Practical Operator Sketching Framework for Accelerating Iterative Data-Driven Solutions in Inverse Problems

Junqi Tang, Guixian Xu, Subhadip Mukherjee, Carola-Bibiane Schönlieb

Practical Operator Sketching Framework for Accelerating Iterative Data-Driven Solutions in Inverse Problems

Junqi Tang[1*], Guixian Xu[1], Subhadip Mukherjee[2,3], Carola-Bibiane Schönlieb[3]

[1*]School of Mathematics, University of Birmingham, UK.
[2]Department of Electronics and Electrical Communication Engineering, IIT Kharagpur, India.
[3]Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK.

*Corresponding author(s). E-mail(s): j.tang.2@bham.ac.uk;
Contributing authors: gxx422@student.bham.ac.uk;
smukherjee@ece.iitkgp.ac.in; cbs31@cam.ac.uk;

Abstract

We propose a new operator-sketching paradigm for designing efficient iterative data-driven reconstruction (IDR) schemes, such as plug-and-play algorithms and deep unrolling networks. These IDR schemes are the state-of-the-art solutions for imaging inverse problems. However, for high-dimensional imaging tasks, such as X-ray CT, PET and MRI imaging, these IDR schemes typically become inefficient both in terms of computation, due to the need to compute the high-dimensional forward and adjoint operators multiple times. In this work, we introduce a universal dimensionality reduction framework for accelerating IDR schemes in solving imaging inverse problems, based on leveraging the sketching techniques from stochastic optimization. Using this framework, we derive several accelerated IDR schemes, including the plug-and-play multi-stage sketched gradient (PnP-MS2G) and sketching-based primal-dual (LSPD and Sk-LSPD) deep unrolling networks. Meanwhile, to fully accelerate PnP schemes when the denoisers are computationally expensive, we further propose novel stochastic lazy denoising schemes (Lazy-PnP and Lazy-PnP-EQ), leveraging the ProxSkip scheme in optimization and equivariant image denoisers, to significantly enhance the practicality and efficiency of PnP algorithms. We provide theoretical analysis for recovery guarantees of instances of the proposed framework. Our numerical experiments on

natural image processing and tomographic image reconstruction demonstrate the remarkable effectiveness of our sketched IDR schemes.*

Keywords: Deep Unrolling, Plug-and-Play Priors, Image Reconstruction, Sketching, Stochastic Optimization, Lazy-PnP

1 Introduction

Randomized sketching and stochastic first-order optimization methods have become the de facto techniques in modern data science and machine learning with a wide range of applications [3–6], due to their remarkable scalability to the size of optimization problems. The underlying optimization tasks in many applications nowadays are large-scale and high-dimensional by nature, as a consequence of big data and overparameterized models (for example, deep neural networks).

Although well-designed optimization algorithms can enable efficient machine learning, one can, on the other hand, utilize machine learning to develop problem-adapted optimization algorithms using the so-called "learning-to-learn" philosophy [7, 8]. Traditionally, the optimization algorithms are designed in a hand-crafted manner, with human-designed choices of rules for computing gradient estimates, step sizes, etc., for some general class of problems. Noting that although the traditional field of optimization has already obtained lower-bound matching algorithms (aka "optimal") [9–11] for many important general classes of problems, for specific instances there could be still much room for improvement. For example, a classical way to solve inverse imaging problems would be to minimize the regularized least squares [12] with specific measurement operators, which is a very narrow subclass of the general class of smooth and convex programs for which these optimization algorithms are developed "optimal". To obtain optimal algorithms adapted for a specific instance of a class, the hand-crafted mathematical design could be totally inadequate, and very often we do not even have a tight lower bound of it.

One of the highly active areas in modern data science is computational imaging (which is also recognized as low-level computer vision), especially medical imaging that includes X-ray computed tomography (CT) [13], magnetic resonance imaging (MRI) [14] and positron emission tomography (PET) [15]. In such applications, clinics seek to infer images of the patient's inner body from the noisy measurements collected from the imaging devices. Traditionally, dimensionality reduction methods, such as stochastic approximation [16] and sketching schemes [17–19] have been widely applied in solving large-scale imaging problems due to their scalability[20–22]. Inspired by their successes, in our work, we focus on developing a framework for efficient sketched iterative data-driven algorithms tailored for solving imaging inverse problems. In our framework, we effectively deal with the computational redundancy that is prevalent

*This paper contains some contents from its short conference version [1], and some early results/contents from our unpublished technical report [2].

in all of current state-of-the-art iterative data-driven reconstruction (IDR) schemes including plug-and-play (PnP)/ regularization-by-denoising (RED) schemes and deep unrolling (DU) networks. For example, our framework can accelerate any plug-and-play algorithm by reducing the computational cost of forward/adjoint operators and denoisers.

1.1 Contributions of this work

In this work, we make four main contributions:

- **Operator sketching framework for accelerating iterative data-driven reconstruction schemes**

  We propose an operator sketching framework for developing computationally efficient iterative data-driven reconstruction (IDR) methods, ranging from plug-and-play algorithms to deep unrolling networks based on a sketching scheme which we have tailored for imaging inverse problems for massively improving efficiency. We first derive our operator sketching scheme and obtain a plug-and-play multi-stage sketched gradient (PnP-MS2G) algorithm. Compared to state-of-the-art approaches such as PnP-SGD [21], we can observe numerically significant acceleration. By applying again our sketching framework to deep unrolling networks, we develop learned Stochastic Primal-Dual (LSPD) network, and its accelerated variant Sketched LSPD (SkLSPD) which is further empowered with the sketching approximation of products [17, 22, 23]. Our proposed networks can be viewed as sketched extensions of the state-of-the-art unrolling network – Learned Primal-Dual (LPD) network introduced in [24]. Noting that the LPD is a generic framework that takes most of the existing unrolling schemes as special cases, our acceleration schemes can be extended and applied to many other deep unrolling networks such as the ISTA-Net [25], ADMM-Net [26] and FISTA-Net [27].
- **Stochastic lazy denoisers for PnP schemes**

  While utilizing operator sketching we mitigate the inefficiency due to high-dimensionality measurement operators, the computational costs of state-of-the-art denoising functions are also not negligible. In our work, we propose first the Lazy-PnP scheme where we further introduce stochastic skipping schemes for mitigating the computational cost of the denoiser, which can be jointly applied with our operator sketching schemes for the ultimate acceleration. Moreover, we leverage recently introduced equivariant PnP priors and propose Lazy-PnP-EQ for improved stability and performance especially when state-of-the-art deep denoising networks are used. By skipping the calls of the denoiser with high-probability, we can achieve order-of-magnitude acceleration for gradient-based PnP algorithms in scenarios where the computational cost of the denoisers are dominant, such as image superresolution.
- **Theoretical analysis of our framework**

  We provide a theoretical analysis of the basic instance of our framework in accelerating proximal gradient descent and plug-and-play algorithms, from the view-point of stochastic non-convex composite optimization. We provide upper and lower bounds of the estimation errors under standard assumptions,

suggesting that our proposed PnP-MS2G has the potential to achieve similar estimation accuracy as its full-batch counterpart.

- **Less is more – the numerical effectiveness of our new plug-and-play methods and deep unrolling methods in imaging inverse problems**

  We provide numerical studies on the performance of the proposed new plug-and-play schemes (PnP-MS2G / Lazy-PnP / Lazy-PnP-EQ), showing significantly improved numerical results compared to the standard plug-and-play schemes in image processing and reconstruction tasks. We also numerically evaluate the performance of our proposed networks on two typical tomographic medical imaging tasks – low-dose and sparse-view X-ray CT. We compare our LSPD and SkLSPD with the full batch LPD. We found that our networks achieve competitive image reconstruction accuracy with the LPD, while only requiring a fraction of the computation of it.

2 Background

2.1 Imaging Inverse Problems

In imaging, the measurement systems can be generally expressed as:

$$b = Ax^\dagger + w, \tag{1}$$

where $x^\dagger \in \mathbb{R}^d$ denotes the ground truth image (vectorized), and $A \in \mathbb{R}^{n\times d}$ denotes the forward measurement operator, $w \in \mathbb{R}^n$ the measurement noise, while $b \in \mathbb{R}^n$ denotes the measurement data. A classical way to obtain a reasonably good estimate of $x^\dagger$ is to solve a composite optimization problem:

$$x^\star \in \arg\min_{x\in\mathbb{R}^d} f_b(Ax) + r(x), \tag{2}$$

where data fidelity term $f_b(Ax) := f(b, Ax)$ is typically a convex function (one example would be the least-squares $\|b - Ax\|_2^2$), while $r(x)$ being a regularization term, for example the total-variation (TV) semi-norm, or a learned regularization [28, 29]. A classical way to solve the composite optimization problem (2) is via the proximal gradient methods [12], which are based on iterations of the gradient descent step in f, proximal step on r and momentum step using previous iterations for fast convergence [30, 31].

Since modern imaging problems are often on huge scales, deterministic methods can be very computationally costly, since they need to apply the full forward and adjoint operators in each iteration. For scalability, stochastic gradient methods [16] and ordered subset methods [20, 32] are widely applied in real-world iterative reconstruction. More recent advanced stochastic variance-reduced gradient methods [5, 33–36] have also been adopted in some suitable scenarios in imaging for better efficiency [37–39].

More recently, deep learning approaches have been adapted in inverse imaging problems, starting from the work of [40] on the FBP-ConvNet approach for

tomographic reconstruction and DnCNN [41] for image denoising. Remarkably, the learned primal-dual (LPD) network [24], which mimics the update rule of the primal-dual gradient method and utilizes the forward operator and its adjoint within a deep convolutional network, achieves state-of-the-art results and outperforms primal-only unrolling approaches. Despite excellent performance, the computation of the learned primal-dual method is significantly larger than direct approaches such as FBP-ConvNet.

2.2 Iterative Data-Driven Reconstruction

In this section, we introduce the notion of iterative data-driven reconstruction (IDR) algorithms, which we will explore in this work. The current dominant IDR schemes can be summarized in two categories: the plug-and-play (PnP) algorithms and deep unrolling networks.

2.2.1 Plug-and-Play algorithms

Iterative reconstruction algorithms have become ubiquitous for solving imaging inverse problems such as image deblurring/inpainting/superresolution and tomographic image reconstruction (for example X-ray CT, MRI and PET, etc.). Due to their strengths in providing robust and consistent data reconstruction, these iterative solvers, especially when combined with advanced image priors [41–43] in a "plug-and-play" (PnP) manner [44–47], can still thrive in the current era where deep neural networks [40] have been successfully adopted in all these problems.

Although these classical convex regularization approaches provide theoretically tractable solutions for inverse problems, they have been significantly outperformed by the PnP priors, constructed by advanced image denoizers or deep neural networks. The very first PnP algorithm (probably not very well known) is proposed in [44], which is a PnP stochastic approximation algorithm with BM3D as denoiser. The PnP-ADMM of [45] and the PnP-LBFGS of [48] extend the classical methods ADMM and L-BFGS, replacing the proximal operator with the denoiser and have been widely applied in solving inverse problems since then. Then a very similar approach named regularization-by-denoising (RED) was proposed by [46, 47], which explicitly constructs the regularization term using the denoiser and provides improved convenience in parameter tuning. Since a strong link between PnP and RED is established in [49] under the RED-PRO unifying framework, in this work we refer to plug-and-play and regularization-by-denoising as "PnP" for simplicity. Although we mainly focus on PnP schemes in this work, our framework is obviously also applicable to RED.

For large-scale problems, PnP algorithms may require long computational times to obtain a good estimate. The PnP-SGD methods [21] and the stochastic PnP-ADMM methods [50, 51] can provide significant acceleration compared to the deterministic PnP-ADMM/PnP-LBFGS methods. In this work, we propose a generic acceleration of PnP gradient methods using dimensionality reduction/sketching in the image space. Moreover, we propose two enhanced acceleration schemes that deal with computational complexity in the denoiser, leveraging stochastic skipping of proximal operators [52]

in optimization, and equivariant denoising schemes for PnP algorithms [53] for stable application of deep denoising networks in PnP.

2.2.2 Deep unrolling

Now we start by presenting the background of the deep-unrolling networks, starting from the primal-dual-gradient-based optimization algorithm. It is well-known that, if the loss function $f_b(\cdot)$ is convex and lower semi-continuous, we can reformulate the original objective function (2) to a saddle-point problem:

$$[x^\star, y^\star] = \min_x \max_y \{r(x) + \langle Ax, y\rangle - f_b^*(y)\}, \tag{3}$$

where $f_b^*(\cdot)$ is the Fenchel conjugate of $f_b(\cdot)$:

$$f_b^*(y) := \sup_h \{\langle h, y\rangle - f_b(h)\} \tag{4}$$

The saddle-point problem (3) can be efficiently solved by the primal-dual hybrid gradient (PDHG) method [54], which is also known as the Chambolle-Pock algorithm in the optimization literature. The PDHG method for solving the saddle-point problem obeys the following updating rule:

Primal-Dual Hybrid Gradient (PDHG)

$-$Initialize $x_0, \bar{x}_0 \in \mathbb{R}^d$ $y_0 \in \mathbb{R}^n$

For $k = 0, 1, 2, ..., K$

$$\begin{cases} y_{k+1} = \text{prox}_{\sigma f_b^*}(y_k + \sigma A\bar{x}_k); \\ x_{k+1} = \text{prox}_{\tau r}(x_k - \tau A^T y_{k+1}); \\ \bar{x}_{k+1} = x_{k+1} + \beta(x_{k+1} - x_k); \end{cases}$$

The PDHG algorithm alternatively takes the gradients with respect to the primal variable x and the dual variable y and performs updates. In practice, it is often more desirable to reformulate the primal problem (2) to the primal-dual form (3), especially when the loss function f is non-smooth (or when the Lipschitz constant of the gradient is large).

Currently, the most successful deep networks in imaging would be the unrolling schemes [55] inspired by gradient-based optimization algorithms that use the knowledge of physical models. The state-of-the-art unrolling scheme —- the learned primal-dual network of [24] is based on the unfolding of the iteration of PDHG by replacing the proximal operators $\text{prox}_{\sigma f^\star}(\cdot)$ and $\text{prox}_{\tau g}(\cdot)$ with multilayer convolutional neural networks $\mathcal{P}_{\theta_p}(\cdot)$ and $\mathcal{D}_{\theta_d}(\cdot)$, with sets of parameters θ_p and θ_d, applied to both primal and dual spaces. The step sizes at each step are also set to be trainable. The learned primal-dual with K iterations can be written as the following, where the learnable paramters are $\{\theta_p^k, \theta_d^k, \tau_k, \sigma_k\}_{k=0}^{K-1}$:

Learned Primal-Dual (LPD)

$$
\begin{aligned}
&-\text{Initialize } x_0 \in \mathbb{R}^d \; y_0 \in \mathbb{R}^n \\
&\text{For} \quad k = 0, 1, 2, ..., K-1 \\
&\left\lfloor \begin{aligned} y_{k+1} &= \mathcal{D}_{\theta_d^k}(y_k, \sigma_k, Ax_k, b); \\ x_{k+1} &= \mathcal{P}_{\theta_p^k}(x_k, \tau_k, A^T y_{k+1}); \end{aligned} \right.
\end{aligned}
$$

When the primal and the dual CNNs are kept fixed across the layers of LPD, it has the potential to learn both the data-fidelity and the regularizer (albeit one might need additional constraints on the CNNs to ensure that they are valid proximal operators). This makes the LPD parameterization more powerful than a learned proximal-gradient network (with only a primal CNN), which can only learn the regularization functional. The capability of learning the data-fidelity term can be particularly useful when the noise distribution is unknown and one does not have a clear analytical choice for the fidelity term.

We choose the LPD as our backbone for deep unrolling because it is a generic framework that covers most existing gradient-based unrolling schemes as special cases. For example, if we choose the dual subnets of LPD to be a simple subtraction $Ax_k - b$, we can recover unrolled proximal gradient descent.

3 Iterative Operator Sketching Framework

We propose an operator sketching framework based on reduction in dimensionality in both the data dimension n and the parameter dimension d, constructing a much smaller proxy operator to replace the role of the full operator in the iterative reconstruction.

Our framework performs sketching in both the image domain (of dimension d) and the data domain (of dimension n). For ease of illustration, we use the least-squares objective and linear forward operator here without loss of generality. For a given forward operator $A \in \mathbb{R}^{n \times d}$, we can often find a low-dimensional proxy $A_s \in \mathbb{R}^{n \times m_0}$ discretized on a reduced image dimension $m_0 < d$ such that $Ax \approx A_s \mathcal{S}(x)$, where $\mathcal{S}(\cdot) : \mathbb{R}^d \to \mathbb{R}^{m_0}$ $(m_0 < d)$ is a sketching/downsampling operator. Furthermore, we can also perform random sketching $M(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ $(m < n)$ in the measurement / data domain, which corresponds to the stochastic approximation [16]. One typical choice of this sketching operator M is the subsampling sketch —- a uniformly sampled minibatch of $I_{n \times n}$ [17], which is suitable for inverse problems. For the image domain sketching operator $\mathcal{S}$, we found that off-the-shelf down-sampling algorithms such as the bicubic interpolation suffice in our framework. Now we can summarize this double-sketching as follows:

$$
\begin{aligned}
\|b - Ax\|_2^2 &\approx \|b - A_s \mathcal{S}(x)\|_2^2 \\
&\propto \mathbf{E}_M \|Mb - MA_s \mathcal{S}(x)\|_2^2.
\end{aligned} \tag{5}
$$

Instead of using standalone data domain sketches [17], our double-sketching framework is more effective in terms of dimensionality reduction and can be applied to generically accelerate PnP methods and also deep unrolling networks. Using the sketched loss in

(5), we can have an approximate data fit that can be efficiently optimized by SGD [16] or its variance-reduced variants [4]. To recover the same reconstruction quality as the original program, we can adjust the image domain sketch size m_0 stagewise in a coarse-to-fine manner.

We now further introduce our framework first under the context of plug-and-play algorithms:

3.1 Doubly-Sketched PnP

In this section, we present our multistage sketched gradient framework PnP-MS2G. Sketching techniques have been widely applied in large-scale optimization, especially in least-squares problems [17, 22, 23, 56, 57]. However, we found that such data-domain sketching methods are not efficient in imaging inverse problems, since very often the forward operator is relatively sparse, and even the most efficient sparse Johnson-Lindenstrauss transform [58] cannot provide significant computational gain here since the sketched operator typically has similar sparsity as the full operator. If we use a subsampling sketch which is the only practical data-domain sketch, the performance is similar to or worse than SGD methods in imaging inverse problems.

Instead of using data-domain sketches, we propose to perform sketching in image-domain, which appears to be much more effective and can be applied to generically accelerate PnP proximal gradient methods.

3.1.1 Algorithmic Framework

Suppose the original objective reads:

$$x^\star \in \arg\min_{x\in\mathcal{M}} f(b, Ax), \tag{6}$$

where $\mathcal{M}$ can be some implicit non-convex constraint set constructed by the denoiser (we write this for the ease of presentation), then our sketched objective can be generally expressed as:

$$x^\star \in \arg\min_{x\in\mathcal{M}} f(b, A_s\mathcal{S}(x)), \tag{7}$$

where $\mathcal{S}(\cdot) : \mathbb{R}^d \to \mathbb{R}^m$ $(m < d)$ being the sketching/downsampling operator, while $A_s \in \mathbb{R}^{n\times m}$ is the forward operator discretized on the reduced image space. We found that such a scheme provides a remarkably efficient approximation of the solution. We present our PnP-MS2G framework in Algorithm 1, where we denote $\mathcal{D}$ as the denoiser, $\mathcal{S}$ as the sketching operator and $\mathcal{U}$ as the upsampling operator.

To explain the motivation and derivation of Algorithm 1, we start by illustrating here a concrete example where the data-fidelity is the least squares. Noting that the PnP proximal gradient descent iteration can be written as:

$$x_{k+1} = \mathcal{D}[x_k - \eta \cdot (A^TAx_k - A^Tb)], \tag{8}$$

where $\mathcal{D}(\cdot)$ denotes the denoiser, which can be a denoising algorithm such as NLM/BM3D/TNRD, or a classical proximal operator of some convex regularization

Algorithm 1 — Plug-and-Play with Multi-Stage Sketched Gradients (PnP-MS2G)

Initialization: $x_0 \in \mathbb{R}^d$, number of stages K, sketch-size $[m_1, ..., m_K]$, sketched forward operator $[A_{s_1}, ...A_{s_K}]$, sketching operators $[\mathcal{S}_1, ..., \mathcal{S}_K]$, up-sampling operators $[\mathcal{U}_1, ..., \mathcal{U}_K]$, number of inner-loops for each stage $[N_1, ..., N_K]$, step-size sequence $[\eta_1, ..., \eta_{\sum_{k=1}^{K} N_k}]$, $\alpha \in (0, 1]$, iteration counter $i = 0$

For $k = 1$ **to** K

  For $j = 1$ **to** N_k

    $i \leftarrow i + 1$

    Generate random subsampling mask M_i

    Compute the image-domain sketch: $v = \mathcal{S}_k(x_i)$

    Compute gradient estimate $G := \nabla_v f(M_i b, M_i A_{s_k} v)$

    $z_{i+1} = x_i - \eta_i \mathcal{U}_k G$,

    $x_{i+1} = (1 - \alpha) z_{i+1} + \alpha \mathcal{D}(z_{i+1})$,

  Endfor

Endfor

Output x_{i+1}

(such as TV-prox), or a pretrained denoising deep network such as (DnCNN). Then our sketched gradient can be written as:

$$x_{k+1} = \mathcal{D}[x_k - \eta \cdot \mathcal{U}(A_s^T A_s \mathcal{S}(x_k) - A_s^T b)], \tag{9}$$

where $\mathcal{U}(\cdot)$ denotes the upsamling operator. Numerically, we found that off-the-shelf up/down-sampling algorithms such as the bicubic interpolation suffice to provide us good estimates of the true gradients. Using this scheme, an efficient approximation of the true gradient can be obtained, since A_s only takes a fraction of the computation of A, and usually $\mathcal{U}$ and $\mathcal{S}$ can be computed very efficiently.

To further reduce the computational complexity, we can also utilize stochastic gradient estimate:

$$x_{k+1} = \mathcal{D}[x_k - \eta_k \cdot \mathcal{U}((M_k A_s)^T M_k A_s \mathcal{S}(x_k) - (M_k A_s)^T M_k b)] \tag{10}$$

where M_k is uniformly sampled minibatch of $I_{n \times n}$ here for computing the stochastic gradient. Here we use a vanilla minibatch stochastic gradient estimator. We can also choose here those advanced stochastic variance-reduced gradient estimators [4, 34, 36, 59] for potentially even faster convergence.

In Algorithm 1 we present our PnP-MS2G framework, where we typically start with an aggressive sketch $\{A_{s_1}, \mathcal{S}_1\}$ with $m_1 \ll d$ for very fast initial convergence, and then for later stages we switch to medium-size sketches $\{A_{s_k}, \mathcal{S}_k\}$ with $m_k < d$ which are increasingly more conservative, to reach a reconstruction accuracy similar to the unsketched counterpart.

We also wish to point out that in our sketching framework both the denoiser $\mathcal{D}$, the upsampling function $\mathcal{U}$ and the sketching function $\mathcal{S}$ can be parameterized as deep (convolutional) neural networks and trained recursively or end-to-end, resulting in a new efficient deep unrolling scheme [24].

In the multisketch framework presented here, we gradually increase the size of the sketch m through stages.

3.2 Stochastic Lazy Denoisers

In many scenarios in imaging applications, the computational costs of the proximal operators/PnP denoisers are also significant compared to the costs of evaluating the gradients of the data-fidelity. Recall that our scheme can be summarized in one line:

$$x_{k+1} = \mathcal{D}[x_k - \eta_k \cdot \mathcal{U}((M_k A_s)^T M_k A_s \mathcal{S}(x_k) - (M_k A_s)^T M_k b)]. \tag{11}$$

Here one can further improve the efficiency in the early iteration if we replace the full operator $\mathcal{D}$ with a sketched/down-scaled version $\mathcal{D}_s$, such that $\mathcal{D}(\cdot) = \mathcal{U} \circ \mathcal{D}_s \circ \mathcal{S}(\cdot)$, then we have:

$$x_{k+1} = \mathcal{U} \circ \mathcal{D}_s \circ \mathcal{S}[x_k - \eta_k \cdot \mathcal{U}((M_k A_s)^T M_k A_s \mathcal{S}(x_k) - (M_k A_s)^T M_k b)]. \tag{12}$$

With this sketching scheme, which we perform triple-dimensionality reduction (both on the forward operator and denoiser), we can accelerate PnP algorithms in reducing the complexity on denoiser but only in early iterations. In this subsection, we propose two much more powerful schemes for accelerating PnP's denoiser computation based on the finding that the denoiser computation can actually be skipped with high-probability in each iteration and hence further accelerates our sketched gradient schemes for PnP.

Lazy-PnP

The computational overhead of computing the denoiser can be effectively reduced by avoiding computing the denoiser at each of the iterations of PnP. We propose a Lazy-Denoiser framework along with sketching, inspired by the ProxSkip algorithm [52] used for convex optimization and federated learning [60]. We present our Lazy-PnP scheme in Alg. 2, which allows us to execute the denoiser in only a fraction of iterations while maintaining the same convergence rates and reconstruction accuracy in practice.

The Lazy-PnP scheme presented in Alg. 2 utilizes an auxiliary variable h throughout the iterations for stabilization. This scheme is a stochastic approach that calls the denoiser with a relatively small probability p (in our experiments we choose $p = 20\%$ and $p = 50\%$), with the denoising step written as

$$x_{i+1} = \mathcal{D}(z_{i+1} - \frac{\eta}{p} h_i),$$

otherwise the denoising step is skipped ($x_{i+1} = z_{i+1}$). If the denoiser step is executed in one iteration, then the auxiliary variable h_i is also updated

$$h_{i+1} = h_i + \frac{p}{\eta}(x_{i+1} - z_{i+1}).$$

Algorithm 2 —Lazy PnP

Initialization: $x_0 \in \mathbb{R}^d$, $h_0 \in \mathbb{R}^d$, probability $p \in (0, 1]$ for calling the denoiser at each iteration. For accelerating PnP-MS2G, one can replace its inner-loop with this algorithm.
For $i = 1$ **to** K
Generate random subsampling mask M_i
Compute gradient estimate $G := \nabla_v f(M_i b, M_i A v)$
Compute: $z_{i+1} = x_i - \eta(G - h_i)$,
With probability p execute: $x_{i+1} = \mathcal{D}(z_{i+1} - \frac{\eta}{p} h_i)$, otherwise $x_{i+1} = z_{i+1}$
Compute: $h_{i+1} = h_i + \frac{p}{\eta}(x_{i+1} - z_{i+1})$
Endfor
Output x_{i+1}

Algorithm 3 —Lazy PnP-EQ (with Equivariant Denoiser)

Initialization: $x_0 \in \mathbb{R}^d$, $h_0 \in \mathbb{R}^d$, probability $p \in (0, 1]$ for calling the denoiser at each iteration. For accelerating PnP-MS2G, one can replace its inner-loop with this algorithm.
For $i = 1$ **to** K
Generate random subsampling mask M_i and group action T_{g_i} where $g_i \sim \mathcal{G}$
Compute gradient estimate $G := \nabla_v f(M_i b, M_i A v)$
Compute: $z_{i+1} = x_i - \eta(G - h_i)$,
With probability p execute: $x_{i+1} = T_{g_i}^{-1} \mathcal{D}(T_{g_i}(z_{i+1} - \frac{\eta}{p} h_i))$, otherwise $x_{i+1} = z_{i+1}$
Compute: $h_{i+1} = h_i + \frac{p}{\eta}(x_{i+1} - z_{i+1})$
Endfor
Output x_{i+1}

While performing the gradient descent step, the auxiliary variable h is included

$$z_{i+1} = x_i - \eta(G - h_i).$$

Since the variable h_i keeps the average of implicit gradients of the denoiser, it can successfully compensate for the fact that for most of the iterations the denoiser is skipped, while keeping the algorithm numerically stable. Numerically, we observe that we can easily skip $50\% - 80\%$ of the denoising steps while maintaining the same convergence rates for gradient-based PnP algorithms.

Equivariant Lazy PnP

Recently, a simple way of increasing the performance and stability of PnP methods has been proposed, namely equivariant PnP [53]. Suppose we denote unitary matrix $\{T_g\}_{g \in \mathcal{G}}$ as transforms for some group $\mathcal{G}$, the equivariant denoiser can be expressed as:

$$\mathcal{D}_{\mathcal{G}}(x) = T_g^{-1} \mathcal{D}(T_g x), \quad g \sim \mathcal{G} \tag{13}$$

Typical choices of the transforms include rotations, translations, reflections, etc. This scheme was, in fact, first found numerically in the work of Zhang et al. [61]. In the work of Terris et al. [53], this approach is formally analyzed and studied, demonstrating remarkable performance in stabilizing iterations of PnP algorithms with performance gains. As this scheme is crucial for the application of deep denoiser in PnP schemes, we also leverage this in our Lazy-PnP framework, leading to a new algorithm as a side contribution.

In our Algorithm 2 and 3 we present Lazy-PnP schemes with stochastic gradients only because of ease of reading. Note that the same technique can be easily merged fully with the PnP-MS2G framework, we omit this to avoid redundant presentations.

3.3 Accelerating the Deep Unrolling Schemes via Operator Sketching

In this section, we present our two schemes for accelerating deep-unrolling networks.

3.3.1 One side sketching: Subset approximation

In our new approach, we propose to replace the forward and adjoint operators in the full-batch LPD network of [24], with only subsets of it. The proposed network can be seen as an unrolled version of stochastic PDHG [6] (but with ordered subsets and without variance reduction). We partition the forward and adjoint operators into subsets m, and also the corresponding measurement data. In each layer, we use only one of the subsets, in cycling order. Let $\mathbf{M} := [M_0, M_1, M_2, ..., M_{m-1}]$ be the set of subsampling operators, then the saddle-point problem (3) can be rewritten as:

$$[x^\star, y^\star] = \min_x \max_y \{r(x) + \sum_{i=0}^{m-1} \langle M_i A x, y_i \rangle - f^*_{b_i}(y_i)\}. \tag{14}$$

Utilizing this finite-sum structure, our learned stochastic primal-dual (LSPD) network can be described as[1]:

Learned Stochastic Primal-Dual (LSPD)

$-$Initialize $x_0 \in \mathbb{R}^d$ $y_0 \in \mathbb{R}^{n/m}$

For $k = 0, 1, 2, ..., K-1$

$i =$ mod (k, m);

(or pick i from $[0, m-1]$ uniformly at random)

$y_{k+1} = \mathcal{D}_{\theta_d^k}(y_k, \sigma_k, (M_i A)x_k, M_i b)$;

$x_{k+1} = \mathcal{P}_{\theta_p^k}(x_k, \tau_k, (M_i A)^T y_{k+1})$;

In the scenarios where the forward operator dominates the computation in the unrolling network, for the same number of layers, our LSPD network is approximately

[1] Alternatively, one may also consider an optional learned momentum acceleration by keeping the memory of the outputs of a number of previous layers: $x_{k+1} = \mathcal{P}_{\theta_p^k}(X_k, \tau_k, (M_i A)^T y_{k+1})$ where $X_k = [x_k, x_{k-1}, ..x_{k-M}]$, at the cost of additional computation and memory. For such case the input channel of the subnets would be $M + 1$.

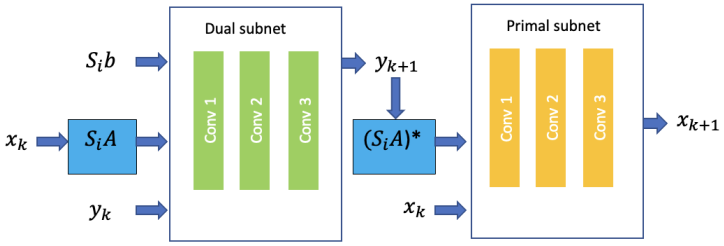

Fig. 1: One simple example of the practical choices for the building blocks of one layer of our LSPD network. Both dual and primal subnetworks are consist of 3 convolutional layers. The dual subnet has 3 input channels concatenating $[M_ib, M_iAx_k, y_k]$, while the primal subnet has 2 input channels for $[(M_iA)^Ty_{k+1}, x_k]$

m-time more efficient than the full-batch LPD network in terms of computational complexity. The LSPD we presented here describes a framework of deep learning-based methods depending on the parameterization of the primal and dual subnetworks and how they are trained. In practice, the LPD and LSPD networks usually achieve best performance when trained completely end-to-end. While being the most recommended in practice, when trained end-to-end, it is almost impossible to provide any non-trivial theoretical guarantees. An alternative approach is to restrict the subnets across layers to be the same and train the subnetwork to perform denoising [21, 50, 62, 63], artifact removal [64], or approximate projection to an image manifold [65], leading to a plug-and-play [45–47] type approach with theoretical convergence guarantees.

Note that our LSPD network covers the SGD-Net of [66] as a special case, by setting the dual network to be a simple subtraction and limiting the primal network to only have one input channel taking in the stochastic gradient descent step with a fixed primal scalar step size. We refer to this type of networks as the Learned SGD (LSGD) in this paper:

LSGD – Initialize $x_0 \in \mathbb{R}^d$ $y_0 \in \mathbb{R}^{n/m}$
For $k = 0, 1, 2, ..., K-1$
  Pick i from $[0, m-1]$ uniformly at random
  $y_{k+1} = M_iAx_k - M_ib$;
  $x_{k+1} = \mathcal{P}_{\theta_p^k}(x_k - \tau \cdot (M_iA)^Ty_{k+1})$;

which is a stochastic variant of ISTA-Net [25]. Stochastic unrolling can potentially give m fold reduction in the complexity per iteration of unrolling.

3.3.2 Double-sided Operator Sketching

Now we are ready to present our sketched LPD and sketched LSPD networks. Our main idea is to speedily approximate the products Ax_k, $A^T y_{k+1}$:

$$Ax_k \approx A_{s_k}\mathcal{S}_{\theta_s^k}(x_k), \quad A^T y_{k+1} \approx \mathcal{U}_{\theta_u^k}(A_{s_k}^T y_{k+1}) \tag{15}$$

where $\mathcal{S}_{\theta_s^k}(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d_{s_k}}$ $(d_{s_k} < d)$ being the sketching/downsampling operator which can be potentially trainable w.r.t parameters θ_s^k, while $A_{s_k} \in \mathbb{R}^{n\times d_{s_k}}$ is the sketched forward operator discretized on the reduced low-dimensional image space, and for the dual step we have $\mathcal{U}_{\theta_u^k} : \mathbb{R}^{d_{s_k}} \to \mathbb{R}^d$ the upsampling operator which can also be trained. In practice, we found that it actually suffices for us to just use the most simple off-the-shelf up/down-sampling operators in Pytorch, for example the bilinear interpolation, to deliver excellent performance for the sketched unrolling networks. Our Sketched LPD network is written as:

Sketched LPD – Initialize $x_0 \in \mathbb{R}^d$ $y_0 \in \mathbb{R}^n$
For $k = 0, 1, 2, ..., K-1$
$\quad y_{k+1} = \mathcal{D}_{\theta_d^k}(y_k, \sigma_k, A_{s_k}\mathcal{S}_{\theta_s^k}(x_k), b);$
$\quad x_{k+1} = \mathcal{P}_{\theta_p^k}(x_k, \tau_k, \mathcal{U}_{\theta_u^k}(A_{s_k}^T y_{k+1}));$

Again, we can use the same approximation for stochastic gradient steps:

$$\begin{aligned}(M_i A)x_k &\approx (M_i A_{s_k})\mathcal{S}_{\theta_s^k}(x_k),\\ (M_i A)^T y_{k+1} &\approx \mathcal{U}_{\theta_u^k}((M_i A_{s_k})^T y_{k+1}),\end{aligned} \tag{16}$$

and hence we can write our Sketched LSPD (SkLSPD) network as:

SkLSPD(Option1)
–Initialize $x_0 \in \mathbb{R}^d$ $y_0 \in \mathbb{R}^{n/m}$
For $k = 0, 1, 2, ..., K-1$
$\quad i = \mod(k, m);$
$\quad$ (or pick i from $[0, m-1]$ uniformly at random)
$\quad y_{k+1} = \mathcal{D}_{\theta_d^k}(y_k, \sigma_k, (M_i A_{s_k})\mathcal{S}_{\theta_s^k}(x_k), M_i b);$
$\quad x_{k+1} = \mathcal{P}_{\theta_p^k}(x_k, \tau_k, \mathcal{U}_{\theta_u^k}((M_i A_{s_k})^T y_{k+1}));$

or alternatively:

SkLSPD(Option2)
–Initialize $x_0 \in \mathbb{R}^d$ $y_0 \in \mathbb{R}^{n/m}$
For $k = 0, 1, 2, ..., K-1$

$$
\begin{cases}
i = \quad \bmod (k, m); \\
\text{(or pick } i \text{ from } [0, m-1] \text{ uniformly at random)} \\
y_{k+1} = \mathcal{D}_{\theta_d^k}(y_k, \sigma_k, (M_i A_{s_k})\mathcal{S}_{\theta_s^k}(x_k), M_i b); \\
x_{k+1} = \mathcal{U}_{\theta_u^k}(\mathcal{P}_{\theta_p^k}(\mathcal{S}_{\theta_s^k}(x_k), \tau_k, (M_i A_{s_k})^T y_{k+1}));
\end{cases}
$$

3.3.3 Remark regarding varying "coarse-to-fine" sketch size for SkLPD and SkLSPD

Numerically we suggest that we should use more aggressive sketch at the beginning for efficiency, while conservative sketch or non-sketch at latter iterations for accuracy. One plausible choice we found numerically pretty successful is: for the last few unrolling layers of SkLPD and SkLSPD, we switch to usual LPD/LSPD (say if the number of unrolling layers is 20, we can choose the last four unrolling layers to be unsketched, that is, $A_{s_k} = A$ for $k > K_{\text{switch}}$), and we found numerically that the reconstruction accuracy is best preserved if we implement this scheme.

3.3.4 Remark regarding the Option 2 for further improving efficiency

The second option of our SkLPD and SkLSPD further accelerates the computation compared to Option 1, making the primal subnet take the low-dimensional images and gradients as input and then upscale. Noting that the usual choice for the up and down sampler would simply be an off-the-shelf interpolation algorithm such as bilinear or bicubic interpolation which can be very efficiently computed, in practice we found the optional 2 often more favorable computationally if we use the coarse-to-fine sketch size. Numerically, we found that SkLPD and SkLSPD with option 2 and coarse-to-fine sketch size can be both trained faster and more efficient in testing due to the further reduction in the computation of the primal-subnet, without loss on reconstruction accuracy compared to option 1.

4 Theoretical Analysis

In this section, we provide a theoretical recovery analysis of our operator sketching framework presented in the previous section. From this motivational analysis, our aim is to demonstrate the reconstruction guarantee of PnP-MS2G and compare it with the recovery guarantee of PnP-PGD/PnP-SGD derived under the same setting.

4.1 General Assumptions

We list here the assumptions we make in our analysis of our generic sketching framework:

$$
x_{k+1} = \mathcal{P}_\theta[x_k - \eta_k \cdot \mathcal{U}((M_k A_s)^T M_k A_s \mathcal{S}(x_k) - (M_k A_s)^T M_k b)] \tag{17}
$$

A. 1. ***(Approximate projection)*** *We assume that the denoiser is a ε-approximate projection towards a manifold $\mathcal{M}$:*

$$\mathcal{P}_\theta(x) = e(x) + P_{\mathcal{M}}(x), \tag{18}$$

where:

$$P_{\mathcal{M}}(x) := \arg\min_{z\in\mathcal{M}} \|x - z\|_2^2, \tag{19}$$

and,

$$\|e(x)\|_2 \leq \varepsilon_0, \quad \forall x \in \mathbb{R}^d. \tag{20}$$

Here we model the denoiser to be a ε-projection towards a manifold. Note that in practice the image manifold $\mathcal{M}$ typically forms a non-convex subset. We also make the conditions on the image manifold as follows:

A. 2. ***(Interpolation)*** *We assume the ground-truth image $x^\dagger \in \mathcal{M}$, where $\mathcal{M}$ is a closed set.*

With this condition on the manifold, we further assume a restricted eigenvalue condition (restricted strong convexity) which is necessary for robust recovery [67–70]:

A. 3. ***(Restricted Eigenvalue Condition)*** *We define a descent cone $\mathcal{C}$ at point $x^\dagger$ as:*

$$\mathcal{C} := \left\{ v \in \mathbb{R}^d | \ v = a(x - x^\dagger), \forall a \geq 0, x \in \mathcal{M} \right\}, \tag{21}$$

and the restricted strong-convexity constant μ_c to be the largest positive constant satisfies the following:

$$\frac{1}{n}\|Av\|_2^2 \geq \mu_c \|v\|_2^2, \quad \forall v \in \mathcal{C}. \tag{22}$$

and the restricted smoothness constant L_c to be the smallest positive constant satisfies:

$$\frac{1}{q}\|M_i A v\|_2^2 \leq L_c \|v\|_2^2, \quad \forall v \in \mathcal{C}.\ \forall i \in [m] \tag{23}$$

The restricted eigenvalue condition is standard and crucial for robust estimation guarantee for linear inverse problems, i.e., for a linear inverse problem to be non-degenerate, this type of condition must hold [68, 70, 71]. For example, in a sparse recovery setting, when the measurement operator is a Gaussian map (compressed sensing measurements) and $x^\dagger$ is s-sparse, one can show that μ_c can be as large as $O(1 - \frac{s \log d}{n})$ [70]. In our setting, we would expect an even better μ_c, since the mainifold of certain classes of real-world images should have much smaller covering numbers compared to the sparse set.

4.2 Estimation error bounds of PnP-MS2G

With the assumptions presented in the previous subsection, here we provide the recovery guarantee of PnP-MS2G on linear inverse problem where we have $b = Ax^\dagger$. Denoting L_s to be the smallest constant satisfying:

$$\frac{1}{q}\|M_i A v\|_2^2 \leq L_s \|v\|_2^2, \quad \forall v \in \mathbb{R}^d, i \in [m], \tag{24}$$

we can have the following result:

Theorem 1. *(Upper bound) Assuming **A.1-3**, let $\eta = \frac{1}{qL_s}$ and $b = Ax^\dagger + w$, the output of k-th stage of PnP-MS2G has the following guarantee for the estimation of $x^\dagger$:*

$$\mathbb{E}\|x_{N_k} - x^\dagger\|_2 \leq \alpha^{N_k}\|x_{\text{init}} - x^\dagger\|_2 + \frac{1-\alpha^{N_k}}{1-\alpha}(\varepsilon + \delta), \tag{25}$$

where x_{init} denotes the initial point of k-th stage of PnP-MS2G, $\alpha = \kappa(1 - \frac{\mu_c}{L_s})$, $\kappa = 1$ if $\mathcal{M}$ is convex, $\kappa = 2$ if $\mathcal{M}$ is non-convex, $\varepsilon = \varepsilon_0 + \kappa\eta\varepsilon_1 + \kappa\eta\varepsilon_2$ and let ,

$$\begin{aligned} &\delta := 2\eta\mathbb{E} \sup_{v\in\mathcal{C}\cap\mathcal{B}^d, i\in[m]} v^T A^T M_i^T M_i w \\ &\|M_i A_{s_k}\|_2 \|M_i A_{s_k}\mathcal{D}(x_k) - M_i A x_k\|_2 \leq \varepsilon_1, \forall i, k \\ &\|\mathcal{U}(M_i A_{s_k})^T y_k - (M_i A)^T y_k\|_2 \leq \varepsilon_2, \forall i, k \end{aligned} \tag{26}$$

When the restricted eigenvalue μ_c is large enough such that $\alpha < 1$, the PnP-MS2G has a linear convergence in the estimation error, up to $\frac{\varepsilon}{1-\alpha}$ only depending on the accuracy of the denoiser approximation in terms of projection. For many inverse problems, for example CT/PET tomographic imaging, we have $L_s \approx L_f$ where L_f being the largest eigenvalue of $\frac{1}{n}A^T A$, and in these tasks the same convergence rate as in Theorem 1 applies for both sketched algorithms and the full-batch counterpart. This suggests the tremendous potential for computational savings of PnP-MS2G over classical methods.

From the above bound, we can observe that the reconstruction accuracy of a certain stage of PnP-MS2G depends on ε_1 and ε_2, which are directly dependent on the sketch size in the image domain. If we eventually reduce the sketch size in the image domain, then $\varepsilon_1, \varepsilon_2 \to 0$ and we can have optimal estimation accuracy. Hence, this bound demonstrates that our multistage strategy is necessary.

On the other hand, using a similar technique, we can provide a complementing lower bound for the estimation error of PnP-MS2G:

Theorem 2. *(Lower bound.) Under the same conditions of Theorem 1, if we further assume the constraint set $\mathcal{M}$ is convex, for any $\gamma > 0$, $\exists R(\gamma)$, if $\|x_{\text{init}} - x^\dagger\|_2 \leq R(\gamma)$, the estimation error of the output of the k-th stage PnP-MS2G satisfies the lower bound:*

$$\mathbb{E}\|x_{N_k} - x^\dagger\|_2 \geq (1-\gamma)^{N_k}(1 - \frac{L_c}{L_s})^{N_k}\|x_{\text{init}} - x^\dagger\|_2 - \frac{L_s}{L_c}\varepsilon_\star \tag{27}$$

where $\varepsilon_\star = \varepsilon_0 + \kappa\eta\varepsilon_1 + \kappa\eta\varepsilon_2$.

Again, we present the proof of this result in the appendix.

5 Numerical Results

5.1 Numerical study for Sketched Plug-and-Play algorithms

We start by presenting the numerical results for applying our sketching framework on accelerating PnP algorithms. We start our illustration with sparse-view CT reconstruction tasks. Here we compare our PnP-MS2G with PnP-PGD and its stochastic variant PnP-SGD [21]. For our PnP-MS2G we perform a 4× downscale in the first

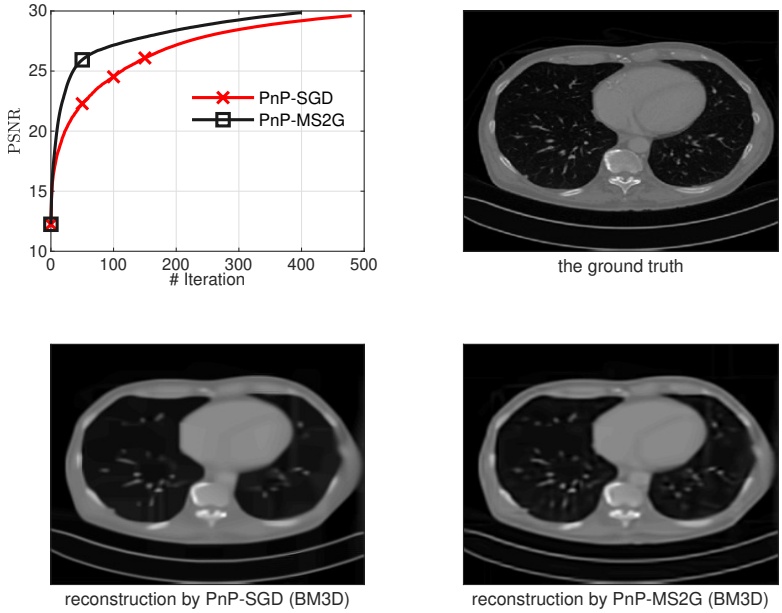

Fig. 2: Example of applying PnP-MS2G in X-ray CT reconstruction, comparing to the PnP stochastic gradient descent (PnP-SGD) proposed by Sun et al [21]. Note that each iteration of PnP-MS2G is more computationally efficient than PnP-SGD due to the dimensionality reduction by operator sketching. Surprisingly, even in terms of iteration-number the PnP-MS2G can provide better convergence rate comparing to PnP-SGD.

50 iterations, then a $2\times$ downscale afterwards, leading to significant improvement in computational efficiency. Here we choose the famous BM3D [42, 72] denoiser. In the numerical results we reported in the figures, we found that surprisingly even in terms of the iteration counts, our scheme can achieve an improvement on the convergence speed, let alone our PnP-MS2G is more efficient per iteration compared to PnP-PGD and PnP-SGD.

In Figures 3,4, and 6 we present numerical results for our Lazy PnP scheme on image superresolution and X-ray CT image reconstruction tasks. We first test our Lazy-PnP scheme with equivariant denoiser presented in Algorithm 3, compared to the standard equivariant PnP scheme [53] in image superresolution task. Here we seek to perform $4\times$ superresolution for low-resolution color images. The setting of this experiment is the same as the experiment in [53], with The interpolation kernel h for the task being the Guassian kernel of standard deviation 1:

$$b = (h * x)_{\downarrow 4} + \epsilon, \tag{28}$$

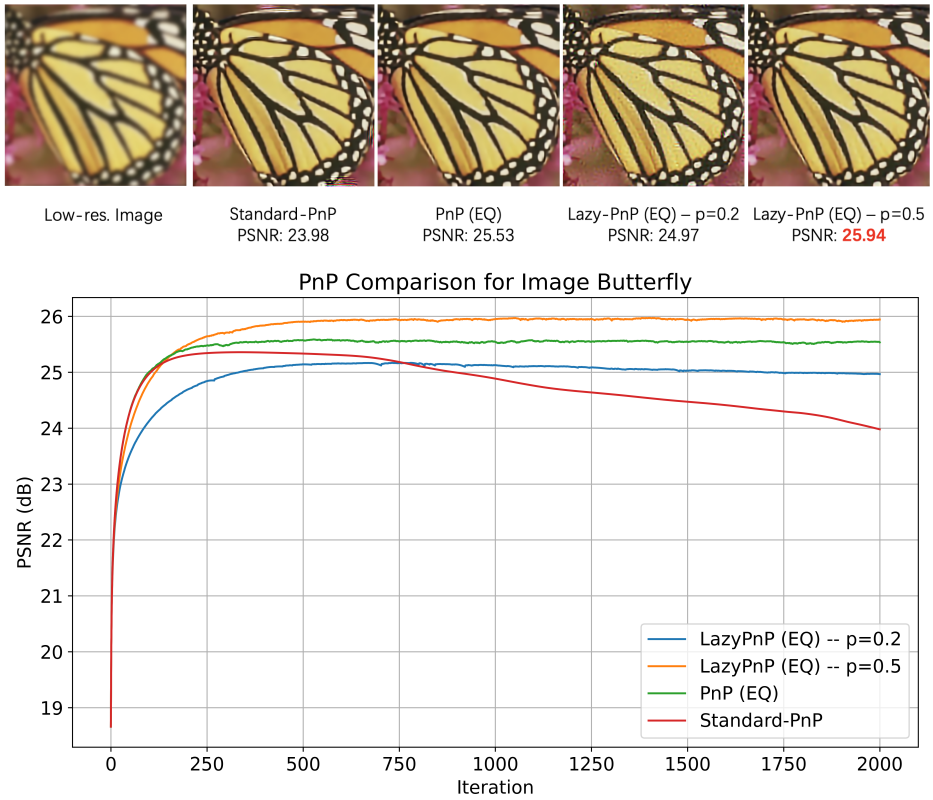

Fig. 3: **(Lazy-PnP)**Example of applying our Lazy-PnP-EQ for image superresolution ($4\times$). The denoiser network we choose here is DnCNN. Here we show the reconstructed image at 2000-th iteration.

where ϵ is a Gaussian noise with standard deviation 0.05. The denoiser we choose is the pre-trained DnCNN. The numerical results for superresolution are demonstrated in Figures 3 and 4, where we can observe that for this denoiser dominant case, our Lazy-PnP (EQ) with $p = 0.5$ consistently outperforms the equivariant PnP scheme, which means we save 50% of the calculation on the denoiser, while the standard PnP-PGD diverges.

In the X-ray CT experiment we presented in Figure 6, we implemented PnP-SGD with or without the Lazy-Denoiser scheme. We can observe that our Lazy-PnP has the same convergence rates compared to vanilla PnP-SGD, while it only requires the computation of the denoiser on 20% of the iteration.

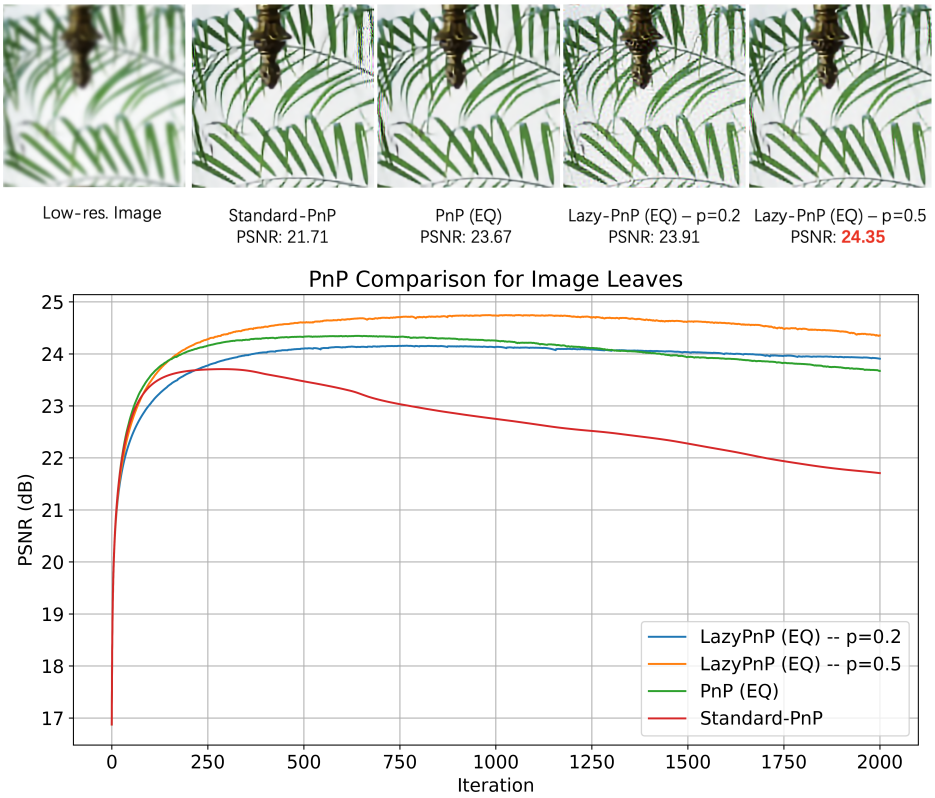

Fig. 4: **(Lazy-PnP)**Example of applying our Lazy-PnP-EQ for image superresolution ($4\times$). The denoiser network we choose here is DnCNN. Here we show the reconstructed image at 2000-th iteration.

5.2 Numerical experiments for sketched deep unrolling networks

The most basic training approach for LSPD/SkLSPD is end-to-end supervised training, where we consider fully paired training samples of measurement and "ground-truth" – which is typically obtained via a reconstruction from high-accuracy and abundant measurements. We take the initialization of LSPD/SkLSPD as a "filtered back-projection" $x^0 = A^\dagger b$. Let θ be the set of parameters $\theta := \{\theta_p^k, \theta_d^k, \tau_k, \sigma_k\}_{k=0}^{K-1}$, applying the LSPD/SkLSPD network on some measurement b can be written as $\mathcal{F}_\theta(b)$, the training objective can typically be written as:

$$\theta^\star \in \arg\min_\theta \sum_{i=1}^{N} \|x_i^\dagger - \mathcal{F}_\theta(b_i, x_i^0)\|_2^2, \tag{29}$$

where we denote by N the number of paired training examples.

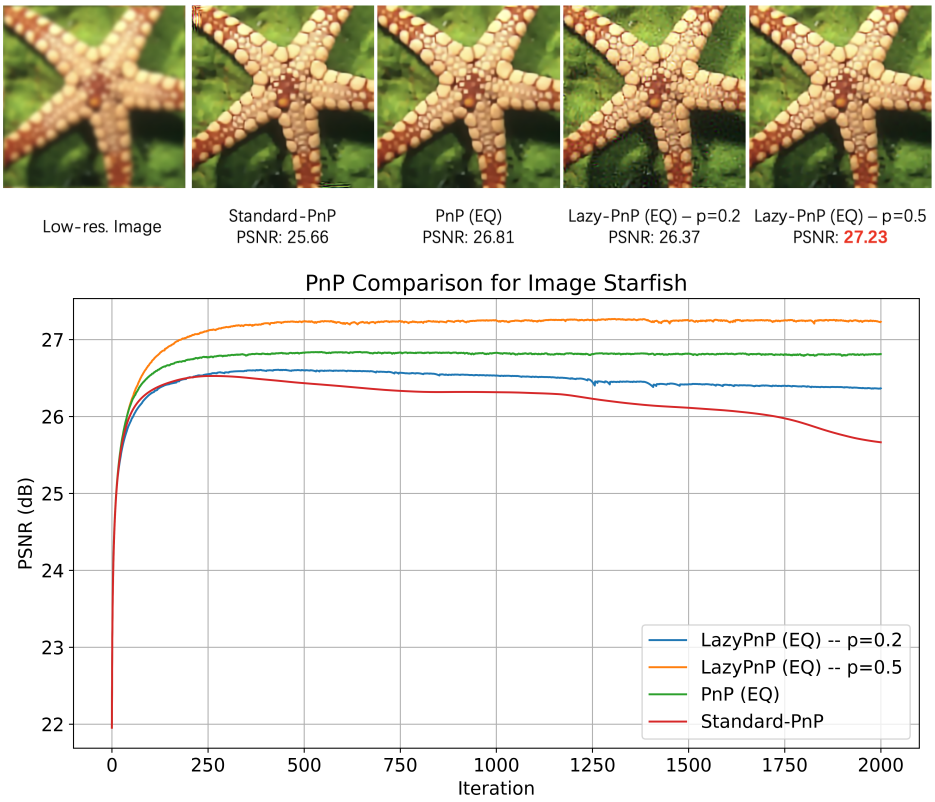

Fig. 5: **(Lazy-PnP)**Example of applying our Lazy-PnP-EQ scheme for image super-resolution (4×). The denoiser network we choose here is DnCNN. Here we show the reconstructed image at 2000-th iteration.

In this subsection, we present numerical results of our proposed networks for low-dose X-ray CT. In real world clinical practice, low-dose CT is widely used and highly recommended, since intense exposures to X-ray could significantly increase the risk of inducing cancers [73]. The low-dose CT takes a large number of low-energy X-ray views, leading to huge volumes of noisy measurements. This makes reconstruction schemes struggle to achieve efficient and accurate estimations. In our X-ray CT experiments, we use the standard Mayo-Clinic dataset [74] that contains 10 patients' 3D CT scans. We used 2111 slices (of 9 patients) of 2D images sized 512×512 for training and 118 slices of the remaining 1 patient for testing. We used the ODL toolbox [24] to simulate fan beam projection data with 800 equally spaced angles of view (each view includes 800 rays). The fan beam CT measurement is corrupted with Poisson noise: $b \sim \text{Poisson}(I_0 e^{-Ax^\dagger})$, where we make the low-dose choice of $I_0 = 7 \times 10^4$. We use the Beer-Lambert law to simulate the noisy projection data, and to linearize the measurements we consider the log data.

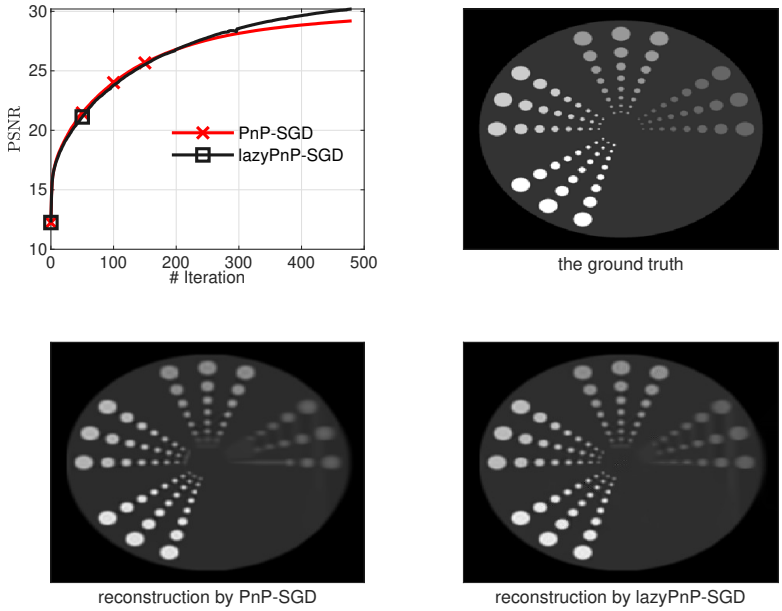

Fig. 6: **(Lazy-PnP)**Example of applying our Lazy-PnP with minibatch sketches in X-ray CT reconstruction. We choose $p = \frac{1}{5}$ for our Lazy-PnP-SGD, which means that the number of calls on the denoiser for our Lazy-PnP-SGD is only 20% of the standard PnP-SGD, while maintaining the same convergence rate.

In our LSPD and SkLSPD networks, we interleave-partition (according to the angles) the forward/adjoint operators and data into $m = 4$ subsets. Our networks has $K = 12$ layers[2] hence correspond to 3 data-passes, which means it takes only 3 calls in total on the forward and adjoint operators. We compare it with the learned primal-dual (LPD) which has $K = 12$ layers, corresponding to 12 calls on the forward and adjoint operator. We train all networks with 50 epochs of Adam optimizer [3] with batch size 1, supervised.

For our SkLSPD we choose Option 2 presented in Section III-B, with the coarse-to-fine sketch size. For all these networks, we choose the subnetworks $\mathcal{P}_{\theta_k}$ and $\mathcal{D}_{\theta_k}$ to have 3 convolutional layers (with a skip connection between the first channel of input and the output) and 32 channels, with kernel size 5. The starting point x_0 is set to be the standard filtered-backprojection for all the unrolling networks. We set all of them to have 12 algorithmic layers ($K = 12$). For the upsamplers/downsamplers in our Sketched LSPD, we simply choose the bilinear upsample and downsample functions in Pytorch. When called, the up-sampler increases the input image 4 times larger

[2]each layer of LSPD includes a primal and a dual subnetwork with 3 convolutional layers with kernel size 5×5 and 32 channels, same for LPD.

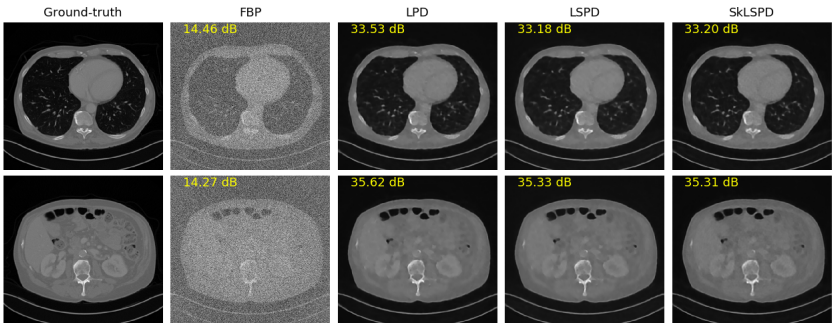

Fig. 7: Examples for Low-dose CT on the test set of Mayo dataset. We can observe that our LSPD networks achieve the same reconstruction performance as the full-batch LPD

Table 1: Low-dose CT testing results for LPD, LSPD and SkLSPD networks on Mayo dataset, with supervised training

Method	# calls on A and A^T	PSNR	SSIM	GPU time (s) on 1 pass of test set
FBP	-	14.3242	0.0663	
LPD *(12 layers)*	24	35.3177	0.9065	48.348
LSGD *(24 layers)*	12	31.5825	0.8528	33.089
LSPD *(12 layers)*	6	35.0577	0.9014	31.196
SkLSPD *(12 layers)*	4	34.9749	0.9028	23.996
SkLSPD *(12 layers, light weight on dual-step)*	4	34.6389	0.8939	19.843

(from 256×256 to 512×512), while the down-sampler makes the input image 4 times smaller (from 512×512 to 256×256). While the full forward operator A is defined on the grid of 512×512, the sketched operator A_s is defined on the grid of 256×256, therefore requiring only half of the computation in this setting. We use the coarse-to-fine strategy for SkLSPD, where we sketch the first 8 layers but left the last 4 layers unsketched. We also implement and test SkLSPD with a light-weight dual subnetwork (corresponding to a proximal operator of a weighted ℓ_2 loss with learnable weights; see SkLSPD-LW in Section IV).

In addition, we also implement the Learned SGD [66] in our setting which can be viewed as a simple special case of our LSPD network (see Section III-A) . Here for LSGD we choose the same parameterization of primal sub-networks as our LSPD (except for their original design, the LSGD sub-networks only take 1 input channel). To make a fair comparison, since LSGD do not have dual subnetworks, we allow the LSGD to have 24 layers, such that the total number of learnable parameters is similar to our LSPD.

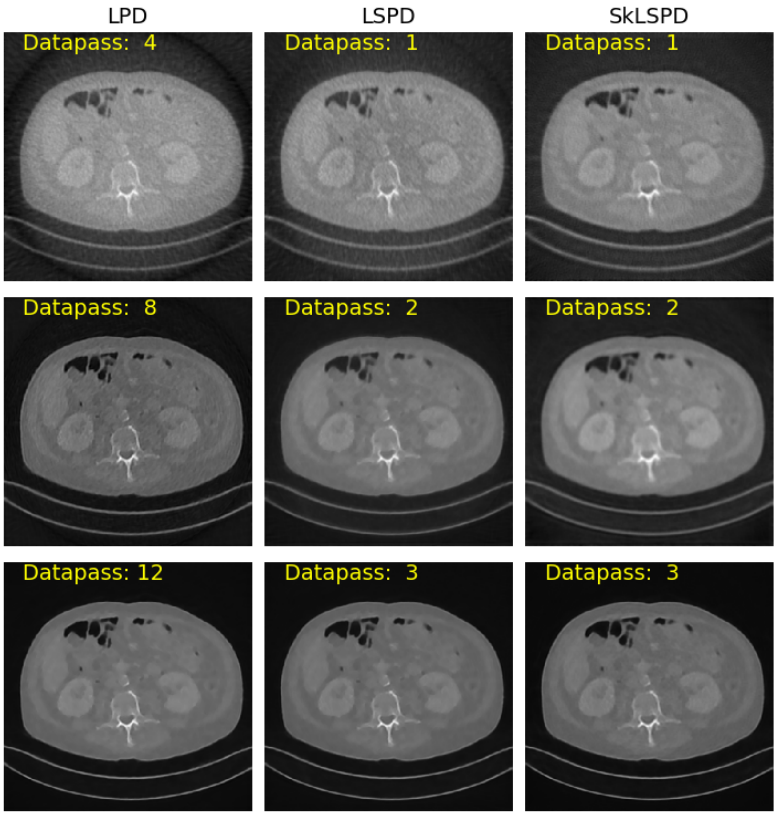

Fig. 8: Example for intermediate layer outputs for Low-dose CT on the test set of Mayo dataset. We can observe that LSPD/SkLSPD achieves competitive reconstruction quality with LPD across intermediate layers.

We present the performance of LPD, LSPD, and SkLSPD in the test set in Table 1, and some illustrative examples in Figure 2 for a visual comparison. We also present the results of the classical Filtered Backprojection (FBP) algorithm, which is widely used in clinical practice. We can observe from the FBP baseline that due to the challenging extreme low-dose setting, the FBP reconstruction fails completely. This can be partially addressed by U-Net postpocessing (FBPConvNet, [40]), whose parameter size is one order of magnitude larger than our unrolling networks. Next, we turn to the learned reconstruction results. From the numerical results, we found that our LSPD and SkLSPD networks both achieve almost the same reconstruction accuracy compared to the baseline of LPD in terms of PSNR (peak signal-to-noise ratio) and SSIM (structural similarity index, [75]) measures, with only requiring a fraction of the computation of forward and adjoint operators. In terms of run time on the GPU, our acceleration can introduce a reduction of around 40% to 60% compared to the full batch LPD.

Table 2: Sparse-View CT testing results for LPD and SkLSPD networks on Mayo dataset, with supervised training

Method	# calls on A and A^T	PSNR	SSIM	GPU time (s) on 1 pass of test set
FBP	-	22.0299	0.2713	
LPD *(12 layers)*	24	36.9198	0.9129	28.018
SkLSPD *(12 layers)*	4	36.6359	0.9178	17.340

In Table II, we present additional results on another widely applied modality in clinical practice, the sparse-view CT, where we take fewer measurements in normal doses. Here we again use the ODL toolbox to simulate fan-beam projection data with 200 equally spaced angles of views (each view includes 1200 rays). The fan beam CT measurement is corrupted with Poisson noise: $b \sim \text{Poisson}(I_0 e^{-Ax^\dagger})$, where we choose the normal dose of $I_0 = 7 \times 10^6$. Unlike low-dose CT, the main challenge of sparse-view CT is the ill-poseness of inverse problems, namely that the measurement operator is highly underdetermined with a nontrivial null space.

6 Conclusion

In this work, we proposed a new paradigm for accelerating iterative data-driven reconstruction (IDR) schemes such as plug-and-play methods and deep unrolling networks, and we performed recovery analysis for such frameworks for the first time and performed a thorough comparison to full-batch unrolling. Our generic framework is based on leveraging the spirit of sketching in stochastic optimization and dimensionality reduction into the design of IDR schemes for computational efficiency and memory efficiency in solving large-scale imaging inverse problems. Moreover, we propose auxiliary denoiser sketching schemes to mitigate the computational overhead of advanced denoisers for plug-and-play methods. We have provided a theoretical analysis of the proposed framework for the estimation guarantees from the viewpoint of stochastic optimization theory. Then we provide a numerical study of the proposed schemes in the context of X-ray CT image reconstruction, demonstrating the effectiveness of our acceleration framework for deep-unrolling networks. Although in this paper we mostly applied our sketching framework for accelerating PnP and deep unrolling, we need to emphasize here that this framework can be easily applied to accelerate newer algorithmic schemes such as deep restoration prior [76] and deep equilibrium models [77].

Declarations

6.1 Ethical approval

This declaration is not applicable.

6.2 Competing interests

There are no competing interests to declare.

6.3 Authors' contribution

JT, SM and CBS conceptualized the initial double-side sketching ideas of the manuscript, while GX and JT conceptualized the Lazy-PnP and Lazy-PnP-EQ schemes which are strong complements of the proposed framework. JT and GX worked on the implementations and numerical experiments. JT worked on the theoretical analysis and the writing of the manuscript. SM revised the writing of the manuscript in details. All authors reviewed the manuscript.

6.4 Funding

The work of GX is supported by a EPSRC international PhD scholarship. CBS acknowledges support from the Philip Leverhulme Prize, the Royal Society Wolfson Fellowship, the EPSRC advanced career fellowship EP/V029428/1, EPSRC Grants EP/S026045/1 and EP/T003553/1, EP/N014588/1, EP/T017961/1, the Wellcome Innovator Awards 215733/Z/19/Z and 221633/Z/20/Z, the European Union Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant agreement No. 777826 NoMADS, the Cantab Capital Institute for the Mathematics of Information and the Alan Turing Institute.

6.5 Availability of data and materials

The code of the algorithms and the image data used in the experiment will be made available on the website https://junqitang.com . For the phantom image example we use the one in the experimental section of [6], while the remaining CT images are all from the Mayo Clinic Dataset [74] which is publicly available. For the natural images and pre-trained DnCNN in the superresolution experiments one can obtain from the DeepInverse toolbox: https://deepinv.github.io/deepinv/.

A Proofs

A.1 Proof of Theorem 3.1

In this proof we utilize several projection identities from [70]. We list them here first for completeness. The first one would be the cone-projection identity:

$$\|P_{\mathcal{C}}(x)\|_2 = \sup_{v \in \mathcal{C} \cap \mathcal{B}^d} v^T x, \tag{30}$$

where $\mathcal{B}^d$ denotes the uni-ball in $\mathbb{R}^d$. The second one is the shift projection identity regarding that the Euclidean distance is preserved under translation:

$$P_{\mathcal{M}}(x+v) - x = P_{\mathcal{M}-x}(v). \tag{31}$$

Now if $0 \in \mathcal{M} - x$, we can have the third projection identity which is an important result from geometric functional analysis [70, Lemma 18]:

$$\|P_{\mathcal{D}}(x)\|_2 \leq \kappa_{\mathcal{D}} \|P_{\mathcal{C}}(x)\|_2, \tag{32}$$

where:

$$\kappa_{\mathcal{D}} = \begin{cases} 1 \text{ if } \mathcal{D} \text{ is convex} \\ 2 \text{ if } \mathcal{D} \text{ is non-convex} \end{cases} \tag{33}$$

where $\mathcal{D}$ is a closed set of potential non-convex included by the cone $\mathcal{C}$. On the other hand, utilizing a simplified result of [78] with partition minibatches, we have:

$$\begin{aligned} &\mathbb{E}_S(\|A^T M^T M A(x-z)\|_2^2) \\ &\leq 2L_s(\frac{q^2}{2n}\|Ax-b\|_2^2 - \frac{q^2}{2n}\|Az-b\|_2^2 - q^2\langle \nabla f(z), x-z\rangle). \end{aligned} \tag{34}$$

where $\nabla f(z) = \frac{1}{n}A^T(Az-b)$. Then for the case of noisy measurements $b = Ax^\dagger + w$, following similar procedure we can have:

$$\begin{aligned} &\mathbb{E}_S(\|A^T M^T M A(x-z)\|_2^2) \\ &\leq 2L_s(\frac{q^2}{2n}\|Ax-b\|_2^2 - \frac{q^2}{2n}\|Az-b\|_2^2 \\ &-q^2\langle \frac{1}{n}A^T(Az-b), x-z\rangle) \\ &\leq \frac{q^2 L_s}{n}(\frac{1}{2}\|A(x-x^\dagger)-w\|_2^2 - \|w\|_2^2 + \langle w, A(x-x^\dagger)\rangle) \\ &= \frac{q^2 L_s}{n}\|A(x-x^\dagger)\|_2^2 \end{aligned}$$

As shown in the theorem, we have assumed the approximation errors of the forward and adjoint operator are bounded:

$$\begin{aligned} &\|M_i A_{s_k}\|_2 \|M_i A_{s_k}\mathcal{D}(x_k) - M_i A x_k\|_2 \leq \varepsilon_1 \\ &\|\mathcal{U}(M_i A_{s_k})^T y_k - (M_i A)^T y_k\|_2 \leq \varepsilon_2, \forall i, k \end{aligned} \tag{35}$$

Denoting:

$$y_k := M_i A_{s_k}\mathcal{D}(x_k) - M_i b,$$

then for k-th iteration of PnP-MS2G we have the following:

$$\|x_{k+1} - x^\dagger\|_2$$

$$
\begin{aligned}
&= \|\mathcal{P}_{\theta_p}(x_k - \tau\mathcal{U}((M_iA_{s_k})^T y_k) - x^\dagger\|_2 \\
&\leq \|P_{\mathcal{M}}(x_k - \tau\mathcal{U}((M_iA_{s_k})^T y_k))) - x^\dagger\|_2 \\
&\quad + \|e(x_k - \tau\mathcal{U}((M_iA_{s_k})^T y_k)))\|_2 \\
&= \|P_{\mathcal{M}-x^\dagger}(x_k - x^\dagger - \tau\mathcal{U}((M_iA_{s_k})^T y_k)))\|_2 + \|e(\bar{x_k})\|_2,
\end{aligned}
$$

Then we can continue:

$$
\begin{aligned}
&\|x_{k+1} - x^\dagger\|_2 \\
&\leq \kappa\|P_{\mathcal{C}}(x_k - x^\dagger - \tau\mathcal{U}((M_iA_{s_k})^T y_{k+1}))\|_2 + \|e(\bar{x_k})\|_2 \\
&= \kappa \sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T(x_k - x^\dagger - \tau\mathcal{U}((M_iA_{s_k})^T y_{k+1}) + \|e(\bar{x_k})\|_2 \\
&\leq \kappa \sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T(x_k - x^\dagger - \tau A^T M_i^T(M_iAx_k - M_ib)) + \varepsilon \\
&= \kappa \sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T[x_k - x^\dagger \\
&\quad -\tau A^T M_i^T(M_iAx_k - M_i(Ax^\dagger + w))] + \varepsilon \\
&\leq \kappa \sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T[(I - \tau A^T {M_i}^T M_iA)(x_k - x^\dagger)] \\
&\quad +2\tau \sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T A^T M_i^T M_iw + \varepsilon \\
&\leq \kappa\|(I - \tau A^T {M_i}^T M_iA)(x_k - x^\dagger)\|_2 \\
&\quad +2\tau \sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T A^T M_i^T M_iw + \varepsilon.
\end{aligned}
$$

Denote $\bar{x_k} := x_k - \tau A^T {M_i}^T y_k$, and take expectation, then we have:

$$
\begin{aligned}
&\mathbb{E}(\|x_{k+1} - x^\dagger\|_2) \\
&\leq \kappa\mathbb{E}(\|(I - \tau A^T {M_i}^T M_iA)(x_k - x^\dagger)\|_2) \\
&\quad +\varepsilon \\
&\quad +2\tau\mathbb{E} \sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T A^T M_i^T M_iw \\
&\leq \kappa\sqrt{\mathbb{E}(\|(I - \tau A^T {M_i}^T M_iA)(x_k - x^\dagger)\|_2^2)} \\
&\quad +\varepsilon \\
&\quad +2\tau\mathbb{E} \sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T A^T M_i^T M_iw \\
&= \kappa\{\mathbb{E}(\|x_k - x^\dagger\|_2^2 - 2\tau\|M_iA(x_k - x^\dagger)\|_2^2 \\
&\quad +\tau^2\|A^T {M_i}^T M_iA(x_k - x^\dagger)\|_2^2)\}^{\frac{1}{2}} \\
&\quad +\varepsilon + 2\tau\mathbb{E} \sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T A^T M_i^T M_iw
\end{aligned}
$$

Now denoting:

$$\delta := 2\tau\mathbb{E} \sup_{v\in\mathcal{C}\cap\mathcal{B}^d, i\in[m]} v^T A^T M_i^T M_i w \tag{36}$$

and then:

$$\begin{aligned}
&\mathbb{E}(\|x_{k+1}-x^\dagger\|_2)\\
\leq\ &\kappa\{\|x_k-x^\dagger\|_2^2 - 2\frac{\tau q v_b}{n}\|A(x_k-x^\dagger)\|_2^2\\
&+\frac{\tau^2 q^2 L_s v_a}{n}\|A(x_k-x^\dagger)\|_2^2\}^{\frac{1}{2}} + \varepsilon + \delta\\
\leq\ &\kappa\{\|x_k-x^\dagger\|_2^2 - (2\tau q v_b - 2L_s\tau^2 q^2 v_a)\cdot\frac{1}{n}\|A(x_k-x^\dagger)\|_2^2\}^{\frac{1}{2}}\\
&+\varepsilon+\delta
\end{aligned}$$

and then due to Assumption A.3 the Restricted Eigenvalue Condition we have:

$$\begin{aligned}
&\mathbb{E}(\|x_{k+1}-x^\dagger\|_2)\\
\leq\ &\kappa\{\|x_k-x^\dagger\|_2^2 - (2\tau q\mu_c v_b - 2L_s\mu_c\tau^2 q^2 v_a)\|x_k-x^\dagger\|_2^2\}^{\frac{1}{2}}\\
&+\varepsilon+\delta\\
=\ &\kappa\{1-2\mu_c\tau q v_b + 2L_s\mu_c\tau^2 q^2 v_a\}^{\frac{1}{2}}\|x_k-x^\dagger\|_2\\
&+\varepsilon+\delta\\
\leq\ &\kappa(1-\frac{\mu_c v_b}{L_s v_a})\|x_k-x^\dagger\|_2+\varepsilon+\delta
\end{aligned}$$

Then let $\alpha = \kappa(1-\frac{\mu_c v_b}{L_s v_a})$, by the tower rule we get:

$$\mathbb{E}(\|x_k-x^\dagger\|_2) \leq \alpha^k\|x_0-x^\dagger\|_2 + \frac{(1-\alpha^k)}{1-\alpha}(\varepsilon+\delta).$$

A.2 Proof for Theorem 3.2

For proving the lower bound we will need to assume the constraint set $\mathcal{M}$ to be convex and apply a know result provide in [79, Lemma F.1], that for a closed convex set $\mathcal{D} := \mathcal{M} - x^\dagger$ containing the origin, given any $a, \gamma \in (0,1]$ there exist a positive constant C such that for any v satisfies $\|\mathcal{P}_\mathcal{C}(v)\|_2 \geq a\|v\|_2$ and $\|v\|_2 \leq c$, we can have:

$$\frac{\|P_\mathcal{D}(v)\|_2}{\|P_\mathcal{C}(v)\|_2} \geq 1-\gamma. \tag{37}$$

Since in A.2 we assume the ground truth $x^\dagger \in \mathcal{M}$ we know that $0 \in \mathcal{M} - x^\dagger$ hence the above claim is applicable. For k-th layer of simplified LSPD we have the following:

$$\begin{aligned}
&\|x_{k+1}-x^\dagger\|_2\\
=\ &\|\mathcal{P}_{\theta_p}(x_k - \tau\mathcal{U}(A_{s_k}^T {M_i}^T y_k)) - x^\dagger\|_2\\
\geq\ &\|P_\mathcal{M}(x_k - \tau\mathcal{U}(A_{s_k}^T {M_i}^T y_k)) - x^\dagger\|_2
\end{aligned}$$

$$
\begin{aligned}
&-\|e(x_k - \tau\mathcal{U}(A_{s_k}^T {M_i}^T y_k))\|_2 \\
=\ &\|P_{\mathcal{M}-x^\dagger}(x_k - x^\dagger - \tau\mathcal{U}(A_{s_k}^T {M_i}^T y_k))\|_2 \\
&-\|e(x_k - \tau\mathcal{U}(A_{s_k}^T {M_i}^T y_k))\|_2.
\end{aligned}
$$

Now due to (37) we can continue:

$$
\begin{aligned}
&\|x_{k+1} - x^\dagger\|_2 \\
\geq\ &(1-\gamma)\|P_{\mathcal{C}}(x_k - x^\dagger - \tau\mathcal{U}(A_{s_k}^T {M_i}^T y_k))\|_2 - \varepsilon_0 \\
=\ &(1-\gamma)\sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T(x_k - x^\dagger - \tau\mathcal{U}(A_{s_k}^T {M_i}^T y_k)) - \varepsilon_0 \\
=\ &(1-\gamma)\sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T(x_k - x^\dagger - \tau A^T {M_i}^T y_k) \\
&-\varepsilon_0 - \tau\varepsilon_2 \\
=\ &(1-\gamma)\sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T(x_k - x^\dagger \\
&-\tau A^T {M_i}^T(M_i A_{s_k}\mathcal{D}(x_k) - M_i b)) - \varepsilon_0 - \tau\varepsilon_2 \\
=\ &(1-\gamma)\sup_{v\in\mathcal{C}\cap\mathcal{B}^d} v^T(x_k - x^\dagger \\
&-\tau A^T {M_i}^T(M_i A x_k - M_i b)) - \varepsilon_0 - \tau\varepsilon_2 - \tau\varepsilon_1 \\
=\ &(1-\gamma)\|P_{\mathcal{C}}[(I - \tau A^T {M_i}^T M_i A)(x_k - x^\dagger)]\|_2 \\
&-\varepsilon \\
=\ &(1-\gamma)\sup_{v\in\mathcal{C}\cap\mathbb{S}^{d-1}} v^T(I - \tau A^T {M_i}^T M_i A)(x_k - x^\dagger) - \varepsilon \\
\geq\ &(1-\gamma)\frac{x_k - x^\dagger}{\|x_k - x^\dagger\|_2}(I - \tau A^T {M_i}^T M_i A)(x_k - x^\dagger) - \varepsilon \\
=\ &(1-\gamma)\|x_k - x^\dagger\|_2\left(1 - \tau\frac{\|M_i A(x_k - x^\dagger)\|_2}{\|x_k - x^\dagger\|_2}\right) - \varepsilon
\end{aligned}
$$

where we denote $\varepsilon = \varepsilon_0 + \tau\varepsilon_2 + \tau\varepsilon_1$. On the other hand since:

$$
\|(I - \tau A^T {M_i}^T M_i A)(x_k - x^\dagger)\|_2 \leq (1 + \tau q L_s)\|(x_k - x^\dagger)\|_2, \tag{38}
$$

and also note the second part of restricted eigenvalue condition we have:

$$
\|M_i A(x_k - x^\dagger)\|_2 \leq q L_c \|x_k - x^\dagger\|_2. \tag{39}
$$

Hence:

$$
\begin{aligned}
&\|P_{\mathcal{C}}[(I - \tau A^T {M_i}^T M_i A)(x_k - x^\dagger)]\|_2 \\
\geq\ &\|x_k - x^\dagger\|_2\left(1 - \tau\frac{\|M_i A(x_k - x^\dagger)\|_2}{\|x_k - x^\dagger\|_2}\right)
\end{aligned}
$$

$$\geq \left(\frac{1-q\tau L_c}{1+q\tau L_s}\right) \|(I-\tau A^T {M_i}^T M_i A)(x_k - x^\dagger)\|_2$$

Combining these three with $\tau = \frac{1}{qL_s}$, we find that (37) is satisfied for the choice $v = (I-\tau A^T {M_i}^T M_i A)(x_k - x^\dagger)$ and $a = \frac{L_s - L_c}{2L_s}$, we can write:

$$\|x_{k+1} - x^\dagger\|_2 \geq (1-\gamma)(1-\frac{L_c}{L_s})\|x_k - x^\dagger\|_2 - \varepsilon, \tag{40}$$

for all $\|x_k - x^\dagger\|_2 \leq \frac{\delta}{2}$ and by unfolding the iterations to x_0 we finish the proof.

An appendix contains supplementary information that is not an essential part of the text itself but which may be helpful in providing a more comprehensive understanding of the research problem or it is information that is too cumbersome to be included in the body of the paper.

References

[1] Tang, J., Mukherjee, S., Schönlieb, C.-B.: Iterative operator sketching framework for large-scale imaging inverse problems. hal-04701732 (2024)

[2] Tang, J., Mukherjee, S., Schönlieb, C.-B.: Accelerating deep unrolling networks via dimensionality reduction. arXiv preprint arXiv:2208.14784 (2022)

[3] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. Proceedings of 3rd International Conference on Learning Representations (2015)

[4] Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. In: Advances in Neural Information Processing Systems, pp. 315–323 (2013)

[5] Allen-Zhu, Z.: Katyusha: The first direct acceleration of stochastic gradient methods. The Journal of Machine Learning Research **18**(1), 8194–8244 (2017)

[6] Chambolle, A., Ehrhardt, M.J., Richtarik, P., Schonlieb, C.-B.: Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. SIAM Journal on Optimization **28**(4), 2783–2808 (2018)

[7] Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M.W., Pfau, D., Schaul, T., Shillingford, B., De Freitas, N.: Learning to learn by gradient descent by gradient descent. In: Advances in Neural Information Processing Systems, pp. 3981–3989 (2016)

[8] Li, Y., Zhang, K., Wang, J., Kumar, S.: Learning adaptive random features. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4229–4236 (2019)

[9] Woodworth, B.E., Srebro, N.: Tight complexity bounds for optimizing composite objectives. In: Advances in Neural Information Processing Systems, pp. 3639–3647 (2016)

[10] Lan, G.: An optimal method for stochastic composite optimization. Mathematical Programming **133**(1-2), 365–397 (2012)

[11] Lan, G., Zhou, Y.: An optimal randomized incremental gradient method. arXiv preprint arXiv:1507.02000 (2015)

[12] Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. Acta Numerica **25**, 161–319 (2016)

[13] Buzug, T.M.: Computed tomography. In: Springer Handbook of Medical Technology, pp. 311–342. Springer, ??? (2011)

[14] Vlaardingerbroek, M.T., Boer, J.A.: Magnetic Resonance Imaging: Theory and Practice. Springer, ??? (2013)

[15] Ollinger, J.M., Fessler, J.A.: Positron-emission tomography. Ieee signal processing magazine **14**(1), 43–55 (1997)

[16] Robbins, H., Monro, S.: A stochastic approximation method. The Annals of Mathematical Statistics **22**(3), 400–407 (1951)

[17] Pilanci, M., Wainwright, M.J.: Randomized sketches of convex programs with sharp guarantees. Information Theory, IEEE Transactions on **61**(9), 5096–5115 (2015)

[18] Drineas, P., Mahoney, M.W., Muthukrishnan, S., Sarlós, T.: Faster least squares approximation. Numerische Mathematik **117**(2), 219–249 (2011)

[19] Avron, H., Sindhwani, V., Woodruff, D.: Sketching structured matrices for faster nonlinear regression. In: Advances in Neural Information Processing Systems, pp. 2994–3002 (2013)

[20] Kim, D., Ramani, S., Fessler, J.A.: Combining ordered subsets and momentum for accelerated x-ray ct image reconstruction. IEEE transactions on medical imaging **34**(1), 167–178 (2015)

[21] Sun, Y., Wohlberg, B., Kamilov, U.S.: An online plug-and-play algorithm for regularized image reconstruction. IEEE Transactions on Computational Imaging (2019)

[22] Tang, J., Golbabaee, M., Davies, M.E.: Gradient projection iterative sketch for large-scale constrained least-squares. In: Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3377–3386. PMLR, ??? (2017)

[23] Pilanci, M., Wainwright, M.J.: Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares. Journal of Machine Learning Research **17**(53), 1–38 (2016)

[24] Adler, J., Öktem, O.: Learned primal-dual reconstruction. IEEE transactions on medical imaging **37**(6), 1322–1332 (2018)

[25] Zhang, J., Ghanem, B.: Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1828–1837 (2018)

[26] Sun, J., Li, H., Xu, Z., et al.: Deep admm-net for compressive sensing mri. Advances in neural information processing systems **29** (2016)

[27] Xiang, J., Dong, Y., Yang, Y.: Fista-net: Learning a fast iterative shrinkage thresholding network for inverse problems in imaging. IEEE Transactions on Medical Imaging **40**(5), 1329–1339 (2021)

[28] Mukherjee, S., Carioni, M., Öktem, O., Schönlieb, C.-B.: End-to-end reconstruction meets data-driven regularization for inverse problems. arXiv preprint arXiv:2106.03538 (2021)

[29] Mukherjee, S., Dittmer, S., Shumaylov, Z., Lunz, S., Öktem, O., Schönlieb, C.-B.: Learned convex regularizers for inverse problems. arXiv preprint arXiv:2008.02839 (2020)

[30] Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Transactions on Image Processing **18**(11), 2419–2434 (2009)

[31] Nesterov, Y.: Gradient methods for minimizing composite objective function. Technical report, UCL (2007)

[32] Erdogan, H., Fessler, J.A.: Ordered subsets algorithms for transmission tomography. Physics in Medicine & Biology **44**(11), 2835 (1999)

[33] Xiao, L., Zhang, T.: A proximal stochastic gradient method with progressive variance reduction. SIAM Journal on Optimization **24**(4), 2057–2075 (2014)

[34] Defazio, A., Bach, F., Lacoste-Julien, S.: Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In: Advances in Neural Information Processing Systems, pp. 1646–1654 (2014)

[35] Tang, J., Golbabaee, M., Bach, F., davies, M.E.: Rest-katyusha: Exploiting the solution's structure via scheduled restart schemes. In: Advances in Neural Information Processing Systems 31, pp. 427–438. Curran Associates, Inc., ??? (2018)

[36] Driggs, D., Tang, J., Liang, J., Davies, M., Schönlieb, C.-B.: A stochastic proximal alternating minimization for nonsmooth and nonconvex optimization. SIAM Journal on Imaging Sciences **14**(4), 1932–1970 (2021)

[37] Tang, J., Egiazarian, K., Davies, M.: The limitation and practical acceleration of stochastic gradient algorithms in inverse problems. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7680–7684 (2019). IEEE

[38] Tang, J., Egiazarian, K., Golbabaee, M., Davies, M.: The practicality of stochastic optimization in imaging inverse problems. IEEE Transactions on Computational Imaging **6**, 1471–1485 (2020)

[39] Karimi, D., Ward, R.K.: A hybrid stochastic-deterministic gradient descent algorithm for image reconstruction in cone-beam computed tomography. Biomedical Physics & Engineering Express **2**(1), 015008 (2016)

[40] Jin, K.H., McCann, M.T., Froustey, E., Unser, M.: Deep convolutional neural network for inverse problems in imaging. IEEE Transactions on Image Processing **26**(9), 4509–4522 (2017)

[41] Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing **26**(7), 3142–3155 (2017)

[42] Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE transactions on image processing: a publication of the IEEE Signal Processing Society **16**(8), 2080–2095 (2007)

[43] Tachella, J., Tang, J., Davies, M.: The neural tangent link between cnn denoisers and non-local filters. IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)

[44] Egiazarian, K., Foi, A., Katkovnik, V.: Compressed sensing image reconstruction via recursive spatially adaptive filtering. In: 2007 IEEE International Conference on Image Processing, vol. 1, p. 549 (2007). IEEE

[45] Venkatakrishnan, S.V., Bouman, C.A., Wohlberg, B.: Plug-and-play priors for model based reconstruction. In: 2013 IEEE Global Conference on Signal and Information Processing, pp. 945–948 (2013). IEEE

[46] Romano, Y., Elad, M., Milanfar, P.: The little engine that could: Regularization by denoising (red). SIAM Journal on Imaging Sciences **10**(4), 1804–1844 (2017)

[47] Reehorst, E.T., Schniter, P.: Regularization by denoising: Clarifications and new interpretations. IEEE Transactions on Computational Imaging **5**(1), 52–67 (2018)

[48] Tan, H.Y., Mukherjee, S., Tang, J., Schönlieb, C.-B.: Provably convergent plug-and-play quasi-newton methods. SIAM Journal on Imaging Sciences **17**(2), 785–819 (2024)

[49] Cohen, R., Elad, M., Milanfar, P.: Regularization by denoising via fixed-point projection (red-pro). SIAM Journal on Imaging Sciences **14**(3), 1374–1406 (2021)

[50] Tang, J., Davies, M.: A fast stochastic plug-and-play admm for imaging inverse problems. arXiv preprint arXiv:2006.11630 (2020)

[51] Sun, Y., Wu, Z., Wohlberg, B., Kamilov, U.S.: Scalable plug-and-play admm with convergence guarantees. arXiv preprint arXiv:2006.03224 (2020)

[52] Papoutsellis, E., Kereta, Z., Papafitsoros, K.: Why do we regularise in every iteration for imaging inverse problems? arXiv preprint arXiv:2411.00688 (2024)

[53] Terris, M., Moreau, T., Pustelnik, N., Tachella, J.: Equivariant plug-and-play image reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 25255–25264 (2024)

[54] Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of mathematical imaging and vision **40**(1), 120–145 (2011)

[55] Gregor, K., LeCun, Y.: Learning fast approximations of sparse coding. In: Proceedings of the 27th International Conference on International Conference on Machine Learning, pp. 399–406 (2010)

[56] Pilanci, M., Wainwright, M.J.: Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence. SIAM Journal on Optimization **27**(1), 205–245 (2017)

[57] Tang, J., Golbabaee, M., Davies, M.: Exploiting the structure via sketched gradient algorithms. In: 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 1305–1309 (2017). IEEE

[58] Woodruff, D.P., *et al.*: Sketching as a tool for numerical linear algebra. Foundations and Trends® in Theoretical Computer Science **10**(1–2), 1–157 (2014)

[59] Roux, N.L., Schmidt, M., Bach, F.R.: A stochastic gradient method with an exponential convergence _rate for finite training sets. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 2663–2671. Curran Associates, Inc., ??? (2012)

[60] Mishchenko, K., Malinovsky, G., Stich, S., Richtárik, P.: Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! In: International Conference on Machine Learning, pp. 15750–15769 (2022). PMLR

[61] Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., Timofte, R.: Plug-and-play image restoration with deep denoiser prior. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(10), 6360–6376 (2021)

[62] Kamilov, U.S., Mansour, H., Wohlberg, B.: A plug-and-play priors approach for solving nonlinear imaging inverse problems. IEEE Signal Processing Letters **24**(12), 1872–1876 (2017)

[63] Ono, S.: Primal-dual plug-and-play image restoration. IEEE Signal Processing Letters **24**(8), 1108–1112 (2017)

[64] Liu, J., Sun, Y., Eldeniz, C., Gan, W., An, H., Kamilov, U.S.: Rare: Image reconstruction using deep priors learned without groundtruth. IEEE Journal of Selected Topics in Signal Processing **14**(6), 1088–1099 (2020)

[65] Rick Chang, J., Li, C.-L., Poczos, B., Vijaya Kumar, B., Sankaranarayanan, A.C.: One network to solve them all–solving linear inverse problems using deep projection models. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5888–5897 (2017)

[66] Liu, J., Sun, Y., Gan, W., Xu, X., Wohlberg, B., Kamilov, U.S.: Sgd-net: Efficient model-based deep learning with theoretical guarantees. IEEE Transactions on Computational Imaging **7**, 598–610 (2021)

[67] Negahban, S.N., Ravikumar, P., Wainwright, M.J., Yu, B.: A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. Statistical Science, 538–557 (2012)

[68] Agarwal, A., Negahban, S., Wainwright, M.J.: Fast global convergence rates of gradient methods for high-dimensional statistical recovery. The Annals of Statistics **40**(5), 2452–2482 (2012)

[69] Wainwright, M.J.: Structured regularizers for high-dimensional problems: Statistical and computational issues. Annual Review of Statistics and Its Application **1**, 233–253 (2014)

[70] Oymak, S., Recht, B., Soltanolkotabi, M.: Sharp time–data tradeoffs for linear inverse problems. IEEE Transactions on Information Theory **64**(6), 4129–4158 (2017)

[71] Chandrasekaran, V., Recht, B., Parrilo, P.A., Willsky, A.S.: The convex geometry of linear inverse problems. Foundations of Computational mathematics **12**(6), 805–849 (2012)

[72] Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image restoration by sparse 3d transform-domain collaborative filtering. In: Image Processing: Algorithms and Systems VI, vol. 6812, p. 681207 (2008). International Society for Optics and

Photonics

[73] Mason, J.H.: Quantitative cone-beam computed tomography reconstruction for radiotherapy planning (2018)

[74] McCollough, C.: Tu-fg-207a-04: Overview of the low dose ct grand challenge. Medical physics **43**(6Part35), 3759–3760 (2016)

[75] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)

[76] Hu, Y., Delbracio, M., Milanfar, P., Kamilov, U.S.: A restoration network as an implicit prior. arXiv preprint arXiv:2310.01391 (2023)

[77] Gilton, D., Ongie, G., Willett, R.: Deep equilibrium architectures for inverse problems in imaging. IEEE Transactions on Computational Imaging **7**, 1123–1133 (2021)

[78] Gower, R.M., Loizou, N., Qian, X., Sailanbayev, A., Shulgin, E., Richtárik, P.: SGD: General analysis and improved rates. In: Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 5200–5209. PMLR, Long Beach, California, USA (2019)

[79] Oymak, S., Hassibi, B.: Sharp mse bounds for proximal denoising. Foundations of Computational Mathematics **16**(4), 965–1029 (2016)