. We also constrains the Lipschitz constant of R_σ to be smaller than 1 by using the spectral normalization technique in [84]. This provides a necessary condition for the satisfaction of Assumption 2.

TABLE II
PER-ITERATION MEMORY USAGE SPECIFICATION FOR RECONSTRUCTING 1024×1024 IMAGES

Algorithms		PnP-ADMM		IPA (**Ours**)	
Variables		size	memory	size	memory
$\{\boldsymbol{A}_i\}$	real	$1024 \times 1024 \times 600$	9.38 GB	$1024 \times 1024 \times 60$	0.94 GB
	imaginary	$1024 \times 1024 \times 600$	9.38 GB	$1024 \times 1024 \times 60$	0.94 GB
$\{\boldsymbol{y}_i\}$		$1024 \times 1024 \times 600$	18.75 GB	$1024 \times 1024 \times 60$	1.88 GB
others combined		—	0.13 GB	—	0.13 GB
Total			**37.63 GB**		**3.88 GB**

TABLE III
OVERVIEW OF SEVERAL EXISTING PNP/RED ALGORITHMS

Algorithms	**Nonsmooth**	**Online**
PnP-ADMM [1], [2], [17], [23]	✓	✗
PnP-ISTA/PnP-FISTA [18], [20], [93]	✗	✗
PnP-SPGM [20]	✗	✓
RED-SD [46]	✗	✗
RED-ADMM [46], [94]	✓	✗
prDeep [95]	✓	✗
RED-PG/RED-APG [94]	✓	✗
SIMBA/On-RED [96], [97]	✗	✓
IPA (proposed)	✓	✓

TABLE IV
PER-ITERATION MEMORY USAGE SPECIFICATION FOR RECONSTRUCTING 512×512 IMAGES

Algorithms		IPA (60)		PnP-ADMM (300)		PnP-ADMM (600)	
Variables		size	memory	size	memory	size	memory
$\{\boldsymbol{A}_i\}$	real	$512 \times 512 \times 60$	0.23 GB	$512 \times 512 \times 300$	1.17 GB	$512 \times 512 \times 600$	2.34 GB
	imaginary	$512 \times 512 \times 60$	0.23 GB	$512 \times 512 \times 300$	1.17 GB	$512 \times 512 \times 600$	2.34 GB
$\{\boldsymbol{y}_i\}$		$512 \times 512 \times 60$	0.47 GB	$512 \times 512 \times 300$	2.34 GB	$512 \times 512 \times 600$	4.69 GB
others combined		—	0.03 GB	—	0.03 GB	—	0.03 GB
Total			**0.97 GB**		**4.72 GB**		**9.41 GB**

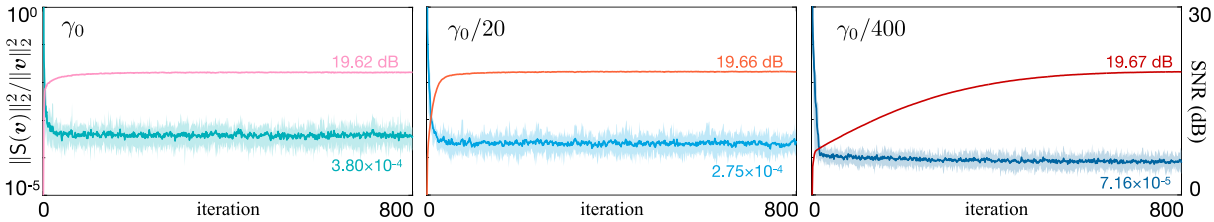


Fig. 4. Illustration of the convergence of IPA for a DnCNN prior under drastically changed γ values. The average normalized distance to $\mathrm{zer}(\mathsf{S})$ and SNR (dB) are plotted against the iteration number with the shaded areas representing the range of values attained over 12 test images. In practice, the convergence speed improves with larger values of γ. However, IPA still can achieve same level of SNR results for a wide range of γ values.

but still consider complex valued measurement operator $\boldsymbol{A}$ that accounts for both absorption and phase [83]. Due to the large size of data, we process the measurements in epochs using minbatches of size 60.

Fig. 2 illustrates the evolution of average SNR against runtime for several PnP algorithms, namely PnP-ADMM, PnP-FISTA, PnP-SGD, and IPA, for images of size $n \in \{512 \times 512, 1024 \times 1024\}$ and the total number of intensity measurements $b \in$

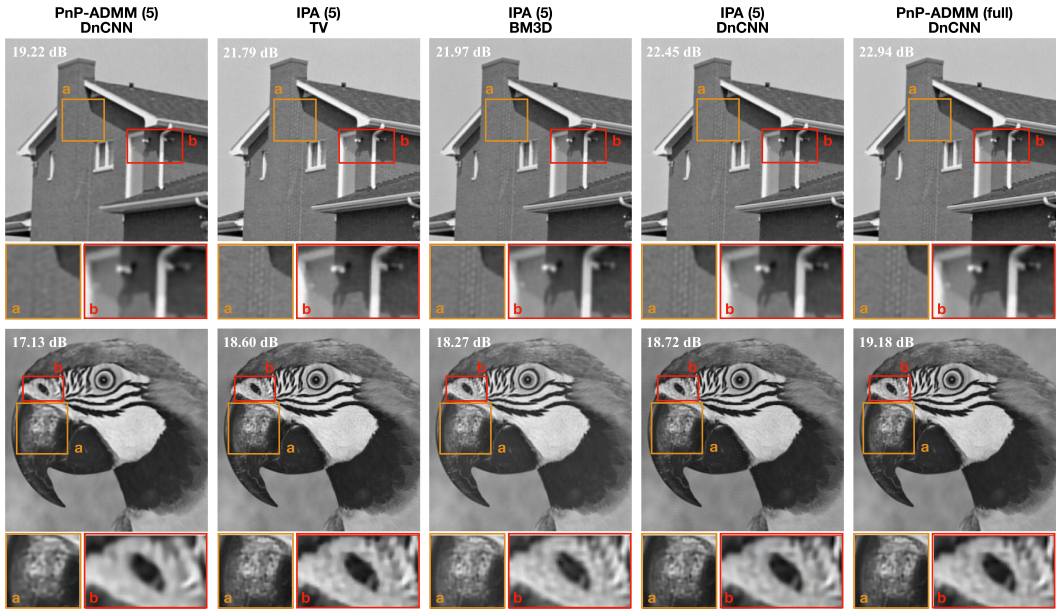


Fig. 5. Visual examples of the reconstructed House (upper) and Parrot (bottom) images by IPA and PnP-ADMM. The first and last columns correspond to PnP-ADMM under DnCNN with 5 fixed measurements and with the full 60 measurements, respectively. The second, third, and fourth column correspond to IPA with a small minibatch of size 5 under TV, BM3D, and DnCNN, respectively. Each image is labeled by its SNR (dB) with respect to the original image, and the visual difference is highlighted by the boxes underneath. Note that IPA recovers the details lost by the batch algorithm with the same computational cost and achieves the same high-quality results as the full batch algorithm.

$\{300, 600\}$. The final values of SNR as well as the total runtimes are summarized in Table I. The table highlights the overall best SNR performance in bold and the shortest runtime in light-green. In every iteration, PnP-ADMM and PnP-FISTA use all the measurements, while IPA and PnP-SGD use only a small subset of 60 measurements. IPA thus retains its effectiveness for large values of b, while batch algorithms become significantly slower. Moreover, the scalability of IPA over PnP-ADMM becomes more notable when the image size increases. For example, Table I highlights the convergence of IPA to 24.95 dB within 11 minutes, while PnP-ADMM takes 1.4 hours to reach a similar SNR value. Note the rapid progress of PnP-ADMM in the first few iterations, followed by a slow but steady progress until its convergence to the values reported in Table I. This behavior of ADMM is well known and has been widely reported in the literature (see *Section 3.2.2 "Convergence in Practice"* in [11]). We also observe faster convergence of IPA compared to both PnP-SGD and PnP-FISTA, further highlighting the potential of IPA to address large-scale problems where partial proximal operators are easy to evaluate.

Another key feature of IPA is its memory efficiency due to incremental processing of data. The memory considerations in optical tomography include the size of all the variables related to the desired image $\boldsymbol{x}$, the measured data $\{\boldsymbol{y}_i\}$, and the variables related to the forward model $\{\boldsymbol{A}_i\}$. Table II records the total memory (GB) used by IPA and PnP-ADMM for reconstructing a 1024×1024 pixel permittivity image, with the smallest value highlighted in light-green. PnP-ADMM requires 37.63 GB of memory due to its batch processing of the whole dataset, while IPA uses only 3.88 GB—nearly *one-tenth* of the former—by adopting incremental processing of data. In short, our numerical evaluations highlight both fast and stable convergence and flexible memory usage of IPA in the context of large-scale optical tomographic imaging.

VI. Conclusion

This work provides several new insights into the widely used PnP methodology in the context of large-scale imaging problems. First, we have proposed IPA as a new incremental PnP algorithm. IPA extends PnP-ADMM to randomized partial processing of measurements and extends traditional optimization-based ADMM by integrating pre-trained deep neural nets. Second, we have theoretically analyzed IPA under a set of realistic assumptions, showing that in expectation IPA can approximate the convergence behavior of PnP-ADMM to a desired precision by controlling the penalty parameter. Third, our simulations highlight the potential of IPA to handle nonsmooth data-fidelity terms, large number of measurements, and deep neural net priors. We observed faster convergence of IPA compared to several baseline PnP methods, including PnP-ADMM and PnP-SGD, when partial proximal operators can be efficiently evaluated. IPA can thus be an effective alternative to existing algorithms for addressing large-scale imaging problems. For future work, we would like to explore strategies to further relax our assumptions and explore distributed variants of IPA to enhance its performance in parallel settings.

APPENDIX

We adopt monotone operator theory [74], [82] for a unified analysis of IPA. In Appendix A, we present the convergence analysis of IPA. In Appendix B, we analyze the convergence of the algorithm for strongly convex data-fidelity terms and contractive denoisers. In Appendix C, we discuss interpretation of IPA's fixed-points from the perspective of monotone operator theory. For completeness, in Appendix D, we discuss the convergence results for traditional PnP-ADMM [23]. In Appendix E, we summarize the major similarities and differences of variations of PnP and RED algorithms. In Appendix F, we provide additional technical details of our deep neural net architecture, the computation of proximal operators, and results of additional simulations. Additionally, in Supplement I, we provide the background material used in our analysis, and summarize the major similarities. In Supplement II, we provide additional numerical results on the comparison with different image priors, the results on the validation of the firmly nonexpansiveness of our denoising neural network, the performance of IPA with different minibatch sizes and the detailed derivations on how the dual formulation was evaluated to estimate the solution of proximal operator used in our intensity diffraction tomography experiment.

For the sake of simplicity, we use $\|\cdot\|$ to denote the standard ℓ_2-norm in $\mathbb{R}^n$. We will also use $\mathsf{D}(\cdot)$ instead of $\mathsf{D}_\sigma(\cdot)$ to denote the denoiser, thus dropping the explicit notation for σ.

A. Convergence Analysis of IPA

In this section, we present one of the main results in this paper, namely the convergence analysis of IPA. A fixed-point convergence of averaged operators is well-known under the name of Krasnosel'skii-Mann theorem (see Section 5.2 in [82]) and was recently applied to the analysis of PnP-SGD [20]. Additionally, PnP-ADMM was analyzed for strongly convex data-fidelity terms g and contractive residual denoisers R_σ [23]. Our analysis extends these results to IPA by providing an explicit upper bound on the convergence. In Appendix A.1, we present the main steps of the proof, while in Appendix A.2 we prove two technical lemmas useful for our analysis.

1) Proof of Theorem 1: Appendix C.3 establishes that S defined in (10) is firmly nonexpansive. Consider any $\boldsymbol{v}^* \in \operatorname{zer}(\mathsf{S})$ and any $\boldsymbol{v} \in \mathbb{R}^n$, then we have

$$\begin{aligned}&\|\boldsymbol{v}-\boldsymbol{v}^*-\mathsf{S}\boldsymbol{v}\|^2\\&=\|\boldsymbol{v}-\boldsymbol{v}^*\|-2(\mathsf{S}\boldsymbol{v}-\mathsf{S}\boldsymbol{v}^*)^{\mathsf{T}}(\boldsymbol{v}-\boldsymbol{v}^*)+\|\mathsf{S}\boldsymbol{v}\|^2\\&\leq\|\boldsymbol{v}-\boldsymbol{v}^*\|^2-\|\mathsf{S}\boldsymbol{v}\|^2\,,\end{aligned}\tag{14}$$

where we used the firm nonexpansiveness of S and $\mathsf{S}\boldsymbol{v}^*=\boldsymbol{0}$. The direct consequence of (14) is that

$$\|\boldsymbol{v}-\boldsymbol{v}^*-\mathsf{S}\boldsymbol{v}\|\leq\|\boldsymbol{v}-\boldsymbol{v}^*\|\,.$$

We now consider the following two equivalent representations of IPA for some iteration $k\geq 1$

$$\begin{cases}\boldsymbol{z}^k=\mathsf{G}_{i_k}(\boldsymbol{x}^{k-1}+\boldsymbol{s}^{k-1})\\\boldsymbol{x}^k=\mathsf{D}(\boldsymbol{z}^k-\boldsymbol{s}^{k-1})\\\boldsymbol{s}^k=\boldsymbol{s}^{k-1}+\boldsymbol{x}^k-\boldsymbol{z}^k\end{cases}\tag{15a}$$

$$\Leftrightarrow\quad\begin{cases}\boldsymbol{x}^{k-1}=\mathsf{D}(\boldsymbol{v}^{k-1})\\\boldsymbol{z}^k=\mathsf{G}_{i_k}(2\boldsymbol{x}^{k-1}-\boldsymbol{v}^{k-1})\\\boldsymbol{v}^k=\boldsymbol{v}^{k-1}+\boldsymbol{z}^k-\boldsymbol{x}^{k-1}\,,\end{cases}\tag{15b}$$

where i_k is a random variable uniformly distributed over $\{1,\dots,b\}$, $\mathsf{G}_i=\operatorname{prox}_{\gamma g_i}$ is the proximal operator with respect to g_i, and D is the denoiser. To see the equivalence between (15a) and (15b), simply introduce the variable $\boldsymbol{v}^k=\boldsymbol{z}^k-\boldsymbol{s}^{k-1}$ into (15b) [23]. It is straightforward to verify that (15a) can also be rewritten as

$$\boldsymbol{v}^k=\boldsymbol{v}^{k-1}-\mathsf{S}_{i_k}(\boldsymbol{v}^{k-1})\text{ with }\mathsf{S}_{i_k}:=\mathsf{D}-\mathsf{G}_{i_k}(2\mathsf{D}-\mathsf{I})\,.\tag{16}$$

Then, for any $\boldsymbol{v}^*\in\operatorname{zer}(\mathsf{S})$, we have that

$$\begin{aligned}&\|\boldsymbol{v}^k-\boldsymbol{v}^*\|^2\\&=\|\boldsymbol{v}^{k-1}-\boldsymbol{v}^*-\mathsf{S}\boldsymbol{v}^{k-1}\|^2+\|\mathsf{S}\boldsymbol{v}^{k-1}-\mathsf{S}_{i_k}\boldsymbol{v}^{k-1}\|^2\\&\quad+2(\mathsf{S}\boldsymbol{v}^{k-1}-\mathsf{S}_{i_k}\boldsymbol{v}^{k-1})^{\mathsf{T}}(\boldsymbol{v}^{k-1}-\boldsymbol{v}^*-\mathsf{S}\boldsymbol{v}^{k-1})\\&\leq\|\boldsymbol{v}^{k-1}-\boldsymbol{v}^*\|^2-\|\mathsf{S}\boldsymbol{v}^{k-1}\|^2+\|\mathsf{S}\boldsymbol{v}^{k-1}-\mathsf{S}_{i_k}\boldsymbol{v}^{k-1}\|^2\\&\quad+2\|\mathsf{S}\boldsymbol{v}^{k-1}-\mathsf{S}_{i_k}\boldsymbol{v}^{k-1}\|\|\boldsymbol{v}^{k-1}-\boldsymbol{v}^*\|\\&\leq\|\boldsymbol{v}^{k-1}-\boldsymbol{v}^*\|^2-\|\mathsf{S}\boldsymbol{v}^{k-1}\|^2+\|\mathsf{S}\boldsymbol{v}^{k-1}-\mathsf{S}_{i_k}\boldsymbol{v}^{k-1}\|^2\\&\quad+2(R+2\gamma L)\|\mathsf{S}\boldsymbol{v}^{k-1}-\mathsf{S}_{i_k}\boldsymbol{v}^{k-1}\|\,,\end{aligned}$$

where in the first inequality we used Cauchy-Schwarz and (14), and in the second inequality we used Lemma 2 in Appendix A.2. By taking the conditional expectation on both sides, invoking Lemma 1 in Appendix A.2, and rearranging the terms, we get

$$\begin{aligned}\|\mathsf{S}\boldsymbol{v}^{k-1}\|^2\leq\|\boldsymbol{v}^{k-1}-\boldsymbol{v}^*\|^2-\mathbb{E}\left[\|\boldsymbol{v}^k-\boldsymbol{v}^*\|^2\mid\boldsymbol{v}^{k-1}\right]\\+4\gamma LR+12\gamma^2L^2\,.\end{aligned}$$

Hence, by averaging over $t\geq 1$ iterations and taking the total expectation, we obtain

$$\mathbb{E}\left[\frac{1}{t}\sum_{k=1}^{t}\|\mathsf{S}\boldsymbol{v}^{k-1}\|^2\right]\leq\frac{(R+2\gamma L)^2}{t}+4\gamma LR+12\gamma^2L^2\,.$$

The final result is obtained by noting that

$$4\gamma LR+12\gamma^2L^2\leq\max\{\gamma,\gamma^2\}(4LR+12L^2)\,.$$

2) Lemmas Useful for the Proof of Theorem 1: This section presents two technical lemmas used in our analysis in Appendix A.1.

Lemma 1: Assume that Assumptions 1-3 hold and let i_k be a uniform random variable over $\{1,\dots,b\}$. Then, we have that

$$\mathbb{E}\left[\|\mathsf{S}_{i_k}\boldsymbol{v}-\mathsf{S}\boldsymbol{v}\|^2\right]\leq 4\gamma^2L^2,\quad\boldsymbol{v}\in\mathbb{R}^n\,.$$

Proof: Let $\boldsymbol{z}_i=\mathsf{G}_i(\boldsymbol{x})$ and $\boldsymbol{z}=\mathsf{G}(\boldsymbol{x})$ for any $1\leq i\leq b$ and $\boldsymbol{x}\in\mathbb{R}^n$. From the optimality conditions for each proximal operator

$$\mathsf{G}_i\boldsymbol{x}=\operatorname{prox}_{\gamma g_i}(\boldsymbol{x})=\boldsymbol{x}-\gamma\boldsymbol{g}_i(\boldsymbol{z}_i),\quad\boldsymbol{g}_i(\boldsymbol{z}_i)\in\partial g_i(\boldsymbol{z}_i)$$

and

$$\mathsf{G}\boldsymbol{x} = \mathrm{prox}_{\gamma g}(\boldsymbol{x}) = \boldsymbol{x} - \gamma \boldsymbol{g}(\boldsymbol{z})$$

such that

$$\boldsymbol{g}(\boldsymbol{z}) = \frac{1}{b}\sum_{i=1}^{b} \boldsymbol{g}_i(\boldsymbol{z}) \in \partial g(\boldsymbol{z}) \ ,$$

where we used Proposition 7 in Supplement I-B. By using the bound on all the subgradients (due to Assumption 1 and Proposition 8 in Supplement I-B), we obtain

$$\begin{aligned}\|\mathsf{G}_i(\boldsymbol{x}) - \mathsf{G}(\boldsymbol{x})\| &= \|\mathrm{prox}_{\gamma g_i}(\boldsymbol{x}) - \mathrm{prox}_{\gamma g}(\boldsymbol{x})\| \\ &= \gamma\|\boldsymbol{g}_i(\boldsymbol{z}_i) - \boldsymbol{g}(\boldsymbol{z})\| \leq 2\gamma L \ ,\end{aligned}$$

where $L > 0$ is the Lipschitz constant of all g_is and g. This inequality directly implies that

$$\|\mathsf{S}\boldsymbol{v} - \mathsf{S}_i\boldsymbol{v}\| = \|\mathsf{G}(2\mathsf{D}\boldsymbol{v} - \boldsymbol{v}) - \mathsf{G}_i(2\mathsf{D}\boldsymbol{v} - \boldsymbol{v})\| \leq 2\gamma L \ .$$

Since, this inequality holds for every i, it also holds in expectation.

Lemma 2: Assume that Assumptions 1-3 hold and let the sequence $\{\boldsymbol{v}^k\}$ be generated via the iteration (16). Then, for any $k \geq 1$, we have that

$$\|\boldsymbol{v}^k - \boldsymbol{v}^*\| \leq (R + 2\gamma L) \quad \text{for all} \quad \boldsymbol{v}^* \in \mathrm{zer}(\mathsf{S}) \ .$$

Proof: The optimality of the proximal operator in (16) implies that there exists $\boldsymbol{g}_{i_k}(\boldsymbol{z}^k) \in \partial g_{i_k}(\boldsymbol{z}^k)$ such that

$$\begin{aligned}&\boldsymbol{z}^k = \mathsf{G}_{i_k}(2\boldsymbol{x}^{k-1} - \boldsymbol{v}^{k-1}) \\ \Leftrightarrow \quad &2\boldsymbol{x}^{k-1} - \boldsymbol{v}^{k-1} - \boldsymbol{z}^k = \gamma \boldsymbol{g}_{i_k}(\boldsymbol{z}^k) \ .\end{aligned}$$

By applying $\boldsymbol{v}^k = \boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}(\boldsymbol{v}^{k-1}) = \boldsymbol{v}^{k-1} + \boldsymbol{z}^k - \boldsymbol{x}^{k-1}$ to the equality above, we obtain

$$\boldsymbol{x}^{k-1} - \boldsymbol{v}^k = \gamma \boldsymbol{g}_{i_k}(\boldsymbol{z}^k) \quad \Leftrightarrow \quad \boldsymbol{v}^k = \boldsymbol{x}^{k-1} - \gamma \boldsymbol{g}_{i_k}(\boldsymbol{z}^k) \ .$$

Additionally, for any $\boldsymbol{v}^* \in \mathrm{zer}(\mathsf{S})$ and $\boldsymbol{x}^* = \mathsf{D}(\boldsymbol{v}^*)$, we have that

$$\begin{aligned}&\mathsf{S}(\boldsymbol{v}^*) = \mathsf{D}(\boldsymbol{v}^*) - \mathsf{G}(2\mathsf{D}(\boldsymbol{v}^*) - \boldsymbol{v}^*) = \boldsymbol{x}^* - \mathsf{G}(2\boldsymbol{x}^* - \boldsymbol{v}^*) = \boldsymbol{0} \\ \Rightarrow \quad &\boldsymbol{x}^* - \boldsymbol{v}^* = \gamma \boldsymbol{g}(\boldsymbol{x}^*) \quad \text{for some} \quad \boldsymbol{g}(\boldsymbol{x}^*) \in \partial g(\boldsymbol{x}^*) \ .\end{aligned}$$

Thus, by using Assumption 3 and the bounds on all the subgradients (due to Assumption 1 and Proposition 8 in Supplement I-B), we obtain

$$\begin{aligned}\|\boldsymbol{v}^k - \boldsymbol{v}^*\| &= \|\boldsymbol{x}^{k-1} - \gamma \boldsymbol{g}_{i_k}(\boldsymbol{z}^k) - \boldsymbol{x}^* - \gamma \boldsymbol{g}(\boldsymbol{x}^*)\| \\ &\leq \|\boldsymbol{x}^{t-1} - \boldsymbol{x}^*\| + 2\gamma L \leq (R + 2\gamma L) \ .\end{aligned}$$

B. Analysis of IPA for Strongly Convex Functions

In this section, we perform analysis of IPA under a different set of assumptions, namely under the assumptions adopted in [23].

Assumption 4: Each g_i is proper, closed, strongly convex with constant $M_i > 0$, and Lipschitz continuous with constant $L_i > 0$. We define the smallest strong convexity constant as $M = \min\{M_1, \ldots, M_b\}$ and the largest Lipschitz constant as $L = \max\{L_1, \ldots, L_b\}$.

This assumption further restricts Assumption 1 to strongly convex functions.

Assumption 5: The residual $\mathsf{R}_\sigma := \mathsf{I} - \mathsf{D}_\sigma$ of the denoiser D_σ is a contraction. It thus satisfies

$$\|\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y}\| \leq \epsilon\|\boldsymbol{x} - \boldsymbol{y}\| \ ,$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ for some constant $0 < \epsilon < 1$.

This assumption replaces Assumption 2 by assuming that the residual of the denoiser is a contraction. Note that this can be practically imposed on deep neural net denoisers via spectral normalization [80]. We can then state the following.

Theorem 3: Run IPA for $t \geq 1$ iterations with random i.i.d. block selection under Assumptions 3-5 using a fixed penalty parameter $\gamma > 0$. Then, the iterates of IPA satisfy

$$\mathbb{E}\left[\|\boldsymbol{x}^k - \boldsymbol{x}^*\|\right] \leq \eta^k(2R + 4\gamma L) + \frac{4\gamma L}{1-\eta}, \quad 0 < \eta < 1 \ .$$

Proof: It was shown in Theorem 2 of [23] that under Asumptions 4 and 5, we have that

$$\|(\mathsf{I} - \mathsf{S})\boldsymbol{x} - (\mathsf{I} - \mathsf{S})\boldsymbol{y}\| \leq \eta\|\boldsymbol{x} - \boldsymbol{y}\| \tag{17}$$

with

$$\eta := \left(\frac{1 + \epsilon + \epsilon\gamma M + 2\epsilon^2\gamma M}{1 + \gamma M + 2\epsilon\gamma M}\right) \ ,$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, where S is given in (10). Hence, when

$$\frac{\epsilon}{\gamma M(1 + \epsilon - 2\epsilon^2)} < 1 \ ,$$

the operator $(\mathsf{I} - \mathsf{S})$ is a contraction. Using the reasoning in Appendix A, the sequence $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ can be written as

$$\boldsymbol{v}^k = \boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}(\boldsymbol{v}^{k-1}) \text{ with } \mathsf{S}_{i_k} := \mathsf{D} - \mathsf{G}_{i_k}(2\mathsf{D} - \mathsf{I}) \ . \tag{18}$$

Then, for any $\boldsymbol{v}^* \in \mathrm{zer}(\mathsf{S})$, we have that

$$\begin{aligned}&\|\boldsymbol{v}^k - \boldsymbol{v}^*\|^2 \\ &= \|(\mathsf{I} - \mathsf{S})\boldsymbol{v}^{k-1} - (\mathsf{I} - \mathsf{S})\boldsymbol{v}^*\|^2 \\ &\quad + 2((\mathsf{I} - \mathsf{S})\boldsymbol{v}^{k-1} - (\mathsf{I} - \mathsf{S})\boldsymbol{v}^*)^{\mathsf{T}}((\mathsf{I} - \mathsf{S}_{i_k})\boldsymbol{v}^{k-1} - \\ &\quad (\mathsf{I} - \mathsf{S})\boldsymbol{v}^{k-1}) + \|(\mathsf{I} - \mathsf{S}_{i_k})\boldsymbol{v}^{k-1} - (\mathsf{I} - \mathsf{S})\boldsymbol{v}^{k-1}\|^2 \\ &\leq \eta^2\|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\|^2 + 2\eta\|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\|\|\mathsf{S}_{i_k}\boldsymbol{v}^{k-1} - \mathsf{S}\boldsymbol{v}^{k-1}\| \\ &\quad + \|\mathsf{S}_{i_k}\boldsymbol{v}^{k-1} - \mathsf{S}\boldsymbol{v}^{k-1}\|^2 \ ,\end{aligned}$$

where we used the Cauchy-Schwarz inequality and the fact that $(\mathsf{I} - \mathsf{S})$ is η-contractive. By taking the conditional expectation on both sides, invoking Lemma 1 in Appendix A.2, and completing the square, we get

$$\mathbb{E}\left[\|\boldsymbol{v}^k - \boldsymbol{v}^*\|^2 | \boldsymbol{v}^{k-1}\right] \leq \left(\eta\|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\| + 2\gamma L\right)^2 \ .$$

Then, by applying the Jensen inequality and taking the total expectation, we get

$$\mathbb{E}\left[\|\boldsymbol{v}^k - \boldsymbol{v}^*\|\right] \leq \eta\mathbb{E}\left[\|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\|\right] + 2\gamma L \ .$$

By iterating this result and invoking Lemma 2 from Appendix A.2, we obtain

$$\mathbb{E}\left[\|\boldsymbol{v}^k - \boldsymbol{v}^*\|\right] \leq \eta^k(R + 2\gamma L) + (2\gamma L)/(1 - \eta) \ .$$

Finally by using the nonexpansiveness of $(1/(1+\epsilon))\mathsf{D}$ (see Lemma 9 in [23]) and the fact that $\boldsymbol{x}^* = \mathsf{D}(\boldsymbol{v}^*)$, we obtain

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{x}^k - \boldsymbol{x}^*\|\right] &\leq (1+\epsilon)\left[\eta^k(R+2\gamma L) + \frac{2\gamma L}{1-\eta}\right] \\ &\leq \eta^k(2R+4\gamma L) + \frac{4\gamma L}{1-\eta}\,.\end{aligned}$$

This concludes the proof.

C. Fixed Point Interpretation

Fixed points of PnP algorithms have been extensively discussed in the recent literature [18], [19], [23]. Our goal in this section is to revisit this topic in a way that leads to a more intuitive equilibrium interpretation of PnP. Our formulation has been inspired from the classical interpretation of ADMM as an algorithm for computing a zero of a sum of two monotone operators [8].

1) Equilibrium Points of PnP Algorithms: It is known that a fixed point $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{s}^*)$ of PnP-ADMM (and of all PnP algorithms [18]) satisfies

$$\boldsymbol{x}^* = \mathsf{G}(\boldsymbol{x}^* + \boldsymbol{s}^*) \quad \text{and} \quad \boldsymbol{x}^* = \mathsf{D}(\boldsymbol{x}^* - \boldsymbol{s}^*)\,, \tag{19}$$

with $\boldsymbol{x}^* = \boldsymbol{z}^*$, where $\mathsf{G} = \operatorname{prox}_{\gamma g}$. Consider the *inverse* of D at $\boldsymbol{x} \in \mathbb{R}^n$, which is a set-valued operator $\mathsf{D}^{-1}(\boldsymbol{x}) := \{\boldsymbol{z} \in \mathbb{R}^n : \boldsymbol{x} = \mathsf{D}_\sigma(\boldsymbol{z})\}$. Note that the inverse operator exists even when D is not a bijection (see Section 2 of [74]). Then, from the definition of D^{-1} and optimality conditions of the proximal operator, we can equivalently rewrite (19) as follows

$$\boldsymbol{s}^* \in \gamma \partial g(\boldsymbol{x}^*) \quad \text{and} \quad -\boldsymbol{s}^* \in \mathsf{D}^{-1}(\boldsymbol{x}^*) - \boldsymbol{x}^*\,.$$

This directly leads to the following equivalent representation of PnP fixed points

$$\boldsymbol{0} \in \mathsf{T}(\boldsymbol{x}^*) := \gamma \partial g(\boldsymbol{x}^*) + (\mathsf{D}^{-1}(\boldsymbol{x}^*) - \boldsymbol{x}^*)\,. \tag{20}$$

Hence, a vector $\boldsymbol{x}^*$ computed by PnP can be interpreted as an equilibrium point between two terms with $\gamma > 0$ explicitly influencing the balance.

2) Equivalence of Zeros of T and S: Define $\boldsymbol{v}^* := \boldsymbol{z}^* - \boldsymbol{s}^*$ for a given fixed point $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{s}^*)$ of PnP-ADMM and consider the operator

$$\mathsf{S} = \mathsf{D} - \mathsf{G}(2\mathsf{D} - \mathsf{I}) \quad \text{with} \quad \mathsf{G} = \operatorname{prox}_{\gamma g}\,,$$

which was defined in (10). Note that from (19), we also have $\boldsymbol{x}^* = \mathsf{D}(\boldsymbol{v}^*)$ and $\boldsymbol{v}^* = \boldsymbol{x}^* - \boldsymbol{s}^*$ (due to $\boldsymbol{z}^* = \boldsymbol{x}^*$). We then have the following equivalence

$$\begin{aligned}&\boldsymbol{0} \in \mathsf{T}(\boldsymbol{x}^*) = \gamma \partial g(\boldsymbol{x}^*) + (\mathsf{D}^{-1}(\boldsymbol{x}^*) - \boldsymbol{x}^*) \\ \Leftrightarrow\ &\begin{cases}\boldsymbol{x}^* = \mathsf{G}(\boldsymbol{x}^* + \boldsymbol{s}^*) \\ \boldsymbol{x}^* = \mathsf{D}(\boldsymbol{x}^* - \boldsymbol{s}^*)\end{cases} \\ \Leftrightarrow\ &\begin{cases}\boldsymbol{x}^* = \mathsf{G}(2\boldsymbol{x}^* - \boldsymbol{v}^*) \\ \boldsymbol{x}^* = \mathsf{D}(\boldsymbol{v}^*)\end{cases} \\ \Leftrightarrow\ &\mathsf{S}(\boldsymbol{v}^*) = \mathsf{D}(\boldsymbol{v}^*) - \mathsf{G}(2\mathsf{D}(\boldsymbol{v}^*) - \boldsymbol{v}^*) = \boldsymbol{0}\,,\end{aligned}$$

where we used the optimality conditions of the proximal operator G. Hence, the condition that $\boldsymbol{v}^* = \boldsymbol{z}^* - \boldsymbol{s}^* \in \operatorname{zer}(\mathsf{S})$ is equivalent to $\boldsymbol{x}^* = \mathsf{D}(\boldsymbol{v}^*) \in \operatorname{zer}(\mathsf{T})$.

3) Firm Nonexpansiveness of S: We finally would like to show that under Assumptions 1-3, the operator S is firmly nonexpansive. Assumption 2 and Proposition 6 in Supplement I-B imply that D and G are firmly nonexpansive. Then, Proposition 4 in Supplement I-A implies that $(2\mathsf{D} - \mathsf{I})$ and $(2\mathsf{G} - \mathsf{I})$ are nonexpansive. Thus, the composition $(2\mathsf{G} - \mathsf{I})(2\mathsf{D} - \mathsf{I})$ is also nonexpansive and

$$(\mathsf{I} - \mathsf{S}) = \frac{1}{2}\mathsf{I} + \frac{1}{2}(2\mathsf{G} - \mathsf{I})(2\mathsf{D} - \mathsf{I}) \tag{21}$$

is $(1/2)$-averaged. Then, Proposition 4 in Supplement I-A implies that S is firmly nonexpansive.

D. Convergence Analysis of PnP-ADMM

The following analysis has been adopted from [23]. For completeness, we summarize the key results useful for our own analysis by restating them under the assumptions in Section IV.

1) Equivalence Between PnP-ADMM and PnP-DRS: An elegant analysis of PnP-ADMM emerges from its interpretation as the Douglas–Rachford splitting (DRS) algorithm [23]. This equivalence is well-known and has been extensively studied in the context of convex optimization [8]. Here, we restate the relationship for completeness.

Consider the sequences of DRS (top) and ADMM (bottom)

$$\begin{cases}\boldsymbol{x}^{k-1} = \mathsf{D}(\boldsymbol{v}^{k-1}) \\ \boldsymbol{z}^k = \mathsf{G}(2\boldsymbol{x}^{k-1} - \boldsymbol{v}^{k-1}) \\ \boldsymbol{v}^k = \boldsymbol{v}^{k-1} + \boldsymbol{z}^k - \boldsymbol{x}^{k-1}\end{cases}$$

$$\Leftrightarrow \quad \begin{cases}\boldsymbol{z}^k = \mathsf{G}(\boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1}) \\ \boldsymbol{x}^k = \mathsf{D}(\boldsymbol{z}^k - \boldsymbol{s}^{k-1}) \\ \boldsymbol{s}^k = \boldsymbol{s}^{k-1} + \boldsymbol{x}^k - \boldsymbol{z}^k\,,\end{cases}$$

where $\mathsf{G} := \operatorname{prox}_{\gamma g}$ is the proximal operator and D is the denoiser. To see the equivalence between them, simply introduce the variable change $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ into DRS. Note also the DRS sequence can be equivalently written as

$$\boldsymbol{v}^k = \boldsymbol{v}^{k-1} - \mathsf{S}(\boldsymbol{v}^{k-1}) \quad \text{with} \quad \mathsf{S} := \mathsf{D} - \mathsf{G}(2\mathsf{D} - \mathsf{I})\,.$$

To see this simply rearrange the terms in DRS as follows

$$\begin{aligned}\boldsymbol{v}^k &= \boldsymbol{v}^{k-1} + \mathsf{G}(2\boldsymbol{x}^{k-1} - \boldsymbol{v}^{k-1}) - \boldsymbol{x}^{k-1} \\ &= \boldsymbol{v}^{k-1} - \left[\mathsf{D}(\boldsymbol{v}^{k-1}) - \mathsf{G}(2\mathsf{D}(\boldsymbol{v}^{k-1}) - \boldsymbol{v}^{k-1})\right]\,.\end{aligned}$$

2) Convergence Analysis of PnP-DRS and PnP-ADMM: It was established in Appendix C.3 that S defined in (10) is firmly nonexpansive.

Consider a single iteration of DRS $\boldsymbol{v}^+ = \boldsymbol{v} - \mathsf{S}\boldsymbol{v}$. Then, for any $\boldsymbol{v}^* \in \operatorname{zer}(\mathsf{S})$, we have

$$\begin{aligned}\|\boldsymbol{v}^+ - \boldsymbol{v}^*\|^2 &= \|\boldsymbol{v} - \boldsymbol{v}^*\|^2 - 2(\mathsf{S}\boldsymbol{v} - \mathsf{S}\boldsymbol{v}^*)^{\mathsf{T}}(\boldsymbol{v} - \boldsymbol{v}^*) + \|\mathsf{S}\boldsymbol{v}\|^2 \\ &\leq \|\boldsymbol{v} - \boldsymbol{v}^*\|^2 - \|\mathsf{S}\boldsymbol{v}\|^2\,,\end{aligned}$$

where we used $\mathsf{S}\boldsymbol{v}^* = \boldsymbol{0}$ and firm nonexpansiveness of S. By rearranging the terms, we obtain the following upper bound at

iteration $k \geq 1$

$$\|\mathsf{S}\boldsymbol{v}^{k-1}\|^2 \leq \|\boldsymbol{v}^{k-1} - \boldsymbol{v}^*\|^2 - \|\boldsymbol{v}^k - \boldsymbol{v}^*\|^2 \,. \tag{22}$$

By averaging the inequality (22) over $t \geq 1$ iterations, we obtain

$$\frac{1}{t}\sum_{k=1}^{t} \|\mathsf{S}\boldsymbol{v}^{k-1}\|^2 \leq \frac{\|\boldsymbol{v}^0 - \boldsymbol{v}^*\|^2}{t} \leq \frac{(R+2\gamma L)^2}{t}$$

where used the bound on $\|\boldsymbol{v}^0 - \boldsymbol{v}^*\| \leq (R + 2\gamma L)$ that can be easily obtained by following the steps in Lemma 2 in Appendix A.2.

This result directly implies that $\|\mathsf{S}\boldsymbol{v}^t\| \to 0$ as $t \to 0$. Additionally, Krasnosel'skii-Mann theorem (see Section 5.2 in [82]) implies that $\boldsymbol{v}^t \to \mathrm{zer}(\mathsf{S})$. Then, from continuity of D, we have that $\boldsymbol{x}^t = \mathsf{D}(\boldsymbol{v}^t) \to \mathrm{zer}(\mathsf{T})$ (see also Appendix C.2). This completes the proof.

E. Variants of PnP/RED Algorithms

Several variants of PnP/RED algorithms are summarized in Table III, focusing on two properties (a) the ability to handle nonsmooth data-fidelity terms, and (b) the ability to handle online/minibatch processing of the measurements. The table highlights the way IPA complements existing work by addressing both (a) and (b).

F. Additional Technical Details

In this section, we present several technical details of our experiments. Section VI-F1 discusses the architecture and training of the DnCNN prior. Section VI-F2 explains the computation of the proximal operators used in our experiments. Section VI-F3 presents extra details and validations that complement the experiments in Section V with additional insights for IPA.

1) Architecture and Training of the DnCNN Prior: Fig. 3 illustrates the architectural details of the DnCNN prior used in our experiments. In total, the network contains 7 layers, of which the first 6 layers consist of a convolutional layer and a rectified linear unit (ReLU), while the last layer is just a convolution. A skip connection from the input to the output is implemented to enforce residual learning. The output images of the first 6 layers have 64 feature maps while that of the last layer is a single-channel image. We set all convolutional kernels to be 3×3 with stride 1, so that intermediate images have the same spatial size as the input image. We generated 11 101 training examples by adding AWGN to 400 images from the BSD400 dataset [85] and extracting patches of 128×128 pixels with stride 64. We trained DnCNN to optimize the *mean squared error* by using the Adam optimizer [98].

We use the spectral normalization technique in [84] to control the global Lipschitz constant (LC) of DnCNN. In the training, we constrain the residual network R_σ to have LC smaller than 1. Since the firm non-expansiveness implies non-expansiveness, this provides a *necessary* condition for R_σ to satisfy Assumption 2. The training of DnCNN *with* and *without* spectral normalization takes 4 and 1.82 hours, respectively, on the same hardware. Thus, for about $2\times$ increase in the denoiser pre-training time, one can make IPA/PnP-ADMM convergent.

2) Computation of Proximal Operators: In the CS experiments, the measurement matrix $\boldsymbol{A}$ is a random matrix, and the data-fidelity term is based on the ℓ_1-norm: $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_1$. While closed form solution of the proximal operator is inaccessible in this setting, we can efficiently approximate the proximal solution in the dual domain by using projected gradient method (PGM) [28], a derivation of which can be found in the supplement. Note that the closed-form solution is also unavailable for other ℓ_1-based proximal operators [28], [99]. The stopping criteria for the PGM algorithm are that either that the total iterations exceeds 200, or that the relative change between two iterates is below 1×10^{-4}. Detailed derivations are in Supplement II-C.

For intensity diffraction tomography (IDT), we adopted the linearized forward model developed in [83], which is based on the Fourier transform. For the i^{th} measurement, the forward model for the 2-dimensional case is described as $\boldsymbol{A}_i = \boldsymbol{F}^{\mathsf{H}}\boldsymbol{H}_i\boldsymbol{F}$, where $\boldsymbol{F}$ and $\boldsymbol{F}^{\mathsf{H}}$ denote the discrete Fourier transform and its imverse, respectively, and $\boldsymbol{H}_i$ corresponds to light transfer function of the i^{th} illumination. Under the ℓ_2-norm, we can directly derive the closed-form solution of the proximal operator in the Fourier space [9], [58].

3) Extra Details and Validations for Optical Tomography: All experiments were run on the machine equipped with an Intel Core i7 Processor that has 6 cores of 3.2 GHz and 32 GBs of DDR memory. We trained all neural nets using NVIDIA RTX 2080 GPUs. We define the SNR (dB) used in the experiments as

$$\mathrm{SNR}(\hat{\boldsymbol{x}}, \boldsymbol{x}) \triangleq \max_{a,b\in\mathbb{R}} \left\{ 20\log_{10}\left(\frac{\|\boldsymbol{x}\|_{\ell_2}}{\|\boldsymbol{x} - a\hat{\boldsymbol{x}} + b\|_{\ell_2}}\right)\right\} \,,$$

where $\hat{\boldsymbol{x}}$ is the estimate and $\boldsymbol{x}$ is the ground truth.

For intensity diffraction tomography, we implemented an epoch-based selection rule due to the large size of data. We randomly divide the measurements (along with the corresponding forward operators) into non-overlapping chunks of size 60 and save these chunks on the hard drive. At every iteration, IPA loads only a single random chunk into the memory while the full-batch PnP-ADMM loads all chunks sequentially and process the full set of measurements. This leads to the lower per iteration cost and less memory usage of IPA than PnP-ADMM. Table IV shows extra examples of the memory usage specification for reconstructing 512×512 pixel permittivity images. These results follow the same trend observed in Table II. We also conduct some extra validations that provide additional insights into IPA. In these simulations, we use images of size 254×254 pixels from *Set 12* as test examples. We assume real permittivity functions with the total number of measurement $b = 60$.

Fig. 4 illustrates the evolution of the convergence of IPA for different values of the penalty parameter. We consider three different values of $\gamma \in \{\gamma_0, \gamma_0/20, \gamma/400\}$ with $\gamma_0 = 20$. The average normalized distance $\|\mathsf{S}(\boldsymbol{v}^k)\|_2^2/\|\boldsymbol{v}^k\|_2^2$ and SNR are plotted against the iteration number and labeled with their respective final values. The shaded areas represent the range of values attained across all test images. IPA randomly selects 5 measurements in every iteration to impose the data-consistency. Fig. 4 complements the results in Fig. 1 by showing the fast convergence speed in practice with larger values of γ. On the

other hand, this plot further demonstrates that IPA is stable in terms of the SNR results for a wide range of γ values.

Our final simulation compares the reconstruction performance of IPA using TV, BM3D, and DnCNN. Since TV has a proximal operator, it serves as a baseline. The reconstruction performance of IPA on *House* and *Parrot* are presented in Fig. 5, while average SNR values for additional images are presented in Table I of the supplementary material. We include the results of PnP-ADMM using 5 fixed measurements and the full batch as reference. First, note the significant improvement of IPA over PnP-ADMM under the same computational budget. Second, using learned priors in IPA leads to better reconstruction than other priors. For instance, DnCNN outperforms TV and BM3D by 0.7 dB in SNR. Finally, the agreement between IPA and the full batch PnP-ADMM highlights the nearly optimal performace of IPA at a lower computational cost and memory usage.

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. Inf. Process.*, Austin, TX, USA, Dec. 3-5, 2013, pp. 945–948.

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

[3] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 123–231, 2014.

[4] 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, Aug. 2003.

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

[6] J. Bect, L. Blanc-Feraud, G. Aubert, and A. Chambolle, "A ℓ_1-unified variational framework for image restoration," in *Proc. Euro. Conf. Comput. Vis.*, New York, vol. 3024, 2004, pp. 1–13.

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

[8] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Prog.*, vol. 55, no. 1, pp. 293–318, 1992.

[9] 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, Sep. 2010.

[10] 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, Aug. 2010.

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

[12] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3929–3938.

[13] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising prior driven deep neural network for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2305–2318, Oct. 2019.

[14] K. Zhang, W. Zuo, and L. Zhang, "Deep plug-and-play super-resolution for arbitrary blur kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1671–1681.

[15] R. Ahmad *et al.*, "Plug-and-play methods for magnetic resonance imaging: Using denoisers for image recovery," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 105–116, Jan. 2020.

[16] K. Wei, A. I. Avilés-Rivero, J. Liang, Y. Fu, C. Schönlieb, and H. Huang, "Tuning-free plug-and-play proximal algorithm for inverse imaging problems," in *Proc. 37th Int. Conf. Mach. Learn.*, Jul. 13–18, 2020, pp. 10 158–10 169.

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

[18] 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. Comput. Vis.*, Venice, Italy, 2017, pp. 1799–1808.

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

[20] Y. Sun, B. Wohlberg, and U. S. Kamilov, "An online plug-and-play algorithm for regularized image reconstruction," *IEEE Trans. Comput. Imag.*, vol. 5, no. 3, pp. 395–408, Sep. 2019.

[21] T. Tirer and R. Giryes, "Image restoration by iterative denoising and backward projections," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1220–1234, Mar. 2019.

[22] A. M. Teodoro, J. M. Bioucas-Dias, and M. 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.

[23] E. K. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin, "Plug-and-play methods provably converge with properly trained denoisers," in *Proc. 36th Int. Conf. Mach. Learn.*, Long Beach, CA, USA, 2019, pp. 5546–5557.

[24] X. Xu, J. Liu, Y. Sun, B. Wohlberg, and U. S. Kamilov, "Boosting the performance of plug-and-play priors via denoiser scaling," in *Proc. 54th Asilomar Conf. Signals, Syst., Comput.*, 2020, pp. 1305–1312.

[25] X. Xu, Y. Sun, J. Liu, B. Wohlberg, and U. S. Kamilov, "Provable convergence of plug-and-play priors with MMSE denoisers," *IEEE Signal Process. Lett.*, vol. 27, pp. 1280–1284, Jul. 2020.

[26] E. J. Candés, 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, Feb. 2006.

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

[28] 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, Nov. 2009.

[29] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, 2010.

[30] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[31] 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, Jul. 2017.

[32] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Rev.*, vol. 60, no. 2, pp. 223–311, 2018.

[33] H. Wang and A. Banerjee, "Online alternating direction method," in *Proc. 29th Int. Conf. Mach. Learn.*, Edinburgh, Scotland, U.K., Jun./Jul., 2012, pp. 1699–1706.

[34] H. Ouyang, N. He, L. Q. Tran, and A. Gray, "Stochastic alternating direction method of multipliers," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, Jun. 16-21, 2013, pp. 80–88.

[35] T. Suzuki, "Dual averaging and proximal gradient descent for online alternating direction multiplier method," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, Jun. 16-21, 2013, pp. 392–400.

[36] W. Zhong and J. Kwok, "Fast stochastic alternating direction method of multipliers," in *Proc. 31th Int. Conf. Mach. Learn., Bejing, China*, Jun. 22-24, 2014, pp. 46–54.

[37] F. Huang, S. Chen, and H. Huang, "Faster stochastic alternating direction method of multipliers for nonconvex optimization," in *Proc. 36th Int. Conf. Mach. Learn.*, Long Beach, CA, USA, Jun. 10-15, 2019, pp. 2839–2848.

[38] M. T. McCann, K. H. Jin, and M. Unser, "Convolutional neural networks for inverse problems in imaging: A review," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 85–95, Nov. 2017.

[39] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos, "Using deep neural networks for inverse problems in imaging: Beyond analytical methods," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 20–36, Jan. 2018.

[40] F. Knoll *et al.*, "Deep-learning methods for parallel magnetic resonance imaging reconstruction: A survey of the current approaches, trends, and issues," *IEEE Signal Process. Mag.*, vol. 37, no. 1, pp. 128–140, Jan. 2020.

[41] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 39–56, May 2020.

[42] 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.

[43] C. A. Metzler, A. Maleki, and R. Baraniuk, "BM3D-PRGAMP: Compressive phase retrieval based on BM3D denoising," in *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 2504–2508.

[44] 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, Sep. 2016.

[45] A. Fletcher, S. Rangan, S. Sarkar, and P. Schniter, "Plug-in estimation in high-dimensional linear inverse problems: A rigorous analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, Montréal, Canada, Dec. 2018, pp. 7451–7460.

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

[47] S. A. Bigdeli, M. Jin, P. Favaro, and M. Zwicker, "Deep mean-shift priors for image restoration," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 763–772.

[48] G. Mataev, M. Elad, and P. Milanfar, "DeepRED: Deep image prior powered by RED," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Seoul, South Korea, Oct./Nov., 2019, pp. 1–10.

[49] Y. Chun and J. A. Fessler, "Deep BCD-Net using identical encoding-decoding CNN structures for iterative image recovery," in *Proc. IEEE 13th Image, Video, Multidimensional Signal Process. Workshop*, 2018, pp. 1–5.

[50] I. Y. Chun, Z. Huang, H. Lim, and J. Fessler, "Momentum-net: Fast and convergent iterative neural network for inverse problems," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2020, doi: 10.1109/TPAMI.2020.3012955.

[51] Y. Sun, J. Liu, and U. S. Kamilov, "Block coordinate regularization by denoising," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 8-14, 2019, pp. 380–390.

[52] Y. Sun, J. Liu, Y. Sun, B. Wohlberg, and U. S. Kamilov, "Async-RED: A provably convergent asynchronous block parallel stochastic method using deep denoising priors," in *Proc. Int. Conf. Learn. Representations*, Jul. 18-24, 2021.

[53] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative priors," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, Australia, 2017, pp. 537–546.

[54] V. Shah and C. Hegde, "Solving linear inverse problems using GAN priors: An algorithm with provable guarantees," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Calgary, AB, Canada, 2018, pp. 4609–4613.

[55] R. Hyder, V. Shah, C. Hegde, and M. S. Asif, "Alternating phase projected gradient descent with generative priors for solving compressive phase retrieval," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brighton, U.K., 2019, pp. 7705–7709.

[56] A. Raj, Y. Li, and Y. Bresler, "GAN-based projector for faster recovery in compressed sensing with convergence guarantees," in *Proc. IEEE Int. Conf. Comput. Vis.*, Seoul, South Korea, Oct./Nov., 2019, pp. 5601–5610.

[57] F. Latorre, A. Eftekhari, and V. Cevher, "Fast and provable ADMM for learning with generative priors," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, USA, Dec. 2019, pp. 12 027–12 039.

[58] B. Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 301–315, Jan. 2016.

[59] S. Ramani and J. A. Fessler, "A splitting-based iterative algorithm for accelerated statistical X-ray CT reconstruction," *IEEE Trans. Med. Imag.*, vol. 31, no. 3, pp. 677–688, Mar. 2012.

[60] M. Almeida and M. Figueiredo, "Deconvolving images with unknown boundaries using the alternating direction method of multipliers," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3074–3086, Aug. 2013.

[61] D. P. Bertsekas, "Incremental proximal methods for large scale convex optimization," *Math. Program. Ser. B*, vol. 129, no. 2, pp. 163–195, 2011.

[62] L. Tian, Z. Liu, L. Yeh, M. Chen, J. Zhong, and L. Waller, "Computational illumination for high-speed in vitro fourier ptychographic microscopy," *Optica*, vol. 2, no. 10, pp. 904–911, 2015.

[63] M. R. Kellman, E. Bostan, N. A. Repina, and L. Waller, "Physics-based learned design: Optimized coded-illumination for quantitative phase imaging," *IEEE Trans. Comput. Imag.*, vol. 5, no. 3, pp. 344–353, Sep. 2020.

[64] T. F. Chan, J. Shen, and H.-M. Zhou, "Total variation wavelet inpainting," *J. Math. Imag. Vis.*, vol. 25, no. 1, pp. 107–125, Jul. 2006.

[65] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, British Columbia, Canada, 2006, pp. 41–48.

[66] H. Huang and C. Ding, "Robust tensor factorization using R1 norm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[67] F. Nie, H. Huang, X. Cai, and C. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, British Columbia, Canada, Dec. 2010, pp. 1813–1821.

[68] C. Lu, J. Shi, and J. Jia, "Online robust dictionary learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 415–422.

[69] W. Jiang, F. Nie, and H. Huang, "Robust dictionary learning with capped l_1-norm," in *Proc. 24th Int. Conf. Artif. Intell.*, Buenos Aires, Argentina, 2015, pp. 3590–3596.

[70] U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, D. Psaltis, and M. Unser, "Isotropic inverse-problem approach for two-dimensional phase unwrapping," *J. Opt. Soc. Amer. A*, vol. 32, no. 6, pp. 1092–1100, Jun. 2015.

[71] A. Beck, *First-Order Methods in Optimization, ser. MOS-SIAM Series on Optimization*. SIAM, 2017, ch. The Proximal Operator, pp. 129–177.

[72] H. H. Bauschke, R. Goebel, Y. Lucet, and X. Wang, "The proximal average: Basic theory," *SIAM J. Optim.*, vol. 19, no. 2, pp. 766–785, 2008.

[73] Y. Yu, "Better approximation and faster algorithm using the proximal average," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, CA, USA, Dec. 2013, pp. 458–466.

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

[75] S. Boyd and L. Vandenberghe, "Subgradients," Class Notes for Convex Optimization II. Stanford Univ., Stanford, CA, USA, Apr. 2008. [Online]. Available: http://see.stanford.edu/materials/lsocoee364b/01-subgradients_notes.pdf

[76] A. Teodoro, J. M. Bioucas-Dias, and M. Figueiredo, "Scene-adapted plug-and-play algorithm with convergence guarantees," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, Tokyo, Japan, Sep. 2017, pp. 1–6.

[77] P. Nair, R. G. Gavaskar, and K. N. Chaudhury, "Fixed-point and objective convergenceof plug-and-play algorithms," *IEEE Trans. Comput. Imag.*, vol. 7, pp. 337–348, Mar. 2021.

[78] R. G. Gavaskar and K. N. Chaudhury, "Plug-and-play ISTA converges with Kernel denoisers," *IEEE Signal Process. Lett.*, vol. 27, pp. 610–614, Apr. 2020.

[79] M. Terris, A. Repetti, J.-C. Pesquet, and Y. Wiaux, "Building firmly nonexpansive convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain, 2020, pp. 8658–8662.

[80] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2018.

[81] M. Fazlyab, A. Robey, H.H.M. Marari, and G. Pappas, "Efficient and accurate estimation of lipschitz constants for deep neural networks," in *Proc Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2019, pp. 11 427–11 438.

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

[83] R. Ling, W. Tahir, H. Lin, H. Lee, and L. Tian, "High-throughput intensity diffraction tomography with a computational microscope," *Biomed. Opt. Exp.*, vol. 9, no. 5, pp. 2130–2141, May 2018.

[84] H. Sedghi, V. Gupta, and P. M. Long, "The singular values of convolutional layers," in *Proc. Int. Conf. Learn. Representations*, 2019.

[85] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE Int. Conf. Comput. Vis.*, Vancouver, Canada, 2001, pp. 416–423.

[86] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vis.*, vol. 20, no. 1, pp. 89–97, 2004.

[87] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 449–458.

[88] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Scalable convolutional neural network for image compressed sensing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 12 282–12 291.

[89] D. You, J. Xie, and J. Zhang, "ISTA-Net : Flexible Deep Unfolding Network for Compressive Sensing," in *2021 IEEE Int. Conf. on Multimedia and Expo*, pp. 1–6, 2021, doi: 10.1109/ICME51207.2021.9428249.

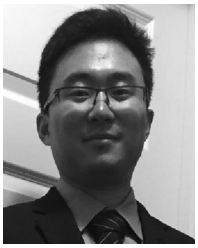
[90] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1828–1837.

[91] J. Liu, S. Asif, B. Wohlberg, and U. S. Kamilov, "Recovery Analysis for Plug-and-Play Priors Using the Restricted Eigenvalue Condition," 2021, *arXiv:2106.03668*.

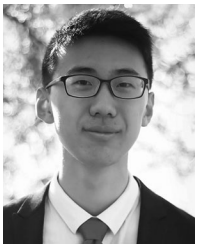
[92] A. Borji and L. Itti, "Cat2000: A large scale fixation dataset for boosting saliency research," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop on "Future of Datasets,"* 2015.

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

[94] 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.
[95] 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. Mach. Learn.*, Stockholm, Sweden, pp. 3501–3510, 2018.
[96] Z. Wu, Y. Sun, J. Liu, and U. S. Kamilov, "Online regularization by denoising with applications to phase retrieval," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 1–9.
[97] Z. Wu, Y. Sun, A. Matlock, J. Liu, L. Tian, and U. S. Kamilov, "SIMBA: Scalable inversion in optical tomography using deep denoising priors," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 6, pp. 1163–1175, Oct. 2020.
[98] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[99] U. S. Kamilov, "A parallel proximal algorithm for anisotropic total variation minimization," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 539–548, Feb. 2017.



Yu Sun (Graduate Student Member, IEEE) received the B.Eng. degree in electronics and information from Sichuan University, Chengdu, China, in 2015, and the M.S. degree in 2017 in data analytics and statistics from Washington University in St. Louis, St. Louis, MO, USA, where he is currently working toward the Ph.D. degree with the Computational Imaging Group. His research interests include computational imaging, machine learning, deep learning, and optimization.



Zihui Wu received the B.Sc. degree in computer science from Washington University in St. Louis (WUSTL), St. Louis, MO, USA. He is currently working toward the Ph.D. degree and Kortschak Scholar with the Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA. He is working with Professor Katherine L. Bouman on computational imaging, optimization, and machine learning. . He was with Professor Ulugbek S. Kamilov in the Computational Imaging Group, WUSTL.

Xiaojian Xu (Graduate Student Member, IEEE) received the B.Eng. degree in communication engineering and the M.Eng. degree in communication and information system from the University of Electronic Science and Technology of China, Chengdu, China, in 2014 and 2017, respectively. She is currently working toward the Ph.D. degree with the Computational Imaging Group. Her research interests include computational imaging, optimization, inverse problems, computer vision, deep learning, machine learning, and signal processing.

Brendt Wohlberg (Senior Member, IEEE) received the B.Sc. (Hons.) degree in applied mathematics, and the M.Sc. (applied science) and Ph.D. degrees in electrical engineering from the University of Cape Town, Cape Town, South Africa, in 1990, 1993, and 1996, respectively. He is currently a Staff Scientist with Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM, USA. His primary research interests include signal and image processing inverse problems and computational imaging. He was the co-recipient of the 2020 SIAM Activity Group on Imaging Science Best Paper Prize. He was an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING from 2010 to 2014, and for the IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING from 2015 to 2017, and was the Chair of the Computational Imaging Special Interest Group (currently the Computational Imaging Technical Committee) of the IEEE Signal Processing Society from 2015 to 2017. He is currently an Associate Editor for the *SIAM JOURNAL ON IMAGING SCIENCES*, and the Editor-in-Chief of the IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING.

Ulugbek S. Kamilov (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in communication systems, and the Ph.D. degree in electrical engineering from EPFL, Lausanne, Switzerland, in 2008, 2011, and 2015, respectively. He is currently an Assistant Professor and the Director of Computational Imaging Group with Washington University in St. Louis, St. Louis, MO, USA. From 2015 to 2017, he was a Research Scientist with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. His primary research focuses on computational imaging, which include biomedical imaging, machine learning, and large-scale optimization. He was the recipient of the NSF CAREER Award and the IEEE Signal Processing Society's 2017 Best Paper Award. His Ph.D. thesis was selected as a finalist for the EPFL Doctorate Award in 2016. He was an Associate Editor for the IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING (2019–present), *Biological Imaging* (2020–present), and on IEEE Signal Processing Society's Computational Imaging Technical Committee (2016–present). He was a Plenary Speaker at iTWIST 2018 and the Program Co-Chair for SampTA 2021 and BASP 2022.