Scalable Plug-and-Play ADMM with Convergence Guarantees

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

Zihui Wu
Washington University in St. Louis
ray.wu@wustl.edu

Brendt Wohlberg
Los Alamos National Laboratory
brendt@ieee.org

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

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 large-scale datasets. 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.

1 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 [17–23]. Despite this progress, current PnP algorithms are not practical for addressing large-scale problems due to their computation time and memory requirements. To the best of our knowledge, the only prior work on developing PnP algorithms that are suitable for large-scale problems 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 solving large-scale inverse problems. 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

Preprint. Under review.

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.

All proofs and some technical details have been omitted due to space restrictions, and are included in the supplement, which also provides additional background and additional simulations.

2 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 common to formulate the solution to this estimation as an optimization problem

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

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 and $\tau > 0$ is the regularization parameter. Common applications of (1) include sparse vector recovery in compressive sensing [24, 25], image restoration using total variation (TV) [26], and low-rank matrix completion [27].

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

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

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

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

where $\gamma > 0$ is the penalty parameter [11]. ADMM relies on the *proximal operator*

$$\mathsf{prox}_{\tau h}(\boldsymbol{z}) := \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^n} \left\{ \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \tau h(\boldsymbol{x}) \right\}, \quad \tau > 0, \quad \boldsymbol{z} \in \mathbb{R}^n, \tag{3}$$

which is well-defined 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

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

by setting $h(\boldsymbol{x}) = -\log(p_{\boldsymbol{x}}(\boldsymbol{x}))$. This perspective inspired the development of PnP [1, 2], where the proximal operator is simply replaced by a more general denoiser $\mathsf{D} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such as BM3D [28] or DnCNN [29]. For example, the widely used PnP-ADMM can be summarized as

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

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

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

where, in analogy to $\tau > 0$ 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] and spurred a great deal of theoretical work on PnP algorithms [17–23].

An elegant fixed-point convergence analysis of PnP-ADMM was recently 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

$$\mathsf{P} := \frac{1}{2}\mathsf{I} + \frac{1}{2}(2\mathsf{G} - \mathsf{I})(2\mathsf{D}_\sigma - \mathsf{I}) \quad \text{with} \quad \mathsf{G} := \mathsf{prox}_{\gamma g}, \tag{6}$$

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].

Scalable optimization algorithms have become increasingly important in the context of large-scale problems arising in machine learning and data science [30]. Stochastic and online optimization techniques have been investigated for traditional ADMM [31–35], where $\mathsf{prox}_{\gamma g}$ 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 pressing need to understand its theoretical properties and ability to address large-scale imaging applications.

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 [36–39]). 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)* [40–43]. The key difference of PnP from AMP is that it does not assume random measurement operators. *Regularization by denoising (RED)*, and the closely related *deep mean-shift priors*, rely on the denoiser to specify an explicit regularizer that has a simple gradient [44–47]. PnP does not assume existence of such an objective, instead interpreting solutions as equilibrum points balancing the data-fit and the prior [19]. Finally, a recent line of work has investigated the recovery and convergence guarantees for priors specified by *generative adversarial networks (GANs)* [48–52]. 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 greatly 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.

3 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

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

We thus consider data-fidelity terms of the form

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

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

PnP-ADMM is often impractical when b is very large due to the complexity of computing $\mathsf{prox}_{\gamma g}$. IPA extends stochastic variants of traditional ADMM [31–35] by integrating denoisers D_σ that are *not* associated with any regularizer 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. 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 (see details in Supplement A).

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 $k = 1, 2, 3, \dots$ do
    Choose an index $i_k \in \{1, \dots, b\}$
    $\boldsymbol{z}^k \leftarrow \mathsf{G}_{i_k}(\boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1})$ where $\mathsf{G}_{i_k}(\boldsymbol{z}) := \mathsf{prox}_{\gamma g_{i_k}}(\boldsymbol{z})$    ▷ impose data-consistency
    $\boldsymbol{x}^k \leftarrow \mathsf{D}_\sigma(\boldsymbol{z}^k - \boldsymbol{s}^{k-1})$    ▷ impose prior knowledge
    $\boldsymbol{s}^k \leftarrow \boldsymbol{s}^{k-1} + \boldsymbol{x}^k - \boldsymbol{z}^k$
end for
```

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 $\{1, \dots, b\}$. An alternative would be to proceed in epochs of b consecutive iterations, where at the start of each epoch the set $\{1, \dots, b\}$ is reshuffled, and i_k is selected from this ordered set [53]. 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 [54, 55].

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. Thus, the maximal benefit of IPA is expected for problems in which G_i is efficient to evaluate. This is a case for a large number of functions commonly used in computational imaging, compressive sensing, and machine learning (see the extensive discussion on proximal operators in [56]).

Let us discuss two widely used scenarios. 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

$$\mathsf{G}_i(\boldsymbol{z}) = \mathsf{prox}_{\gamma g_i}(\boldsymbol{z}) = \left(\mathbf{I} + \gamma \boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{A}_i\right)^{-1}\left(\boldsymbol{z} + \gamma \boldsymbol{A}_i^{\mathsf{T}}\boldsymbol{y}\right), \quad \gamma > 0, \quad \boldsymbol{z} \in \mathbb{R}^n. \tag{8}$$

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, 57–59]. 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 [56]. We numerically evaluate the effectiveness of IPA on both ℓ_1- and ℓ_2-norm data-fidelity terms and deep neural net priors in Section 5.

4 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.

4.1 Fixed Point Interpretation

IPA 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. The full technical exposition of the discussion here can be found in Supplement D.

Consider the following set-valued operator

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

where ∂g is the subdifferential of the data-fidelity term and $\mathsf{D}_\sigma^{-1}(\boldsymbol{x}) \coloneqq \{\boldsymbol{z} \in \mathbb{R}^n : \boldsymbol{x} = \mathsf{D}_\sigma(\boldsymbol{z})\}$ is the inverse operator of the denoiser D_σ. Note that this inverse operator exists even when D_σ is not one-to-one [8, 60]. 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

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

Consider the following two sets

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

where $\mathsf{zer}(\partial g)$ is the set of all critical points of the data-fidelity term and $\mathsf{fix}(\mathsf{D}_\sigma)$ 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 $\boldsymbol{x}^* \in \mathsf{zer}(\partial g) \cap \mathsf{fix}(\mathsf{D}_\sigma)$, then $\boldsymbol{x}^* \in \mathsf{zer}(\mathsf{T})$, 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 $\mathsf{zer}(\partial g) \cap \mathsf{fix}(\mathsf{D}_\sigma) = \varnothing$, then $\boldsymbol{x}^* \in \mathsf{zer}(\mathsf{T})$ 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 through different mechanisms. Hence, the best performance is obtained by jointly tuning both parameters for a given experimental setting. In the special case of $\mathsf{D}_\sigma = \mathsf{prox}_{\gamma h}$ with $\gamma = \sigma^2$, we have

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

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

4.2 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 $L = \mathsf{max}\{L_1, \dots, L_b\}$.*

This assumption is commonly adopted in nonsmooth optimization and is equivalent to existence of a global upper bound on subgradients [32, 61, 62]. 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 the denoiser 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 supplement. 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, which implies that the proximal operator automatically satisfies Assumption 2 [3].

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 [29]. Prior work has also shown that Lipschitz constrained residual networks yield excellent performance without sacrificing stable convergence [23, 46]. Additionally, there has recently been an explosion of techniques for training Lipschitz constrained and firmly nonexpansive deep neural nets [23, 63–65].

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

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

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 $[0, 255]^n \subset \mathbb{R}^n$.

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 traditional PnP-ADMM can be interpreted as algorithms for computing an element in $\mathsf{zer}(\mathsf{S})$, which is equivalent to finding an element of $\mathsf{zer}(\mathsf{T})$ (see details in Supplement D).

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 fixed penalty parameter $\gamma > 0$. Then, the sequence $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ satisfies*

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

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 fixed penalty parameter $\gamma > 0$. Then, the sequence $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ satisfies*

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

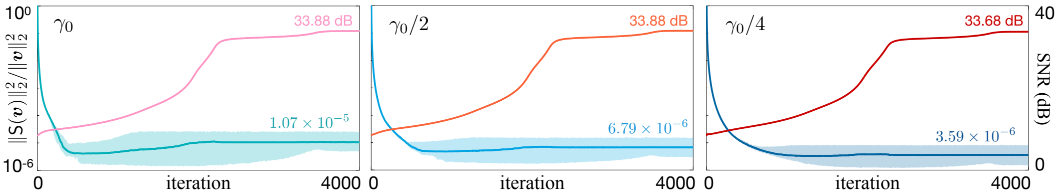

Figure 1: *Illustration of the influence of the penalty parameter $\gamma > 0$ on the convergence of IPA for a DnCNN prior. The average normalized distance to* $\mathsf{zer}(\mathsf{S})$ *and SNR (dB) are plotted against the iteration number with the shaded areas representing the range of values attained 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.*

Both proofs are provided in the supplement. The proof of Theorem 2 is a minor 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 and D_σ is nonexpansive, the Krasnosel'skii-Mann theorem (see Section 5.2 in [66]) directly implies that $\boldsymbol{v}^t \to \mathsf{zer}(\mathsf{S})$ and $\boldsymbol{x}^t = \mathsf{D}_\sigma(\boldsymbol{v}^t) \to \mathsf{zer}(\mathsf{T})$.

Theorem 1 establishes that IPA approximates the solution obtained by the full 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 5 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.

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

$$\mathbb{E}\left[\|\boldsymbol{x}^t - \boldsymbol{x}^*\|_2\right] \leq \eta^t(2R + 4\gamma L) + (4\gamma L)/(1-\eta), \quad 0 < \eta < 1, \tag{13}$$

which establishes a linear convergence to $\mathsf{zer}(\mathsf{T})$ up to an error term. A proof of (13) is provided in the supplement. As corroborated by our simulations in Section 5, the actual convergence of IPA holds even more broadly than suggested by both sets of sufficient conditions. This motivates further analysis of IPA under more relaxed assumptions that we leave to future work.

5 Numerical Validation

Recent work has shown the excellent performance of PnP algorithms 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* [67].

Our deep neural net prior is based on the DnCNN architecture [29], with its batch normalization layers removed for controlling the Lipschitz constant of the network via spectral normalization [68] (see details in Supplement G.1). We train a nonexpansive residual network R_σ by predicting the noise residual from its noisy input. This means that R_σ satisfies the necessary condition for firm nonexpansiveness of D_σ. The training data is generated by adding AWGN to the images from the BSD400 dataset [69]. 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 in each experiment.

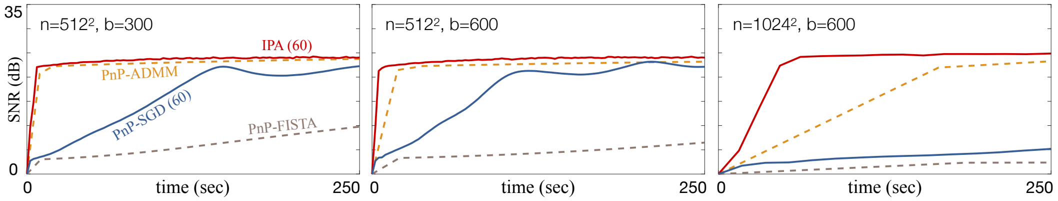

Figure 2: *Illustration of scalability of IPA and several widely used PnP algorithms on problems of different sizes. The parameters n and b denote the image size and the number of acquired intensity images, respectively. The average SNR is plotted against time in seconds. Both IPA and PnP-SGD use random 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 1: 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**)

5.1 Integration of Non-smooth Data-Fidelity Terms and Pre-Trained Deep Priors

We first validate the effectiveness of Theorem 1 for 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 noise motivates the usage of the ℓ_1-norm $g(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_1$, since it can effectively mitigate outliers. 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 [26, 70].

We set the measurement ratio to be approximately $m/n = 0.7$ with AWGN of standard deviation 5. Twelve standard images from *Set 12* 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 $\mathsf{zer}(\mathsf{S})$ for smaller values of γ, which is consistent with our theoretical analysis. Specifically, the final accuracy improves approximately 3× (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 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.

Table 2: 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**

5.2 Scalability in Large-scale Optical Tomography

We now discuss the scalability of IPA on intensity diffraction tomography, which is a data intensive computational imaging modality [67]. 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 $[\boldsymbol{A}_1, \dots, \boldsymbol{A}_b]^\mathsf{T}$, 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.

In the simulation, we follow the experimental setup in [67] under AWGN corresponding to an input SNR of 20 dB. We select six images from the CAT2000 dataset [71] as our test examples, each cropped to n pixels. We assume real permittivity functions, but still consider complex valued measurement operator $\boldsymbol{A}$ that accounts for both absorption and phase [67]. Due to the large size of data, we process the measurements in epochs using minibatches of size 60 (see also Supplement G.2).

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 $b \in \{300, 600\}$. The final values of SNR as well as the total runtimes are summarized in Table 1. 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 1 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 1. 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 2 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 (see also Supplement G.2 for additional examples). 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.

6 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 IPA can approximate PnP-ADMM to a desired precision by controlling the penalty parameter. Third, our simulations highlight the effectiveness of IPA for nonsmooth data-fidelity terms and deep neural net priors, as well as its scalability to large-scale imaging. 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.

Broader Impact

This work is expected to impact the area of computational imaging with potential applications to computational microscopy, computerized tomography, medical imaging, and image restoration. There is a growing need in computational imaging to deal with noisy and incomplete measurements by integrating multiple information sources, including physical information describing the imaging instrument and learned information characterizing the statistical distribution of the desired image. The ability to solve large-scale computational imaging problems has the potential to enable new technological advances in 3D (space), 4D (space + time), or 5D (space, time, and spectrum) applications. These advances might lead to new imaging tools for diagnosing health conditions, understanding biological processes, or inferring properties of complex materials.

Traditionally, imaging relied on linear models and fixed transforms (filtered back projection, wavelet transform) that are relatively straightforward to understand. Learning based methods, including our algorithm, have the potential to enable new technological capabilities; yet, they also come with a downside of being much more complex. Their usage might thus lead to unexpected outcomes and surprising results when used by non-experts. While we aim to use our method to enable positive contributions to humanity, one can also imagine nonethical usage of large-scale imaging technology. This work focuses on imaging using large-scale algorithms with learned priors, but it might be adopted within broader data science, which might lead to broader impacts that we have not anticipated.

Acknowledgments and Disclosure of Funding

Research presented in this article was supported by NSF award CCF-1813910 and by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20200061DR.

References

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

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

[3] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, 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, August 2003.

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

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

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

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

[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, September 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, August 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," *Foundations and Trends in Machine Learning*, 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. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[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. Patt. Anal. and Machine 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. Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019, pp. 1671–1681.

[15] R. Ahmad, C. A. Bouman, G. T. Buzzard, S. Chan, S. Liu, E. T. Reehorst, and P. Schniter, "Plug-and-play methods for magnetic resonance imaging: Using denoisers for image recovery," *IEEE Signal Processing Magazine*, vol. 37, no. 1, pp. 105–116, 2020.

[16] K. Wei, A. Aviles-Rivero, J. Liang, Y. Fu, C.-B. Schnlieb, and H. Huang, "Tuning-free plug-and-play proximal algorithm for inverse imaging problems," in *Proc. 37th Int. Conf. Machine Learning (ICML)*, 2020, arXiv:2002.09611.

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

[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. Comp. Vis. (ICCV)*, Venice, Italy, Oct. 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. Imaging 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. Imaging*, vol. 5, no. 3, pp. 395–408, Sept. 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, 2019.

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

[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. Machine Learning (ICML)*, Long Beach, CA, USA, June 2019, pp. 5546–5557.

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

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

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

[27] 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.

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

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

[30] 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.

[31] H. Wang and A. Banerjee, "Online alternating direction method," in *Proc. 29th Int. Conf. Machine Learning (ICML)*, Edinburgh, Scotland, UK, June 26-July 1, 2012, pp. 1699–1706.

[32] H. Ouyang, N. He, L. Q. Tran, and A. Gray, “Stochastic alternating direction method of multipliers,” in *Proc. 30th Int. Conf. Machine Learning (ICML)*, Atlanta, GA, USA, 16-21 June, 2013, pp. 80–88.

[33] T. Suzuki, “Dual averaging and proximal gradient descent for online alternating direction multiplier method,” in *Proc. 30th Int. Conf. Machine Learning (ICML)*, Atlanta, GA, USA, June 2013, pp. 392–400.

[34] W. Zhong and J. Kwok, “Fast stochastic alternating direction method of multipliers,” in *Proc. 31th Int. Conf. Machine Learning (ICML)*, Bejing, China, Jun 22-24, 2014, pp. 46–54.

[35] F. Huang, S. Chen, and H. Huang, “Faster stochastic alternating direction method of multipliers for nonconvex optimization,” in *Proc. 36th Int. Conf. Machine Learning (ICML)*, Long Beach, CA, USA, June 10-15, 2019, pp. 2839–2848.

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

[37] 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.

[38] F. Knoll, K. Hammernik, C. Zhang, S. Moeller, T. Pock, D. K. Sodickson, and M. Akcakaya, “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.

[39] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, “Deep learning techniques for inverse problems in imaging,” 2020, arXiv:2005.06001.

[40] 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.

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

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

[43] A. Fletcher, S. Rangan, S. Sarkar, and P. Schniter, “Plug-in estimation in high-dimensional linear inverse problems: A rigorous analysis,” in *Proc. Advances in Neural Information Processing Systems 32*, Montréal, Canada, Dec 3-8, 2018, pp. 7451–7460.

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

[45] S. A. Bigdeli, M. Jin, P. Favaro, and M. Zwicker, “Deep mean-shift priors for image restoration,” in *Proc. Advances in Neural Information Processing Systems 31*, Long Beach, CA, USA, Dec 4-9, 2017, pp. 763–772.

[46] Y. Sun, J. Liu, and U. S. Kamilov, “Block coordinate regularization by denoising,” in *Proc. Advances in Neural Information Processing Systems 33*, Vancouver, BC, Canada, December 8-14, 2019, pp. 382–392.

[47] G. Mataev, M. Elad, and P. Milanfar, “DeepRED: Deep image prior powered by RED,” in *Proc. IEEE Int. Conf. Comp. Vis. Workshops (ICCVW)*, Seoul, South Korea, Oct 27-Nov 2, 2019, pp. 1–10.

[48] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, “Compressed sensing using generative priors,” in *Proc. 34th Int. Conf. Machine Learning (ICML)*, Sydney, Australia, Aug. 2017, pp. 537–546.

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

[50] 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. Acoustics, Speech and Signal Process.*, Brighton, UK, May 2019, pp. 7705–7709.

[51] A. Raj, Y. Li, and Y. Bresler, “GAN-based projector for faster recovery in compressed sensing with convergence guarantees,” in *Proc. IEEE Int. Conf. Comp. Vis. (ICCV)*, Seoul, South Korea, Oct 27-Nov 2, 2019, pp. 5601–5610.

[52] F. Latorre, A. Eftekhari, and V. Cevher, “Fast and provable ADMM for learning with generative priors,” in *Advances in Neural Information Processing Systems 33*, Vancouver, BC, USA, December 8-14, 2019, pp. 12027–12039.

[53] D. P. Bertsekas, “Incremental proximal methods for large scale convex optimization,” *Math. Program. Ser. B*, vol. 129, pp. 163–195, 2011.

[54] 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.

[55] 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, 2020.

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

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

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

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

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

[61] S. Boyd and L. Vandenberghe, "Subgradients," April 2008, Class notes for Convex Optimization II. `http://see.stanford.edu/materials/lsocoee364b/01-subgradients_notes.pdf`.

[62] Y.-L. Yu, "Better approximation and faster algorithm using the proximal average," in *Proc. Advances in Neural Information Processing Systems 26*, 2013.

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

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

[65] M. Fazlyab, A. Robey, Hassani. H., M. Marari, and G. Pappas, "Efficient and accurate estimation of Lipschitz constants for deep neural networks," in *Proc. Advances in Neural Information Processing Systems 33*, Vancouver, BC, Canada, Dec. 2019, pp. 11427–11438.

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

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

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

[69] 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. Comp. Vis. (ICCV)*, Vancouver, Canada, July 7-14, 2001, pp. 416–423.

[70] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1, pp. 89–97, 2004.

[71] A. Borji and L. Itti, "Cat2000: A large scale fixation dataset for boosting saliency research," *Comput. Vis. Patt. Recong. (CVPR) 2015 Workshop on "Future of Datasets"*, 2015.

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

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

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

[76] R. T. Rockafellar, *Convex Analysis*, chapter Conjugate Saddle-Functions and Minimax Theorems, pp. 388–398, Princeton Univ. Press, Princeton, NJ, 1970.

[77] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

Supplementary Material for "Scalable Plug-and-Play ADMM with Convergence Guarantees"

We adopt the monotone operator theory [60, 66] for a unified analysis of IPA. In Supplement A, we discuss a minibatch variant of IPA that enables parallel evaluation of several component proximal operators. In Supplement B, we present the convergence analysis of IPA. In Supplement C, we analyze the convergence of the algorithm for strongly convex data-fidelity terms and contractive denoisers. In Supplement D, we discuss interpretation of IPA's fixed-points from the perspective of monotone operator theory. For completeness, in Supplement E, we discuss the convergence results for traditional PnP-ADMM [23]. In Supplement F, we provide the background material used in our analysis. In Supplement G, we provide additional technical details, omitted from the main paper due to space, such as the details on our deep neural net architecture and results of additional simulations.

For the sake of simplicity, this supplement uses $\|\cdot\|$ 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 Minibatch Implementation of IPA

Algorithm 2 Incremental Plug-and-Play ADMM (IPA) *(Minibatch Version)*

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

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. The convergence analysis in Theorem 1 can be easily extended to minibatch IPA with a straightforward extension of Lemma 1 in Supplement B.2 to several indices, and by following the steps of the main proof in Supplement B.1.

Minibatch IPA is related to the *proximal average* approximation of $\mathsf{G} = \mathsf{prox}_{\gamma g}$ [62, 72]

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

When Assumption 1 is satisfied, then the approximation error is bounded for any $\boldsymbol{x} \in \mathbb{R}^n$ as [62]

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

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 $\gamma > 0$.

B 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 [66]) 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 here extends these results to IPA by providing an explicit upper bound on the convergence of IPA. In Section B.1, we present the main steps of the proof, while in Section B.2 we prove two technical lemmas useful for our analysis.

B.1 Proof of Theorem 1

Supplement D.3 establishes that S defined in (10) is firmly nonexpansive. Consider any $\boldsymbol{v}^* \in \mathsf{zer}(\mathsf{S})$ and any $\boldsymbol{v} \in \mathbb{R}^n$, then we have

$$\|\boldsymbol{v} - \boldsymbol{v}^* - \mathsf{S}\boldsymbol{v}\|^2 = \|\boldsymbol{v} - \boldsymbol{v}^*\| - 2(\mathsf{S}\boldsymbol{v} - \mathsf{S}\boldsymbol{v}^*)^\mathsf{T}(\boldsymbol{v} - \boldsymbol{v}^*) + \|\mathsf{S}\boldsymbol{v}\|^2 \leq \|\boldsymbol{v} - \boldsymbol{v}^*\|^2 - \|\mathsf{S}\boldsymbol{v}\|^2, \quad (14)$$

where we used the firm nonexpansiveness of S and $\mathsf{S}\boldsymbol{x}^* = \boldsymbol{0}$. 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 $k \geq 1$

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

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

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

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

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

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

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

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

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

The final result is obtained by noting that

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

B.2 Lemmas Useful for the Proof of Theorem 1

This section presents two technical lemmas used in our analysis in Section B.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*

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

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

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

and

$$\mathsf{G}\boldsymbol{x} = \mathsf{prox}_{\gamma g}(\boldsymbol{x}) = \boldsymbol{x} - \gamma \boldsymbol{g}(\boldsymbol{z}) \quad \text{such that} \quad \boldsymbol{g}(\boldsymbol{z}) = \frac{1}{b}\sum_{i=1}^{b}\boldsymbol{g}_i(\boldsymbol{z}) \in \partial g(\boldsymbol{z}),$$

where we used Proposition 7 in Supplement F.2. By using the bound on all the subgradients (due to Assumption 1 and Proposition 8 in Supplement F.2), we obtain

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

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*

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

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

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

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

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

Thus, by using Assumption 3 and the bounds on all the subgradients (due to Assumption 1 and Proposition 8 in Supplement F.2), we obtain

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

□

C 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 $M = \mathsf{min}\{M_1, \dots, M_b\}$ and the largest Lipschitz constant as $L = \mathsf{max}\{L_1, \dots, L_b\}$.*

This assumption further restricts Assumption 1 in the main paper 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 in the main paper 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 [64]. 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

$$\|(\mathsf{I}-\mathsf{S})\boldsymbol{x}-(\mathsf{I}-\mathsf{S})\boldsymbol{y}\| \leq \eta\|\boldsymbol{x}-\boldsymbol{y}\| \quad \text{with} \quad \eta \,:=\, \left(\frac{1+\epsilon+\epsilon\gamma M+2\epsilon^2\gamma M}{1+\alpha M+2\epsilon\gamma M}\right),$$

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 Supplement B, the sequence $\boldsymbol{v}^k = \boldsymbol{z}^k - \boldsymbol{s}^{k-1}$ can be written as

$$\boldsymbol{v}^k = \boldsymbol{v}^{k-1} - \mathsf{S}_{i_k}(\boldsymbol{v}^{k-1}) \quad \text{with} \quad \mathsf{S}_{i_k} \,:=\, \mathsf{D}-\mathsf{G}_{i_k}(2\mathsf{D}-\mathsf{I}). \tag{17}$$

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

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

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 Supplement B.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 Supplement B.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

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

This concludes the proof. □

D 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].

D.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

$$\boldsymbol{x}^* = \mathsf{G}(\boldsymbol{x}^*+\boldsymbol{s}^*) \tag{18a}$$

$$\boldsymbol{x}^* = \mathsf{D}(\boldsymbol{x}^*-\boldsymbol{s}^*), \tag{18b}$$

with $\boldsymbol{x}^*=\boldsymbol{z}^*$, where $\mathsf{G}=\mathsf{prox}_{\gamma g}$. 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 [60]). Then, from the definition of D^{-1} and optimality conditions of the proximal operator, we can equivalently rewrite (18) as follows

$$\boldsymbol{s}^*\in\gamma\partial g(\boldsymbol{x}^*)$$

$$-\boldsymbol{s}^*\in\mathsf{D}^{-1}(\boldsymbol{x}^*)-\boldsymbol{x}^*.$$

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

$$\mathbf{0}\in\mathsf{T}(\boldsymbol{x}^*) \,:=\, \gamma\partial g(\boldsymbol{x}^*)+(\mathsf{D}^{-1}(\boldsymbol{x}^*)-\boldsymbol{x}^*). \tag{19}$$

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.

D.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

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

which was defined in (10) of the main paper. Note that from (18), 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

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

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

D.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 F.2 imply that D and G are firmly nonexpansive. Then, Proposition 4 in Supplement F.1 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

$$(\mathsf{I} - \mathsf{S}) = \frac{1}{2}\mathsf{I} + \frac{1}{2}(2\mathsf{G} - \mathsf{I})(2\mathsf{D} - \mathsf{I}) \tag{20}$$

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

E 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 the main paper.

E.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 following DRS (left) and ADMM (right) sqeuences

$$\begin{cases} \boldsymbol{x}^{k-1} = \mathsf{D}(\boldsymbol{v}^{k-1}) \\ \boldsymbol{z}^k = \mathsf{G}(2\boldsymbol{x}^{k-1} - \boldsymbol{v}^{k-1}) \\ \boldsymbol{v}^k = \boldsymbol{v}^{k-1} + \boldsymbol{z}^k - \boldsymbol{x}^{k-1} \end{cases} \quad \Leftrightarrow \quad \begin{cases} \boldsymbol{z}^k = \mathsf{G}(\boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1}) \\ \boldsymbol{x}^k = \mathsf{D}(\boldsymbol{z}^k - \boldsymbol{s}^{k-1}) \\ \boldsymbol{s}^k = \boldsymbol{s}^{k-1} + \boldsymbol{x}^k - \boldsymbol{z}^k, \end{cases} \tag{21}$$

where $\mathsf{G} := \mathsf{prox}_{\gamma g}$ 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

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

E.2 Convergence Analysis of PnP-DRS and PnP-ADMM

It was established in Supplement D.3 that S defined in (10) of the main paper is firmly nonexpansive.

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

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

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 Supplement B.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 [66]) implies that $\boldsymbol{v}^t \to \mathsf{zer}(\mathsf{S})$. Then, from continuity of D, we have that $\boldsymbol{x}^t = \mathsf{D}(\boldsymbol{v}^t) \to \mathsf{zer}(\mathsf{T})$ (see also Supplement D.2). This completes the proof.

F Background material

This section summarizes well-known results from the optimization literature that can be found in different forms in standard textbooks [66, 73–75].

F.1 Properties of Monotone Operators

Definition 1. *An operator T is Lipschitz continuous with constant $\lambda > 0$ if*

$$\|\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y}\| \leq \lambda\|\boldsymbol{x} - \boldsymbol{y}\|, \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

When $\lambda = 1$, we say that T is nonexpansive. When $\lambda < 1$, we say that T is a contraction.

Definition 2. *T is monotone if*

$$(\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) \geq 0, \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

We say that it is strongly monotone or coercive with parameter $\mu > 0$ if

$$(\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) \geq \mu\|\boldsymbol{x} - \boldsymbol{y}\|^2, \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

Definition 3. *T is cocoercive with constant $\beta > 0$ if*

$$(\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y})^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{y}) \geq \beta\|\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y}\|^2, \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

When $\beta = 1$, we say that T is firmly nonexpansive.

The following results are derived from the definition above.

Proposition 1. *Consider $\mathsf{R} = \mathsf{I} - \mathsf{T}$ where $\mathsf{T} : \mathbb{R}^n \to \mathbb{R}^n$.*

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

Proof. First suppose that R is $1/2$ cocoercive. Let $\boldsymbol{h} := \boldsymbol{x} - \boldsymbol{y}$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$. We then have

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

We also have that

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

By combining these two and simplifying the expression

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

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

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

T *is Lipschitz continuous with constant* $\lambda < 1 \;\Rightarrow\; \mathsf{R}$ *is* $(1-\lambda)$*-strongly monotone.*

Proof. By using the Cauchy-Schwarz inequality, we have for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$

$$\begin{aligned}(\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y})^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}) &= \|\boldsymbol{x} - \boldsymbol{y}\|^2 - (\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y})^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}) \\ &\geq \|\boldsymbol{x} - \boldsymbol{y}\|^2 - \|\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y}\|\|\boldsymbol{x} - \boldsymbol{y}\| \\ &\geq \|\boldsymbol{x} - \boldsymbol{y}\|^2 - \lambda\|\boldsymbol{x} - \boldsymbol{y}\|^2 \geq (1-\lambda)\|\boldsymbol{x} - \boldsymbol{y}\|^2.\end{aligned}$$

□

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

The following characterization is often convenient.

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

(a) T *is* α*-averaged*

(b) $(1 - 1/\alpha)\mathsf{I} + (1/\alpha)\mathsf{T}$ *is nonexpansive*

(c) $\|\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y}\|^2 \leq \|\boldsymbol{x} - \boldsymbol{y}\|^2 - \left(\frac{1-\alpha}{\alpha}\right)\|\mathsf{R}\boldsymbol{x} - \mathsf{R}\boldsymbol{y}\|^2$*,* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$.

Proof. See Proposition 4.35 in [66]. □

Proposition 4. *Consider* $\mathsf{T} : \mathbb{R}^n \to \mathbb{R}^n$ *and* $\beta > 0$*. Then, the following are equivalent*

(a) T *is* β*-cocoercive*

(b) $\beta\mathsf{T}$ *is firmly nonexpansive*

(c) $\mathsf{I} - \beta\mathsf{T}$ *is firmly nonexpansive.*

(d) $\beta\mathsf{T}$ *is* $(1/2)$*-averaged.*

(e) $\mathsf{I} - 2\beta\mathsf{T}$ *is nonexpansive.*

Proof. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, let $\boldsymbol{h} := \boldsymbol{x} - \boldsymbol{y}$. The equivalence between (a) and (b) is readily observed by defining $\mathsf{P} := \beta\mathsf{T}$ and noting that

$$\begin{aligned}(\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y})^\mathsf{T}\boldsymbol{h} &= \beta(\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y})^\mathsf{T}\boldsymbol{h} \\ \text{and} \quad \|\mathsf{P}\boldsymbol{x} - \mathsf{P}\boldsymbol{y}\|^2 &= \beta^2\|\mathsf{T}\boldsymbol{x} - \mathsf{T}\boldsymbol{y}\|.\end{aligned} \tag{23}$$

Define $\mathsf{R} := \mathsf{I} - \mathsf{P}$ and suppose (b) is true, then

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

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

The equivalence of (b) and (d) can be seen by noting that

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

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

F.2 Convex functions, subdifferentials, and proximal operators

Proposition 5. *Let f be a proper, closed, and convex function. Then, ∂f is a monotone operator*

$$(\boldsymbol{g} - \boldsymbol{h})^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}) \geq 0, \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, \quad \boldsymbol{g} \in \partial f(\boldsymbol{x}), \quad \boldsymbol{h} \in \partial f(\boldsymbol{y}).$$

If f is strongly convex with constant $\mu > 0$, then ∂f is strongly monotone with the same constant.

$$(\boldsymbol{g} - \boldsymbol{h})^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}) \geq \mu\|\boldsymbol{x} - \boldsymbol{y}\|^2, \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, \quad \boldsymbol{g} \in \partial f(\boldsymbol{x}), \quad \boldsymbol{h} \in \partial f(\boldsymbol{y}).$$

Proof. Consider a strongly convex function f with a constant $\mu \geq 0$. Then, we have that

$$\begin{cases} f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}^\mathsf{T}(\boldsymbol{y} - \boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 \\ f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \boldsymbol{h}^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}) + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 \end{cases} \quad \Rightarrow \quad (\boldsymbol{g} - \boldsymbol{h})^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}) \geq \mu\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

The proof for a weakly convex f is obtained by considering $\mu = 0$ in the inequalities above. □

It is well-known that the proximal operator is firmly nonexpansive.

Proposition 6. *Proximal operator $\mathsf{prox}_{\gamma f}$ of a proper, closed, and convex f is firmly nonexpansive.*

Proof. Denote with $\boldsymbol{x}_1 = \mathsf{G}\boldsymbol{z}_1 = \mathsf{prox}_{\gamma f}(\boldsymbol{z}_1)$ and $\boldsymbol{x}_2 = \mathsf{G}\boldsymbol{z}_2 = \mathsf{prox}_{\gamma f}(\boldsymbol{z}_2)$, then

$$\begin{aligned}
&\begin{cases} (\boldsymbol{z}_1 - \boldsymbol{x}_1) \in \gamma\partial f(\boldsymbol{x}_1) \\ (\boldsymbol{z}_2 - \boldsymbol{x}_2) \in \gamma\partial f(\boldsymbol{x}_2) \end{cases} \quad &&\Rightarrow \quad (\boldsymbol{z}_1 - \boldsymbol{x}_1 - \boldsymbol{z}_2 + \boldsymbol{x}_2)^\mathsf{T}(\boldsymbol{x}_1 - \boldsymbol{x}_2) \geq 0 \\
& &&\Rightarrow \quad (\mathsf{G}\boldsymbol{z}_1 - \mathsf{G}\boldsymbol{z}_2)^\mathsf{T}(\boldsymbol{z}_1 - \boldsymbol{z}_2) \geq \|\mathsf{G}\boldsymbol{z}_1 - \mathsf{G}\boldsymbol{z}_2\|^2
\end{aligned}$$

□

The following proposition is sometimes referred to as *Moreau-Rockafellar theorem*. It establishes that for functions defined over all of $\mathbb{R}^n$, we have that $\partial f = \partial f_1 + \cdots + \partial f_m$.

Proposition 7. *Consider $f = f_1 + \cdots + f_m$, where $f_1, \dots, f_m$ are proper, closed, and convex functions on $\mathbb{R}^n$. Then*

$$\partial f_1(\boldsymbol{x}) + \cdots + \partial f_m(\boldsymbol{x}) \subset \partial f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^n$$

Moreover, suppose that convex sets $\mathsf{ri}(\mathsf{dom}\, f_i)$ have a point in common, then we also have

$$\partial f(\boldsymbol{x}) \subset \partial f_1(\boldsymbol{x}) + \cdots + \partial f_m(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^n.$$

Proof. See Theorem 23.8 in [76]. □

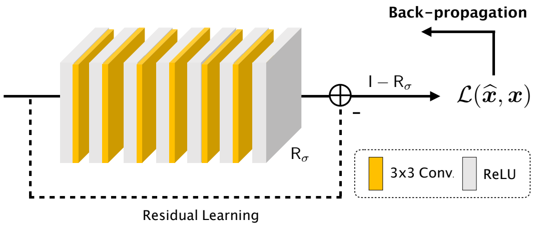

Figure 3: *Illustration of the architecture of DnCNN used in all experiments. Vectors $\widehat{\boldsymbol{x}}$ and $\boldsymbol{x}$ denote the denoised image and ground truth, respectively. The neural net is trained to remove the AWGN from its noisy input image. We also constrains the Lipschitz constant of* R_σ *to be smaller than* 1 *by using the spectral normalization technique in [68]. This provides a necessary condition for the satisfaction of Assumption 2.*

Proposition 8. *Let f be a convex function, then we have that*

$$f \text{ is Lipschitz continuous with constant } L > 0 \quad \Leftrightarrow \quad \|\boldsymbol{g}(\boldsymbol{x})\| \leq L, \quad \boldsymbol{g}(\boldsymbol{x}) \in \partial f(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{R}^n.$$

Proof. First assume that $\|\boldsymbol{g}(\boldsymbol{x})\| \leq L$ for all subgradients. Then, from the definition of subgradient

$$\begin{cases} f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \boldsymbol{g}(\boldsymbol{y})^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}) \\ f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})^\mathsf{T}(\boldsymbol{y} - \boldsymbol{x}) \end{cases} \quad \Leftrightarrow \quad \boldsymbol{g}(\boldsymbol{y})^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}) \leq f(\boldsymbol{x}) - f(\boldsymbol{y}) \leq \boldsymbol{g}(\boldsymbol{x})^\mathsf{T}(\boldsymbol{x} - \boldsymbol{y}).$$

Then, from Cauchy-Schwarz inequality, we obtain

$$-L\|\boldsymbol{x} - \boldsymbol{y}\| \leq -\|\boldsymbol{g}(\boldsymbol{y})\|\|\boldsymbol{x} - \boldsymbol{y}\| \leq f(\boldsymbol{x}) - f(\boldsymbol{y}) \leq \|\boldsymbol{g}(\boldsymbol{x})\|\|\boldsymbol{x} - \boldsymbol{y}\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|.$$

Now assume that g is L-Lipschitz continuous. Then, we have for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$

$$\boldsymbol{g}(\boldsymbol{x})^\mathsf{T}(\boldsymbol{y} - \boldsymbol{x}) \leq f(\boldsymbol{y}) - f(\boldsymbol{x}) \leq L\|\boldsymbol{y} - \boldsymbol{x}\|.$$

Consider $\boldsymbol{v} = \boldsymbol{y} - \boldsymbol{x} \neq \boldsymbol{0}$, then we have that

$$\boldsymbol{g}(\boldsymbol{x})^\mathsf{T}\left(\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\right) \leq L.$$

Since, this must be true for any $\boldsymbol{v} \neq \boldsymbol{0}$, we directly obtain $\|\boldsymbol{g}(\boldsymbol{x})\| \leq L$.

□

G Additional Technical Details

In this section, we present several technical details that were omitted from the main paper for space. Section G.1 discusses the architecture and training of the DnCNN prior. Section G.2 presents extra details and validations that compliment the experiments in Section 5 of the main paper with additional insights for IPA.

G.1 Architecture and Training of the DnCNN Prior

Fig. 3 visualizes 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, which indicates that intermediate images have the same spatial size as the input image. We generated 11101 training examples by adding AWGN to 400 images from the BSD400 dataset [69] and extracting patches of 128×128 pixels with stride 64. We trained DnCNN to optimize the *mean squared error* by using the Adam optimizer [77].

We use the spectral normalization technique in [68] 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 firmly non-expansiveness implies non-expansiveness, this provides a *necessary* condition for R_σ to satisfy Assumption 2.

Table 3: 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	$512\times512\times300$	1.17 GB	$512\times512\times600$	2.34 GB
	imaginary	$512 \times 512 \times 60$	0.23 GB	$512\times512\times300$	1.17 GB	$512\times512\times600$	2.34 GB
$\{\boldsymbol{y}_i\}$		$512 \times 512 \times 60$	0.47 GB	$512\times512\times300$	2.34 GB	$512\times512\times600$	4.69 GB
others combined		—	0.03 GB	—	0.03 GB	—	0.03 GB
Total			**0.97 GB**		**4.72 GB**		**9.41 GB**

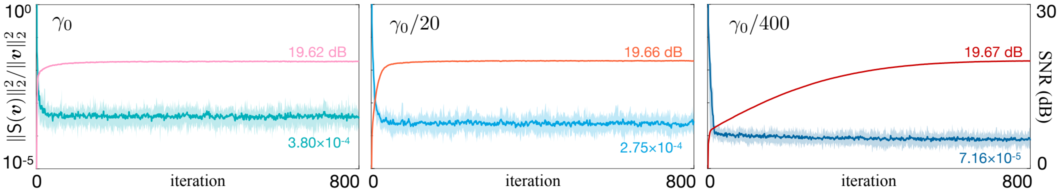

Figure 4: *Illustration of the convergence of IPA for a DnCNN prior under drastically changed γ values. The average normalized distance to* $\mathsf{zer}(\mathsf{S})$ *and SNR (dB) are plotted against the iteration number with the shaded areas representing the range of values attained over* 12 *test 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.*

G.2 Extra Details and Validations for Optical Tomography

All experiments are 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 $\hat{\boldsymbol{x}}$ represents the estimate and $\boldsymbol{x}$ denotes the ground truth.

In the intensity diffraction tomography, we implemented 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 3 shows extra examples of the memory usage specification for reconstructing 512×512 pixel permittivity images. These results follow the same trend observed in Table 2 of the main paper.

We also conduct some extra validations that provides 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 select 5 measurements in every iteration to impose the data-consistency. Fig. 4 compliments 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.

Prior work has discussed the influence of the denoising prior on the final result. Our last simulation compares the final reconstructed images of IPA by using TV, BM3D, and DnCNN. Since TV is

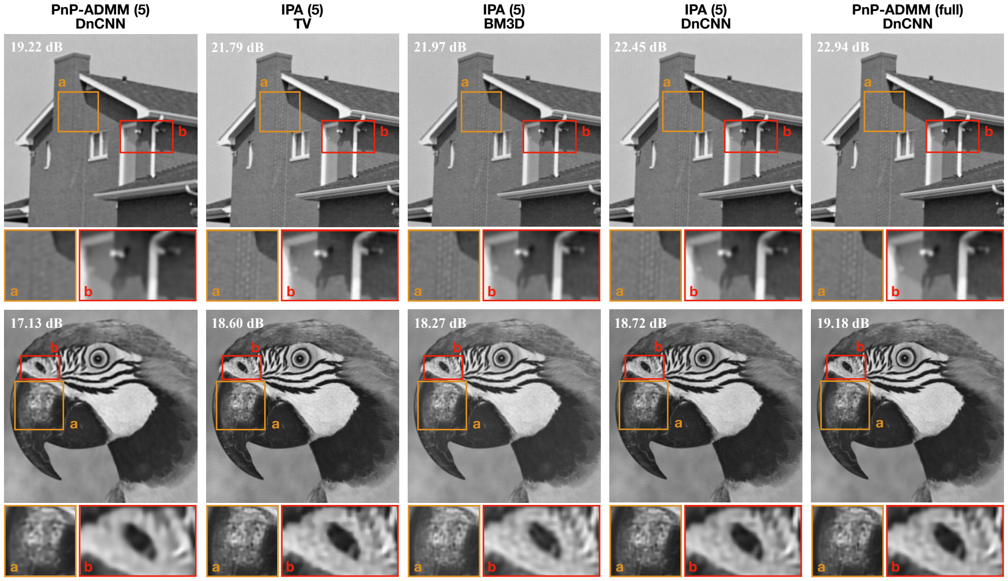

Figure 5: *Visual examples of the reconstructed House (upper) and Parrot (bottom) images by IPA and PnP-ADMM. The first and last columns correspond to PnP-ADMM under DnCNN with 5 fixed measurements and with the full 60 measurements, respectively. The second, third, and fourth column correspond to IPA 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.*

Table 4: Optimized SNR (dB) obtained by IPA under different priors for images from *Set 12*

Algorithms	PnP-ADMM (Fixed 5)	IPA (**Ours**) (Random 5 from full 60)			PnP-ADMM (Full 60)
Denoisers	DnCNN	TV	BM3D	DnCNN	DnCNN
Cameraman	15.95	17.45	17.38	18.16	18.34
House	19.22	21.79	21.97	22.45	22.94
Pepper	17.06	18.68	19.55	20.60	21.11
Starfish	18.20	19.29	20.29	21.64	22.22
Monarch	17.70	19.81	18.66	20.85	21.60
Aircraft	17.15	18.67	18.83	19.28	19.54
Parrot	17.13	18.60	18.27	18.72	19.18
Lenna	15.41	16.48	16.32	16.94	17.13
Barbara	13.63	16.00	17.53	16.58	16.85
Boat	17.98	19.35	20.21	20.95	21.34
Pirate	17.93	19.36	19.45	19.88	20.10
Couple	15.40	17.31	17.53	18.24	18.57
Average	**16.90**	**18.57**	**18.83**	**19.52**	**19.91**

a proximal operator, it serves as a baseline. Table 4 compares the average SNR values obtained by different image priors. We include the results of PnP-ADMM using 5 fixed measurements and the full batch as reference. Visual examples of *House* and *Parrot* are shown in Fig. 5. First, the table numerically illustrates significant improvement of IPA over PnP-ADMM under the same computational budget. Second, leveraging learned priors in IPA leads to the better reconstruction than other priors. For instance, DnCNN outperforms TV and BM3D by 0.7 dB in SNR. Last, the agreement between IPA and the full batch PnP-ADMM highlights the nearly optimal performance of our algorithm at a significantly lower computational cost and memory usage.