Abstract

We propose a new operator-sketching paradigm for designing efficient iterative data-driven reconstruction (IDR) schemes, e.g., Plug-and-Play algorithms and deep unrolling networks. These IDR schemes are currently the state-of-the-art solutions for imaging inverse problems. However, for high-dimensional imaging tasks, especially X-ray CT and MRI imaging, these IDR schemes typically become inefficient both in terms of computation, due to the need of computing multiple times the high-dimensional forward and adjoint operators. In this work, we explore and propose 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 a number of accelerated IDR schemes, such as the plug-and-play multi-stage sketched gradient (PnP-MS2G) and sketching-based primal-dual (LSPD and Sk-LSPD) deep unrolling networks. Meanwhile, for fully accelerating PnP schemes when the denoisers are computationally expensive, we provide novel stochastic lazy denoising schemes (Lazy-PnP and Lazy-PnP-EQ), leveraging the ProxSkip scheme in optimization and equivariant image denoisers, which can massively accelerate the PnP algorithms with improved practicality. 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.

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 (Kingma and Ba, 2015; John-
son and Zhang, 2013; Allen-Zhu, 2017; Chambolle et al., 2018, 2024), due to their remarkable scalability to the size of the 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 the deep neural networks).

While well-designed optimization algorithms can enable efficient machine learning, one can, on the other hand, utilize machine learning to develop problem-adapted optimization algorithms via the so-called “learning-to-learn” philosophy (Andrychowicz et al., 2016; Li et al., 2019). 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 obtain lower-bound matching (aka, “optimal”) algorithms (Woodworth and Srebro, 2016; Lan, 2012; Lan and Zhou, 2015) for many important general classes of problems, for specific instances there could be still much room for improvement. For example, a classical way of solving imaging inverse problems would be via minimizing a total-variation regularized least-squares (Chambolle and Pock, 2016) with specific measurement operators – which is a very narrow sub-class of the general class of smooth and convex programs where these “optimal” optimization algorithms are developed for. To obtain optimal algorithm adapted for a specific instance of a class, the hand-crafted mathematical design could be totally inadequate, and very often we do not even know 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 including X-ray computed tomography (CT) (Buzug, 2011), magnetic resonance imaging (MRI) (Vlaardingerbroek and Boer, 2013), and positron emission tomography (PET) (Ollinger and Fessler, 1997). In such applications, the clinics seek to infer images of patients’ inner body from the noisy measurements collected from the imaging devices. Traditionally, dimensionality reduction methods such as stochastic approximation (Robbins and Monro, 1951) and sketching schemes (Pilanci and Wainwright, 2015; Drineas et al., 2011; Avron et al., 2013) have been widely applied in solving large-scale imaging problems due to their scalability (Kim et al., 2015; Sun et al., 2019; Tang et al., 2017b). 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 which are prevalent 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 via 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 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. Comparing to state-of-the-art approaches such as PnP-SGD (Sun et al., 2019), we can observe numerically significant acceleration. Via applying against our sketching framework to deep unrolling networks, we develop Learned Stochastic Primal-Dual (LSPD) network, and its accelerated variant Sketched LSPD (SkLSPD) which is fur-
ther empowered with the sketching approximation of products (Pilanci and Wainwright, 2015, 2016; Tang et al., 2017b). Our proposed networks can be viewed as a minibatch and sketched extension of the state-of-the-art unrolling network – Learned Primal-Dual (LPD) network of (Adler and Öktem, 2018). Noting that the LPD is a very generic framework which 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 (Zhang and Ghanem, 2018), ADMM-Net (Sun et al., 2016) and FISTA-Net (Xiang et al., 2021).

- Stochastic lazy denoisers for PnP schemes

While utilizing operator sketching we mitigate the inefficiency due to high-dimensionality measurement operators, the computational cost 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. Via 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 have 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 comparing 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 out that our networks achieve competitive image reconstruction accuracy with the LPD, while only requiring a fraction of the computation of it as a free-lunch, in both supervised and unsupervised training settings.

2. Background

2.1 Imaging Inverse Problems

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

\[ b = Ax^\dagger + w, \]  \hspace{1cm} (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:

\[
x^* \in \text{arg min}_{x \in \mathbb{R}^d} f_b(Ax) + r(x),
\]

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) seminorm, or a learned regularization (Mukherjee et al., 2021, 2020). A classical way of solving the composite optimization problem (2) is via the proximal gradient methods (Chambolle and Pock, 2016), which are based on iterations of gradient descent step on \( f \), proximal step on \( r \) and momentum step using previous iterates for fast convergence (Beck and Teboulle, 2009; Nesterov, 2007).

Since modern imaging problems are often in huge-scales, deterministic methods can be very computationally costly since they need to apply the full forward and adjoint operation in each iteration. For scalability, stochastic gradient methods (Robbins and Monro, 1951) and ordered-subset methods (Erdogan and Fessler, 1999; Kim et al., 2015) are widely applied in real world iterative reconstruction. More recent advanced stochastic variance-reduced gradient methods (Xiao and Zhang, 2014; Defazio et al., 2014; Allen-Zhu, 2017; Tang et al., 2018; Driggs et al., 2021) have also been adopted in some suitable scenarios in imaging for better efficiency (Tang et al., 2019, 2020; Karimi and Ward, 2016).

More recently, deep learning approaches have been adapted in imaging inverse problems, starting from the work of (Jin et al., 2017) on the FBP-ConvNet approach for tomographic reconstruction, and DnCNN (Zhang et al., 2017) for image denoising. Remarkably, the learned primal-dual (LPD) network (Adler and Öktem, 2018) which mimics the update rule of 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 the 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 be exploring in this work. The current dominating IDR schemes can be summarized under to 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 delivering data-consistent and robust reconstruction, these iterative solvers, especially when combined with advanced image priors (Dabov et al., 2007; Zhang et al., 2017; Tachella et al., 2021) in a “plug-and-play” (PnP) manner (Égiazarian et al., 2007; Venkatakrishnan et al., 2013; Romano et al., 2017; Reehorst and Schniter, 2018), can still thrive in current era where deep neural networks (Jin et al., 2017) have been successfully adopted in all these problems.

While 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 denoisers or deep neural networks. The very first PnP algorithm (probably not very well-known) is actually proposed by (Égiazarian et al., 2007), which is a PnP stochastic approximation algorithm with BM3D as the denoiser. The PnP-ADMM of (Venkatakrishnan et al., 2013) and PnP-LBFGS of (Tan et al., 2024) 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) has been proposed by (Romano et al., 2017; Reehorst and Schniter, 2018), which explicitly construct the regularization term using the denoiser and provide improved convenience in parameter tuning. Since a strong link between PnP and RED is established in (Cohen et al., 2021) under the RED-PRO unifying framework, in this work we refer both plug-and-play and regularization-by-denoising as “PnP” for simplicity. Although we mostly focusing on PnP schemes in this work, our framework is obviously also applicable to RED.

For large-scale problems, the PnP-ADMM and PnP-PGD may require long computational time to obtain a good estimate. The PnP-SGD (Sun et al., 2019) and stochastic PnP-ADMM methods (Tang and Davies, 2020; Sun et al., 2020) 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 which deal with the computational complexity on the denoiser, via leveraging stochastic skipping of proximal operators (Papoutsellis et al., 2024) in optimization, and equivariant denoising schemes for PnP algorithms (Terris et al., 2024) 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:

$$\begin{align*}
[x^*, y^*] &= \min_x \max_y \{r(x) + \langle Ax, y \rangle - f^*_b(y)\}, \\
\end{align*}$$

(3)

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

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

(4)

The saddle-point problem (3) can be efficiently solved by the primal-dual hybrid gradient (PDHG) method (Chambolle and Pock, 2011), 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 $k = 0, 1, 2, \ldots, K$

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

The PDHG algorithm takes alternatively the gradients regarding the primal variable $x$ and dual variable $y$ and performs the 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 most successful deep networks in imaging would be the unrolling schemes (Gregor and LeCun, 2010) inspired by the gradient-based optimization algorithms leveraging the knowledge of the physical
models. The state-of-the-art unrolling scheme – learned primal-dual network of (Adler and Öktem, 2018) is based on unfolding the iteration of PDHG by replacing the proximal operators \( \text{prox}_{\sigma_f}(\cdot) \) and \( \text{prox}_{\tau_g}(\cdot) \) with multilayer convolutional neural networks \( P_{\theta_p}(\cdot) \) and \( D_{\theta_d}(\cdot) \), with sets of parameters \( \theta_p \) and \( \theta_d \), applied on the both primal and dual spaces. The step sizes at each steps are also set to be trainable. The learned primal-dual with \( K \) iterations can be written as the following, where the learnable parameters are \( \{\theta^k_p, \theta^k_d, \tau^k, \sigma^k\}_{k=0}^{K-1} \).

\[
\begin{align*}
\text{Learned Primal-Dual (LPD)} \\
-\text{Initialize } x_0 \in \mathbb{R}^d, y_0 \in \mathbb{R}^n \\
\text{For } k = 0, 1, 2, ..., K-1 \\
| y_{k+1} &= D_{\theta_d}(y_k, \sigma_k, Ax_k, b) \\
| x_{k+1} &= P_{\theta_p}(x_k, \tau_k, A^T y_{k+1})
\end{align*}
\]

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 parametrization 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 because it is a very generic framework which covers all existing gradient-based unrolling schemes and plug-and-play algorithms as special cases. For instance, if we choose the dual subnetworks of LPD to be a simple subtraction \( Ax_k - b \), we can recover Learned ISTA/FISTA.

3. Iterative Operator Sketching Framework

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

Our framework performs sketching in both image domain (of dimension \( d \)) and 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 \( A_s \in \mathbb{R}^{n_0 \times m_0} \) discretized on a reduced image dimension \( m_0 < d \) such that \( Ax \approx A_s S(x) \), where \( S(\cdot) : \mathbb{R}^d \to \mathbb{R}^{m_0} \) is a sketching/downsampling operator. Furthermore, we can also perform random sketching \( M(\cdot) : \mathbb{R}^n \to \mathbb{R}^m \) on the measurement/data domain, which corresponds to stochastic approximation (Robbins and Monro, 1951). One typical choice of this sketching operator \( M \) is the subsampling sketch – uniformly sampled minibatch of \( I_{n \times n} \) (Pilanci and Wainwright, 2015), which is suitable for inverse problems. For the image domain sketching operator \( S \), we found that off-the-shelf downsampling algorithms such as the bi-cubic interpolation suffice in our framework. Now we can summarize this double-sketching as follows:

\[
\begin{align*}
\| b - Ax \|_2^2 &\approx \| b - A_s S(x) \|_2^2 \\
&\propto E_M \| Mb - MA_s S(x) \|_2^2.
\end{align*}
\]

Instead of using standalone data-domain sketches (Pilanci and Wainwright, 2015), 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. When using the sketched loss in (5), we can have an approximate data fit that can be efficiently optimized by SGD (Robbins and Monro, 1951) or its variance-reduced variants (Johnson and Zhang, 2013). To recover the same reconstruction quality as the original program, we can adjust the image-domain sketch size $m_0$ stage-wise 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 multi-stage sketched gradient framework PnP-MS2G. The sketching techniques have been widely applied in large-scale optimization especially the least-squares problems (Pilanci and Wainwright, 2017, 2016, 2015; Tang et al., 2017a,b). However, the author has 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 (Woodruff et al., 2014) cannot provide significant computational gain here since the sketched operator typically has similar sparsity as the full operator. If we use subsampling sketch which is the only practical data-domain sketch, the performance is similar 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:

$$x^* \in \arg \min_{x \in \mathcal{X}} f(b, Ax),$$

where $\mathcal{X}$ 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:

$$x^* \in \arg \min_{x \in \mathcal{X}} f(b, A_s S(x)),$$

where $S(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^m (m < d)$ 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 $D$ as the denoiser, $S$ as the sketching operator, and $U$ 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:

$$x_{k+1} = D[x_k - \eta \cdot (A^T A x_k - A^T b)],$$

where $D(\cdot)$ denotes the denoiser, which can be a denoising algorithm such as NLM/BM3D/TNRD, or a classical proximal operator of some convex regularization (such as TV-prox), or a pretrained denoising deep network such as (DnCNN). Then our sketched gradient can be written as:

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

where $U(\cdot)$ denotes the upsampling operator. Numerically we found that off-the-shelf up/down-sampling algorithms such as the bi-cubic 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 $U$ and $S$ can be very efficiently computed.
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 $[m_1, \ldots, m_K]$, sketched forward operator $[A_{s_1}, \ldots, A_{s_K}]$, sketching operators $[S_1, \ldots, S_K]$, up-sampling operators $[U_1, \ldots, U_K]$, number of inner-loops for each stage $[N_1, \ldots, N_K]$, step-size sequence $[\eta_1, \ldots, \eta_{\sum_{k=1}^{K} N_k}]$, $\alpha \in (0, 1]$, 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: $v = S_k(x_i)$
    Compute gradient estimate $G := \nabla_v f(M_i b, M_i A_{s_k} v)$
    $z_{i+1} = x_i - \eta_i U_k G$
    $x_{i+1} = (1 - \alpha)z_{i+1} + \alpha D(z_{i+1})$
  Endfor
Endfor

Output $x_{i+1}$

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

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

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 estimator (Roux et al., 2012; Johnson and Zhang, 2013; Defazio et al., 2014; Driggs et al., 2021) for potentially even faster convergence.

In Algorithm 1 we present our PnP-MS2G framework, where we typically start by an aggressive sketch $\{A_{s_1}, S_1\}$ with $m_1 \ll d$ for very fast initial convergence, and then for later stages we switch to medium size sketches $\{A_{s_k}, S_k\}$ with $m_k < d$ which are increasingly more conservative, to reach a similar reconstruction accuracy as the unsketched counterpart.

We also wish to point out that in our sketching framework both the denoiser $D$, the upsampling function $U$ and sketching function $S$ can be parameterized as deep (convolutional) neural-networks and trained either in a recursive or end-to-end manner, resulting in a new efficient deep-unrolling scheme (Adler and Öktem, 2018; Tang et al., 2021).

In the multi-sketch framework presented here we gradually increase the sketch size $m$ throughout stages.

3.2 Stochastic Lazy Denoisers

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

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

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

$$x_{k+1} = U \circ D_s \circ S[x_k - \eta_k \cdot U((M_k A_{s_k})^T M_k A_{s_k} S(x_k) - (M_k A_{s_k})^T M_k b)]$$

(12)
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 $G := \nabla_x f(M_i b, M_i A v)$
- Compute: $z_{i+1} = x_i - \eta(G - h_i)$
  - With probability $p$ execute: $x_{i+1} = D(z_{i+1} - \frac{\eta}{p} h_i)$, otherwise $x_{i+1} = z_{i+1}$
  - Compute: $h_{i+1} = h_i + \frac{p}{\eta} (x_i - z_{i+1})$

Endfor
Output $x_{i+1}$

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 via avoiding computing the denoiser at each of the iteration of PnP. We propose Lazy-Denoiser framework alongside with sketching, inspired by the ProxSkip algorithm (Papoutsellis et al., 2024) used for convex optimization and federated learning (Mishchenko et al., 2022). 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 utilize an auxiliary variable $h$ throughout the iterations for stabilization. This scheme is a stochastic approach which 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

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

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

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

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 the fact that for the 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 maintainly the same convergence rates for gradient-based PnP algorithms.
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: \( x_{i+1} = T_{g_i}^{-1} D(T_{g_i}(z_{i+1} - \frac{p}{\eta} h_i)) \), otherwise \( x_{i+1} = z_{i+1} \)
- Compute: \( h_{i+1} = h_i + \frac{p}{\eta} (x_{i+1} - z_{i+1}) \)
Endfor

Output \( x_{i+1} \)

Equivariant Lazy PnP  Recently a simple way of boosting the performance and stability of PnP methods has been proposed, namely the equivariant PnP (Terris et al., 2024). 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:

\[
D_g(x) = T_{g^{-1}} D(T_g x), \quad g \sim \mathcal{G}
\]  

Typical choices of the transforms include rotations, translations and reflections, etc. This scheme was in fact first found numerically in the work of Zhang et al (Zhang et al., 2021). In the work of Terris et al (Terris et al., 2024), this approach is formally analyzed and studied, demonstrating remarkable performance in stabilizing the 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 PnP-MS2G framework, we omit this to avoid redundant presentations.

3.3 Accelerating the Deep Unrolling Schemes via Operator Sketching

In this section, we will 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 (Adler and Öktem, 2018), with only subsets of it. The proposed network can be view as an unrolled version of stochastic PDHG (Chambolle et al., 2018) (but with ordered-subsets, and without variance-reduction). We partition the forward and adjoint operators into \( m \) subsets, and also the corresponding measurement data. In each layer, we use only one of the subsets, in a 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:

\[
[x^*, y^*] = \min_x \max_y \{ r(x) + \sum_{i=0}^{m-1} \langle M_i Ax, y_i \rangle - f_i^*(y_i) \}.
\]  

(14)
Figure 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 $[M_i b, M_i A x_k, y_k]$, while the primal subnet has 2 input channels for $[(M_i A)^T y_{k+1}, x_k]$

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

\[\text{Learned Stochastic Primal-Dual (LSPD)}\]

- Initialize $x_0 \in \mathbb{R}^d$, $y_0 \in \mathbb{R}^{n/m}$
- For $k = 0, 1, 2, ..., K - 1$
  - $i = \text{mod}(k, m)$;
  - (or pick $i$ from $[0, m - 1]$ uniformly at random)
  - $y_{k+1} = D_{\theta_k}^b(y_k, \sigma_k, (M_i A)x_k, M_i b)$;
  - $x_{k+1} = P_{\theta_k}^p(x_k, \tau_k, (M_i A)^T y_{k+1})$;

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 $m$-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 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 subnetworks across layers to be the same and train the subnetwork to perform denoising (Kamilov et al., 2017; Sun et al., 2019; Tang and Davies, 2020; Ono, 2017), artifact removal (Liu et al., 2020), or approximate projection to a image manifold (Rick Chang et al., 2017), leading to a plug-and-play (Venkatakrishnan et al., 2013; Romano et al., 2017; Reehorst and Schniter, 2018) type of approach with theoretical convergence guarantees.

Note that our LSPD network covers the SGD-Net of (Liu et al., 2021) as a special case, by setting the dual network to be a simple subtraction, and limit 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

\[\text{2. Alternatively, one may also consider an optional learned momentum acceleration by keeping the memory of the outputs of a number of previous layers: } x_{k+1} = P_{\theta_k}^p(X_k, \tau_k, (M_i A)^T y_{k+1}) \text{ where } X_k = [x_k, x_{k-1}, ..x_{k-M}], \text{ at the cost of extra computation and memory. For such case the input channel of the subnets would be } M + 1.\]
as the Learned SGD (LSGD) in this paper:

\[
\text{LSGD} \quad \text{Initialize } x_0 \in \mathbb{R}^d, y_0 \in \mathbb{R}^{n/m} \\
\text{For } k = 0, 1, 2, ..., K - 1 \\
\left\{ \\
\text{Pick } i \text{ from } [0, m - 1] \text{ uniformly at random} \\
y_{k+1} = M_i A x_k - M_i b; \\
x_{k+1} = P_{\theta_k^p}(x_k - \tau \cdot (M_i A)^T y_{k+1}); \\
\right. \\
\]

which is a stochastic variant of the ISTA-Net (Zhang and Ghanem, 2018). The stochastic unrolling can potentially give \( m \) fold reduction in the per-iteration complexity 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 \( A x_k, A^T y_{k+1} \):

\[
A x_k \approx A_{sk} S_{\theta_k^s}(x_k), \quad A^T y_{k+1} \approx U_{\theta_k^u}(A_{sk}^T y_{k+1}) (15)
\]

where \( S_{\theta_k^s}(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{d_{sk}} (d_{sk} < d) \) being the sketching/downsampling operator which can be potentially trainable w.r.t parameters \( \theta_k^s \), while \( A_{sk} \in \mathbb{R}^{n \times d_{sk}} \) is the sketched forward operator discretized on the reduced low-dimensional image space, and for the dual step we have \( U_{\theta_k^u} : \mathbb{R}^{d_{sk}} \rightarrow \mathbb{R}^d \) the upsampling operator which can also be trained. In practice, we found that it is actually suffice 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:

\[
\text{Sketched LPD} \quad \text{Initialize } x_0 \in \mathbb{R}^d, y_0 \in \mathbb{R}^n \\
\text{For } k = 0, 1, 2, ..., K - 1 \\
\left\{ \\
y_{k+1} = D_{\theta_k^d}(y_k, \sigma_k, A_{sk} S_{\theta_k^s}(x_k), b); \\
x_{k+1} = P_{\theta_k^p}(x_k, \tau_k, U_{\theta_k^u}(A_{sk}^T y_{k+1})); \\
\right. \\
\]

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

\[
(M_i A) x_k \approx (M_i A_{sk}) S_{\theta_k^s}(x_k), \\
(M_i A)^T y_{k+1} \approx U_{\theta_k^u}((M_i A_{sk})^T y_{k+1}), (16)
\]

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

\[
\text{SkLSPD(Option1)} \quad \text{Initialize } x_0 \in \mathbb{R}^d, y_0 \in \mathbb{R}^{n/m} \\
\text{For } k = 0, 1, 2, ..., K - 1 \\
\left\{ \\
i = \text{mod } (k, m); \\
\text{(or pick } i \text{ from } [0, m - 1] \text{ uniformly at random}) \\
y_{k+1} = D_{\theta_k^d}(y_k, \sigma_k, (M_i A_{sk}) S_{\theta_k^s}(x_k), M_i b); \\
x_{k+1} = P_{\theta_k^p}(x_k, \tau_k, U_{\theta_k^u}((M_i A_{sk})^T y_{k+1})); \\
\right. \\
\]
or alternatively:

\[
\text{SkLSPD (Option2)}
\]

- Initialize \( x_0 \in \mathbb{R}^d, y_0 \in \mathbb{R}^{n/m} \)

For \( k = 0, 1, 2, ..., K - 1 \)

\[
\begin{align*}
  i &= \text{mod} (k, m) ; \\
  (\text{or pick } i \text{ from } [0, m - 1] \text{ uniformly at random}) \\
  y_{k+1} &= D_{\theta_k}(y_k, \sigma_k, (M_iA_{s_k})S_{\theta_k}(x_k), M_ib) ; \\
  x_{k+1} &= U_{\theta_k}(P_{\theta_k}(S_{\theta_k}(x_k), \tau_k, (M_iA_{s_k})^T y_{k+1})) ;
\end{align*}
\]

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 last 4 unrolling layers to be unsketched, that is, \( A_{sk} = A \) for \( k > K_{\text{switch}} \)), and we found numerically that the reconstruction accuracy is best preserved if we implement this scheme.

Remark regarding the Option 2 for further improving efficiency

The second option of our SkLPD and SkLSPD further accelerates the computation comparing to Option 1, by making the primal-subnet taking 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 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 comparing to option 1.

4. Theoretical Analysis

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

4.1 General Assumptions

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

\[
x_{k+1} = P_{\theta}[(x_k - \eta_k \cdot U((M_k A_s)^T M_k A_s S(x_k) - (M_k A_s)^T M_k b))] \quad (17)
\]

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

\[
P_{\theta}(x) = e(x) + P_{\mathcal{M}}(x), \quad (18)
\]
where:
\[
P_M(x) := \arg \min_{z \in \mathcal{M}} \|x - z\|_2^2,
\]
and,
\[
\|e(x)\|_2 \leq \varepsilon_0, \quad \forall x \in \mathbb{R}^d.
\]
Here we model the denoiser to be a $\varepsilon$-projection towards a manifold. Note that in practice the image manifold $\mathcal{M}$ typically form a non-convex subset. We also make conditions on the image manifold as the following:

**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 restricted eigenvalue (restricted strong-convexity) condition which is necessary for robust recovery (Negahban et al., 2012; Agarwal et al., 2012; Wainwright, 2014; Oymak et al., 2017):

**A. 3 (Restricted Eigenvalue Condition)** We define a descent cone $C$ at point $x^\dagger$ as:
\[
C := \left\{ v \in \mathbb{R}^d \mid v = a(x - x^\dagger), \exists a \geq 0, x \in \mathcal{M} \right\},
\]
and the restricted strong-convexity constant $\mu_c$ to be the largest positive constant satisfies the following:
\[
\frac{1}{n} \|Av\|_2^2 \geq \mu_c \|v\|_2^2, \quad \forall v \in C.
\]
and the restricted smoothness constant $L_c$ to be the smallest positive constant satisfies:
\[
\frac{1}{q} \|M_i Av\|_2^2 \leq L_c \|v\|_2^2, \quad \forall v \in C. \forall i \in [m]
\]
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, such type of condition must hold (Oymak et al., 2017; Agarwal et al., 2012; Chandrasekaran et al., 2012). For example, in sparse-recovery setting, when the measurement operator is a Gaussian map (compressed-sensing measurements) and $x^\dagger$ is $s$-sparse, one can show that $\mu_c$ can be as large as $O(1 - s \log d / n)$ (Oymak et al., 2017). In our setting we would expect an even better $\mu_c$, 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:
\[
\frac{1}{q} \|M_i Av\|_2^2 \leq L_s \|v\|_2^2, \quad \forall v \in \mathbb{R}^d. \forall i \in [m],
\]
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 $x^\dagger$:
\[
\mathbb{E}\|x_{N_k} - x^\dagger\|_2 \leq \alpha^{N_k} \|x_{\text{init}} - x^\dagger\|_2 + \frac{1 - \alpha^{N_k}}{1 - \alpha} (\varepsilon + \delta),
\]

14
where \( x_{\text{init}} \) denotes the initial point of \( k \)-th stage of PnP-MS2G, \( \alpha = \kappa (1 - \frac{\mu_c}{L_f}) \), \( \kappa = 1 \) 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,

\[
\delta := 2\eta \sup_{v \in \mathcal{C} \cap B, i \in [m]} \| A v A^T M_i w \|/
\]

\[
\| M_i A_{sk} \|_2 \| M_i A_{sk} D(x_k) - M_i A x_k \|_2 \leq \varepsilon_1, \forall i, k
\]

\[
\| \mathcal{U} (M_i A_{sk})^T y_k - (M_i A)^T y_k \|_2 \leq \varepsilon_2, \forall i, k
\]

(26)

When the restricted eigenvalue \( \mu_c \) is large enough such that \( \alpha < 1 \), the PnP-MS2G has a linear convergence in estimation error, up to \( \frac{\varepsilon}{1-\alpha} \) only depending the approximation accuracy of the denoiser 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 in Theorem 1 apply for both sketched algorithms and the full-batch counterpart. This suggests the tremendous potential of computational saving of PnP-MS2G over classical methods.

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

One the other hand, using 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:

\[
\mathbb{E} \| x_{N_k} - x^\dagger \|_2 \geq (1 - \gamma)^N_k (1 - \frac{L_c}{L_s})^N_k \| x_{\text{init}} - x^\dagger \|_2 - \frac{L_s}{L_c} \varepsilon^*_k
\]

(27)

where \( \varepsilon^* = \varepsilon_0 + \kappa \eta \varepsilon_1 + \kappa \eta \varepsilon_2 \).

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

5. Numerical Experiments

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 by sparse-view CT reconstruction tasks. Here we compare our PnP-MS2G with PnP-PGD and its stochastic variant PnP-SGD (Sun et al., 2019). For our PnP-MS2G we perform a \( 4 \times \) downscale at the first 50 iterations, then a \( 2 \times \) downscale afterwards, leading to significant improvement on computational efficiency. We choose here the famous BM3D (Dabov et al., 2007, 2008) 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 comparing to PnP-PGD and PnP-SGD.

In Figure 3, 4, and 6 we present numerical results for our Lazy-Denoiser 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, comparing to standard equivariant PnP scheme (Terris et al., 2024) in image superresolution task. Here we seek to perform \( 4 \times \) superresolution for low-resolution color images. The setting
Figure 2: Example for applying MS2G (minibatched) in X-ray CT reconstruction, comparing to the PnP stochastic gradient descent (PnP-SGD) proposed by Sun et al (Sun et al., 2019). 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.

of this experiment is the same as the experiment in (Terris et al., 2024), with The interpolation kernel $h$ for the task being the Guassian kernel of standard deviation 1:

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

(28)

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

In the X-ray CT experiment we presented in Figure 6, we implement PnP-SGD with or without the Lazy-Denoiser scheme. We can observe that, our Lazy-PnP has the same convergence rates comparing to vanilla PnP-SGD, while only require to compute the denoiser on 20% of the iteration.
Figure 3: (Lazy-Denoiser) Example for applying our Lazy-Denoiser scheme with equivariant-PnP for image superresolution (4×). The denoiser network we choose here is DnCNN. Here we show the reconstructed image at 2000-th iteration.

5.2 Training of LSPD and SkLSPD

5.2.1 Supervised End-to-End Training

The most basic training approach for LSPD/SkLSPD is the end-to-end supervised training where we consider fully paired training samples of measurement and the “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 $\theta$ be the set of parameters $\theta := \{\theta^k_p, \theta^k_d, \tau_k, \sigma_k\}_{k=0}^{K-1}$, applying the LSPD/SkLSPD network on some measurement $b$ can be written as $F_\theta(b)$, the training objective can typically be written as:

$$\theta^* \in \arg \min_\theta \sum_{i=1}^N \|x^\dagger_i - F_\theta(b_i, x^0_i)\|_2^2,$$

(29)
where we denote by $N$ the number of paired training examples. Since the paired high-quality ground-truth data is expensive and idealistic in practice, we would prefer more advanced methods for training which have relaxed requirements on available training data. In Appendix, we present unsupervised training results for the unrolling networks using only measurement data.

In this subsection we present numerical results of our proposed networks for low-dose X-ray CT. In real world clinical practice, the low dosage CT is widely used and highly recommended, since the intense exposures to the X-ray could significantly increase the risk of inducing cancers. The low-dose CT takes a large number of low-energy X-ray views, leading to huge volumes of noisy measurements. This makes the reconstruction schemes struggle to achieve efficient and accurate estimations. In our X-ray CT experiments we use the standard Mayo-Clinic dataset (McCollough, 2016) which contains 10 patients’ 3D CT scans. We use 2111 slices (from 9 patients) of 2D images sized $512 \times 512$ for training and 118 slices of the remaining 1 patient for testing. We use the ODL toolbox (Adler and Öktem, 2018) to simulate fan beam projection data.
Figure 5: **Lazy-Denoiser** Example for applying our Lazy-Denoiser scheme with equivariant-PnP for image superresolution (4 ×). The denoiser network we choose here is DnCNN. Here we show the reconstructed image at 2000-th iteration.

with 800 equally-spaced angles of views (each view includes 800 rays). The fan-beam CT measurement is corrupted with Poisson noise: \( b \sim \text{Poisson}(I_0e^{-Ax}) \), where we make a 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.

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\(^3\) 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

---

\(^3\) 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.
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. We train all the networks with 50 epochs of Adam optimizer (Kingma and Ba, 2015) with batch size 1, in the supervised manner.

For our SkLSPD we choose the Option 2 presented in our 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 up/down-samplers in our Sketched LSPD, we simply choose the bilinear upsample and downsample functions in Pytorch. When called, the up-sampler increase the input image 4 times larger (from $256 \times 256$ to $512 \times 512$), while the down-sampler makes the input image 4 times smaller (from $512 \times 512$ to $256 \times 256$). While the full forward operator $A$ is defined on the grid of $512 \times 512$, the sketched operator $A_s$ is defined on the grid of $256 \times 256$ hence requires only a 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 the SkLSPD with a light-weight dual-subnetwork (corresponding to a proximal operator of a weighted $\ell_2$ loss with learnable weights, see the SkLSPD-LW in Section IV).
Figure 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 |

In addition, we also implement the Learned SGD (Liu et al., 2021) 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 by their original design the LSGD subnetworks 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 learnable parameters is similar to our LSPD.

We present the performance of the LPD, LSPD, and SkLSPD on 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, due to the challenging extreme low-dose setting, the FBP reconstruction fails completely. This can be partially addressed by U-Net postprocessing (FBPConvNet, Jin et al., 2017), whose parameter size...
Figure 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.

is one order of magnitude larger than our unrolling networks. Next we turn to the learned reconstruction results. From the numerical results, we found out that our LSPD and SkLSPD networks both achieve almost the same reconstruction accuracy compare to LPD baseline in terms of PSNR (peak signal-to-noise ratio) and SSIM (structural similarity index, (Wang et al., 2004)) measures, with requiring only a fraction of the computation of the forward and adjoint operators. In terms of run time on GPU, our acceleration can introduce a reduction of around 40% to 60% compared to the full batch LPD.

In our Table II, we present additional results on another widely applied modality in clinical practice – the sparse-view CT, where we take fewer amount of normal-dose measurements. Here we use again 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: \( b \sim \text{Poisson}(I_0 e^{-Ax}) \), where we make a normal-dose choice of \( I_0 = 7 \times 10^6 \). Different to the low-dose CT, the main challenge of
sparse-view CT is the ill-posedness of the inverse problems, that the measurement operator is highly under-determined with a non-trivial null-space. Meanwhile we also present in the Appendix our numerical results on instance-adaptation in out-of-distribution reconstruction for sparse-view CT, demonstrating again the effectiveness of our approach.

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 have done recovery analysis for such frameworks for the first time and perform 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 for mitigating the computational overheads of advanced denoisers for plug-and-play methods. We have provided theoretical analysis of the proposed framework for the estimation guarantees from the viewpoint of stochastic optimization theory. Then we provide 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 emphasis here that this framework can be readily applied to accelerate newer algorithmic schemes such as deep restoration prior (Hu et al., 2023) and deep equilibrium models (Gilton et al., 2021) and generative conditional sampling (Cai et al., 2024).

7. Appendix

7.1 Proof of Theorem 3.1

In this proof we utilize several projection identities from (Oymak et al., 2017). We list them here first for completeness. The first one would be the cone-projection identity:

$$\| P_C(x) \|_2 = \sup_{v \in C \cap B^4} v^T x,$$

(30)
where $B^d$ 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:

$$P_M(x + v) - x = P_{M-v}(v).$$

(31)

Now if $0 \in M - x$, we can have the third projection identity which is an important result from geometric functional analysis (Oymak et al., 2017, Lemma 18):

$$\|P_D(x)\|_2 \leq \kappa_D \|P_C(x)\|_2,$$

(32)

where:

$$\kappa_D = \begin{cases} 
1 & \text{if } D \text{ is convex} \\
2 & \text{if } D \text{ is non-convex} 
\end{cases}$$

(33)

where $D$ is a potentially non-convex closed set included by cone $C$. On the other hand, utilizing a simplified result of (Gower et al., 2019) with partition minibatches, we have:

$$\mathbb{E}_S(\|A^T M^T MA(x - z)\|_2^2)$$

$$\leq 2L_s\left(\frac{q^2}{2n}\|Ax - b\|_2^2 - \frac{q^2}{2n}\|Az - b\|_2^2 - q^2(\nabla f(z), x - z)\right).$$

(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:

$$\mathbb{E}_S(\|A^T M^T MA(x - z)\|_2^2)$$

$$\leq 2L_s\left(\frac{q^2}{2n}\|Ax - b\|_2^2 - \frac{q^2}{2n}\|Az - b\|_2^2$$

$$- q^2\left(\frac{1}{n}A^T(Az - b), x - z\right)\right)$$

$$\leq \frac{q^2 L_s}{n}\left(\frac{1}{2}\|Ax - x^\dagger\|_2^2 - \|w\|_2^2 - \|w\|_2^2 + \langle w, A(x - x^\dagger) \rangle\right)$$

$$= \frac{q^2 L_s}{n}\|A(x - x^\dagger)\|_2^2$$

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

$$\|M_iA_{s_k}\|_2\|M_iA_{s_k}D(x_k) - M_iAx_k\|_2 \leq \varepsilon_1$$

$$\|U(M_iA_{s_k})^Ty_k - (M_iA)^Ty_k\|_2 \leq \varepsilon_2, \forall i, k$$

(35)

Denoting:

$$y_k := M_iA_{s_k}D(x_k) - M_ib,$$

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

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

$$= \|P_{\theta_p}(x_k - \tau U((M_iA_{s_k})^Ty_k) - x^\dagger\|_2$$

$$\leq \|P_{M}(x_k - \tau U((M_iA_{s_k})^Ty_k)) - x^\dagger\|_2$$

$$+ \|e(x_k - \tau U((M_iA_{s_k})^Ty_k))\|_2$$

$$= \|P_{M - x^\dagger}(x_k - x^\dagger - \tau U((M_iA_{s_k})^Ty_k))\|_2 + \|e(x_k)\|_2,$$
Then we can continue:

\[ \| x_{k+1} - x^\dagger \|_2 \leq \kappa \| P_C(x_k - x^\dagger - \tau U((M_iA_{sk})^T y_{k+1})) \|_2 + \| e(x_k) \|_2 \]

\[ = \kappa \sup_{v \in C \cap B} v^T (x_k - x^\dagger - \tau U((M_iA_{sk})^T y_{k+1}) + \| e(x_k) \|_2 \]

\[ \leq \kappa \sup_{v \in C \cap B} v^T (x_k - x^\dagger - \tau A^T M_i^T (M_iA x_k - M_i b)) + \varepsilon \]

\[ = \kappa \sup_{v \in C \cap B} v^T [x_k - x^\dagger - \tau A^T M_i^T (M_iA x_k - M_i b)] + \varepsilon \]

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

\[ \mathbb{E}((x_{k+1} - x^\dagger)^2) \leq \kappa \mathbb{E}((I - \tau A^T M_i^T M_i A)(x_k - x^\dagger)) + \varepsilon \]

\[ + 2\tau \mathbb{E} \sup_{v \in C \cap B} v^T A^T M_i^T M_i w \]

\[ \leq \kappa \mathbb{E}((I - \tau A^T M_i^T M_i A)(x_k - x^\dagger))^{\frac{1}{2}} + \varepsilon \]

\[ + 2\tau \mathbb{E} \sup_{v \in C \cap B} v^T A^T M_i^T M_i w \]

\[ = \kappa \{ \mathbb{E}(\| x_k - x^\dagger \|_2^2 - 2\tau \| M_i A(x_k - x^\dagger) \|_2^2 \]

\[ + \tau^2 \| A^T M_i^T M_i A(x_k - x^\dagger) \|_2^2 \}^{\frac{1}{2}} + \varepsilon + 2\tau \mathbb{E} \sup_{v \in C \cap B} v^T A^T M_i^T M_i w \]

Now denoting:

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

(36)
and then:

\[
\mathbb{E} (\| x_{k+1} - x^\dagger \|_2 ) \\
\leq \kappa \{(\| x_k - x^\dagger \|_2^2 - 2\frac{\tau qv_b}{n} \| A(x_k - x^\dagger) \|_2^2 \\
+ \frac{\tau^2 q^2 L_s v_a}{n} \| A(x_k - x^\dagger) \|_2^2 \}^{\frac{1}{2}} + \varepsilon + \delta \}
\]

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

\[
\mathbb{E} (\| x_{k+1} - x^\dagger \|_2 ) \\
\leq \kappa \{(\| x_k - x^\dagger \|_2^2 - (2\tau q \mu c v_b - 2L_s \mu c \tau^2 q^2 v_a) \| x_k - x^\dagger \|_2^2 \}^{\frac{1}{2}} + \varepsilon + \delta \}
\]

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

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

### 7.2 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 known result provided in (Oymak and Hassibi, 2016, 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 \( \| P_\mathcal{C} (v) \|_2 \geq a \| v \|_2 \) and \( \| v \|_2 \leq c \), we can have:

\[
\frac{\| P_\mathcal{D} (v) \|_2}{\| P_\mathcal{C} (v) \|_2} \geq 1 - \gamma.
\]

(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:

\[
\| x_{k+1} - x^\dagger \|_2 \\
= \| P_{\theta_p} (x_k - \tau U (A_{sk}^T M_i^T y_k)) - x^\dagger \|_2 \\
\geq \| P_{\mathcal{M}} (x_k - \tau U (A_{sk}^T M_i^T y_k)) - x^\dagger \|_2 \\
- \| e (x_k - \tau U (A_{sk}^T M_i^T y_k)) \|_2 \\
= \| P_{\mathcal{M}} (x_k - x^\dagger - \tau U (A_{sk}^T M_i^T y_k)) \|_2 \\
- \| e (x_k - \tau U (A_{sk}^T M_i^T y_k)) \|_2.
\]

26
Now due to (37) we can continue:

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

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

\[
\|(I - \tau A^T M_i^T M_i A)(x_k - x^\dagger)\|_2 \leq (1 + \tau q L_s)\|x_k - x^\dagger\|_2,
\]  
and also note the second part of restricted eigenvalue condition we have:

\[
\|M_i A (x_k - x^\dagger)\|_2 \leq q L_c \|x_k - x^\dagger\|_2.
\]  

Hence:

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

Combining these three with \( \tau = \frac{1}{q L_s} \), 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 \( a = \frac{L_s - L_c}{2 L_s} \), we can write:

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

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

Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

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

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

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pages 3981–3989, 2016.

Haim Avron, Vikas Sindhwani, and David Woodruff. Sketching structured matrices for faster nonlinear regression. In *Advances in neural information processing systems*, pages 2994–3002, 2013.

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

Thorsten M Buzug. Computed tomography. In *Springer handbook of medical technology*, pages 311–342. Springer, 2011.

Ziruo Cai, Junqi Tang, Subhadip Mukherjee, Jinglai Li, Carola-Bibiane Schönlieb, and Xiaoqun Zhang. Nf-ula: Normalizing flow-based unadjusted langevin algorithm for imaging inverse problems. *SIAM Journal on Imaging Sciences*, 17(2):820–860, 2024.

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

Antonin Chambolle and Thomas Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25:161–319, 2016.

Antonin Chambolle, Matthias J Ehrhardt, Peter Richtarik, and Carola-Bibiane Schönlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM Journal on Optimization*, 28(4):2783–2808, 2018.

Antonin Chambolle, Claire Delplancke, Matthias J Ehrhardt, Carola-Bibiane Schönlieb, and Junqi Tang. Stochastic primal–dual hybrid gradient algorithm with adaptive step sizes. *Journal of Mathematical Imaging and Vision*, 66(3):294–313, 2024.

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

Regev Cohen, Michael Elad, and Peyman Milanfar. Regularization by denoising via fixed-point projection (red-pro). *SIAM Journal on Imaging Sciences*, 14(3):1374–1406, 2021.

K Dabov, A Foi, V Katkovnik, and K Egiazarian. 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.
Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image restoration by sparse 3d transform-domain collaborative filtering. In *Image Processing: Algorithms and Systems VI*, volume 6812, page 681207. International Society for Optics and Photonics, 2008.

A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

Derek Driggs, Junqi Tang, Jingwei Liang, Mike Davies, and Carola-Bibiane Schönlieb. A stochastic proximal alternating minimization for nonsmooth and nonconvex optimization. *SIAM Journal on Imaging Sciences*, 14(4):1932–1970, 2021.

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

Karen Egiazarian, Alessandro Foi, and Vladimir Katkovnik. Compressed sensing image reconstruction via recursive spatially adaptive filtering. In *2007 IEEE International Conference on Image Processing*, volume 1, pages I–549. IEEE, 2007.

Hakan Erdogan and Jeffrey A Fessler. Ordered subsets algorithms for transmission tomography. *Physics in Medicine & Biology*, 44(11):2835, 1999.

Davis Gilton, Gregory Ongie, and Rebecca Willett. Deep equilibrium architectures for inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 7:1123–1133, 2021.

Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. SGD: General analysis and improved rates. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5200–5209, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010.

Yuyang Hu, Mauricio Delbracio, Peyman Milanfar, and Ulugbek S Kamilov. A restoration network as an implicit prior. *arXiv preprint arXiv:2310.01391*, 2023.

Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

Ulugbek S Kamilov, Hassan Mansour, and Brendt Wohlberg. A plug-and-play priors approach for solving nonlinear imaging inverse problems. *IEEE Signal Processing Letters*, 24(12):1872–1876, 2017.

Davood Karimi and Rabab K Ward. A hybrid stochastic-deterministic gradient descent algorithm for image reconstruction in cone-beam computed tomography. *Biomedical Physics & Engineering Express*, 2(1):015008, 2016.
Donghwan Kim, Sathish Ramani, and Jeffrey A Fessler. Combining ordered subsets and momentum for accelerated x-ray ct image reconstruction. *IEEE transactions on medical imaging*, 34(1):167–178, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proceedings of 3rd International Conference on Learning Representations*, 2015.

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

Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *arXiv preprint arXiv:1507.02000*, 2015.

Yanjun Li, Kai Zhang, Jun Wang, and Sanjiv Kumar. Learning adaptive random features. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4229–4236, 2019.

Jiaming Liu, Yu Sun, Cihat Eldeniz, Weijie Gan, Hongyu An, and Ulugbek S Kamilov. Rare: Image reconstruction using deep priors learned without groundtruth. *IEEE Journal of Selected Topics in Signal Processing*, 14(6):1088–1099, 2020.

Jiaming Liu, Yu Sun, Weijie Gan, Xiaojian Xu, Brendt Wohlberg, and Ulugbek S Kamilov. Sgd-net: Efficient model-based deep learning with theoretical guarantees. *IEEE Transactions on Computational Imaging*, 7:598–610, 2021.

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

Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, and Peter Richtárik. Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! In *International Conference on Machine Learning*, pages 15750–15769. PMLR, 2022.

Subhadip Mukherjee, Sören Dittmer, Zakhar Shumaylov, Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. Learned convex regularizers for inverse problems. *arXiv preprint arXiv:2008.02839*, 2020.

Subhadip Mukherjee, Marcello Carioni, Ozan Öktem, and Carola-Bibiane Schönlieb. End-to-end reconstruction meets data-driven regularization for inverse problems. *arXiv preprint arXiv:2106.03538*, 2021.

Sahand N Negahban, Pradeep Ravikumar, Martin J Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, pages 538–557, 2012.

Y. Nesterov. Gradient methods for minimizing composite objective function. Technical report, UCL, 2007.

John M Ollinger and Jeffrey A Fessler. Positron-emission tomography. *Ieee signal processing magazine*, 14(1):43–55, 1997.

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

Samet Oymak and Babak Hassibi. Sharp mse bounds for proximal denoising. *Foundations of Computational Mathematics*, 16(4):965–1029, 2016.
Samet Oymak, Benjamin Recht, and Mahdi Soltanolkotabi. Sharp time–data tradeoffs for linear inverse problems. *IEEE Transactions on Information Theory*, 64(6):4129–4158, 2017.

Evangelos Papoutsellis, Zeljko Kereta, and Kostas Papafitsoros. Why do we regularise in every iteration for imaging inverse problems? *arXiv preprint arXiv:2411.00688*, 2024.

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

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

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

Edward T Reehorst and Philip Schniter. Regularization by denoising: Clarifications and new interpretations. *IEEE Transactions on Computational Imaging*, 5(1):52–67, 2018.

JH Rick Chang, Chun-Liang Li, Barnabas Poczos, BVK Vijaya Kumar, and Aswin C Sankaranarayanan. One network to solve them all—solving linear inverse problems using deep projection models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5888–5897, 2017.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

Nicolas L. Roux, Mark Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2663–2671. Curran Associates, Inc., 2012.

Jian Sun, Huibin Li, Zongben Xu, et al. Deep admm-net for compressive sensing mri. *Advances in neural information processing systems*, 29, 2016.

Yu Sun, Brendt Wohlberg, and Ulugbek S Kamilov. An online plug-and-play algorithm for regularized image reconstruction. *IEEE Transactions on Computational Imaging*, 2019.

Yu Sun, Zihui Wu, Brendt Wohlberg, and Ulugbek S Kamilov. Scalable plug-and-play admm with convergence guarantees. *arXiv preprint arXiv:2006.03224*, 2020.

Julián Tachella, Junqi Tang, and Mike Davies. The neural tangent link between cnn denoisers and non-local filters. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Hong Ye Tan, Subhadip Mukherjee, Junqi Tang, and Carola-Bibiane Schönlieb. Provably convergent plug-and-play quasi-newton methods. *SIAM Journal on Imaging Sciences*, 17(2):785–819, 2024.

Junqi Tang and Mike Davies. A fast stochastic plug-and-play admm for imaging inverse problems. *arXiv preprint arXiv:2006.11630*, 2020.
Junqi Tang, Mohammad Golbabaee, and Mike Davies. Exploiting the structure via sketched gradient algorithms. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1305–1309. IEEE, 2017a.

Junqi Tang, Mohammad Golbabaee, and Mike E. Davies. Gradient projection iterative sketch for large-scale constrained least-squares. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3377–3386. PMLR, 2017b.

Junqi Tang, Mohammad Golbabaee, Francis Bach, and Mike E. Davies. Rest-katyusha: Exploiting the solution's structure via scheduled restart schemes. In *Advances in Neural Information Processing Systems 31*, pages 427–438. Curran Associates, Inc., 2018.

Junqi Tang, Karen Egiazarian, and Mike Davies. 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)*, pages 7680–7684. IEEE, 2019.

Junqi Tang, Karen Egiazarian, Mohammad Golbabaee, and Mike Davies. The practicality of stochastic optimization in imaging inverse problems. *IEEE Transactions on Computational Imaging*, 6:1471–1485, 2020.

Junqi Tang, Subhadip Mukherjee, and Carola-Bibiane Schönlieb. Stochastic primal-dual deep unrolling. *arXiv preprint arXiv:2110.10093*, 2021.

Junqi Tang, Subhadip Mukherjee, and Carola-Bibiane Schönlieb. Accelerating deep unrolling networks via dimensionality reduction. *arXiv preprint arXiv:2208.14784*, 2022.

Junqi Tang, Subhadip Mukherjee, and Carola-Bibiane Schönlieb. Iterative operator sketching framework for large-scale imaging inverse problems. *hal-04701732*, 2024.

Matthieu Terris, Thomas Moreau, Nelly Pustelnik, and Julian Tachella. Equivariant plug-and-play image reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25255–25264, 2024.

Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.

Marinus T Vlaardingerbroek and Jacques A Boer. *Magnetic resonance imaging: theory and practice*. Springer Science & Business Media, 2013.

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

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

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

Blake E Woodworth and Nati Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in neural information processing systems*, pages 3639–3647, 2016.
Jinxi Xiang, Yonggui Dong, and Yunjie Yang. Fista-net: Learning a fast iterative shrinkage thresholding network for inverse problems in imaging. *IEEE Transactions on Medical Imaging*, 40(5):1329–1339, 2021.

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

Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1828–1837, 2018.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6360–6376, 2021.