Bagged Deep Image Prior for Recovering Images in the Presence of Speckle Noise

Xi Chen [1] **Zhewen Hou** [2] **Christopher A. Metzler** [3] **Arian Maleki** [2] **Shirin Jalali** [1]

Abstract

We investigate both the theoretical and algorithmic aspects of likelihood-based methods for recovering a signal from multiple sets of complex-valued measurements, referred to as looks, affected by speckle (multiplicative) noise. Our theoretical contributions include establishing the first theoretical upper bound on the Mean Squared Error (MSE) of the maximum likelihood estimator under the deep image prior hypothesis. Our results capture the dependence of MSE on the number of parameters in the deep image prior, the number of looks, the signal dimension, and the number of measurements per look. On the algorithmic side, we introduce the concept of bagged Deep Image Prior (Bagged-DIP) and integrate it with projected gradient descent. Furthermore, we show how employing the Newton-Schulz algorithm for calculating matrix inverses within the iterations of projected gradient descent reduces the computational complexity of the algorithm. We show that this method achieves state-of-the-art performance. Code is available at https://github.com/Computational-Imaging-RU/Bagged-DIP-Speckle.

1. Introduction

One of the most fundamental and challenging issues faced by many coherent imaging systems is the presence of speckle noise. An imaging system with "fully-developed" speckle noise can be modeled as

$$\mathbf{y} = AX_o\mathbf{w} + \mathbf{z}. \tag{1}$$

Here, $X_o = \mathrm{diag}(\mathbf{x}_o)$, where $\mathbf{x}_o \in \mathbb{C}^n$ denotes the complex-valued signal of interest. $\mathbf{w} \in \mathbb{C}^n$ represents speckle (or multiplicative) noise, where $w_1, \dots, w_n$ are independent and identically distributed (iid) $\mathcal{CN}(0, \sigma_w^2 I_n)$, and finally $\mathbf{z} \in \mathcal{C}^m$ denotes the additive noise, often caused by the sensors, is modeled as iid $\mathcal{CN}(0, \sigma_z^2)$. In this paper, we explore the scenario where $m \leq n$, allowing imaging systems to capture higher resolution images than constrained by the number of sensors.[1]

As is clear from (1), the multiplicative nature of the speckle noise poses a challenge in extracting accurate information from measurements, especially when the measurement matrix A is ill-conditioned. To alleviate this issue, many practical systems employ a technique known as multilook or multishot (Argenti et al., 2013; Bate et al., 2022). Instead of taking a single measurement of the image, multilook systems capture multiple measurements, aiming for each group of measurements to have independent speckle and additive noise. In an L look system, the measurements captured at look ℓ, $\ell = 1, \dots, L$, can be represented as

$$\mathbf{y}_\ell = AX_o\mathbf{w}_\ell + \mathbf{z}_\ell,$$

where, $\mathbf{w}_1, \dots, \mathbf{w}_L \in \mathbb{C}^n$ and $\mathbf{z}_1, \dots, \mathbf{z}_L \in \mathbb{C}^m$ denote the indepdent speckle noise and additive noise vectors, respectively. In this model, we have assumed that the measurement kernel A remains constant across the looks. This assumption holds true in multilooking for several imaging systems, such as when the sensors' locations change slightly for different looks.

Since fully-developed noises are complex-valued Gaussian and have uniform phases, the phase of $\mathbf{x}_o$ cannot be recovered. Hence, the goal of a multilook system is to obtain a precise estimate of $|\mathbf{x}_o|$ based on the L observations $\{\mathbf{y}_1, \dots, \mathbf{y}_L\}$, given the measurement matrix A. (Here, $|\cdot|$ denotes the element-wise absolute value operation.) Therefore, since the phase of $\mathbf{x}_o$ is not recoverable, in the rest of the paper, we assume that $\mathbf{x}_o$ is real-valued.

A standard approach for estimating $\mathbf{x}_o$ is to minimize the negative log-likelihood function subject to the signal structure constraint. More precisely, in a constrained-likelihood-

[1]Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, NJ, USA [2]Department of Statistics, Columbia University, NY, USA [3]Department of Computer Science, University of Maryland, College Park, MD, USA. Correspondence to: Shirin Jalali <shirin.jalali@rutgers.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]Considering $m < n$ for simpler imaging systems (with no speckle noise) has led to the development of the fields of compressed sensing and compressive phase retrieval.

based approach, one aims to solve the following optimization problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{C}} f_L(\mathbf{x}), \tag{2}$$

where $\mathcal{C}$ represents the set encompassing all conceivable images and $f_L(\mathbf{x})$ is defined as:

$$f_L(\mathbf{x}) = \log\det(B(\mathbf{x})) + \frac{1}{L}\sum_{\ell=1}^{L} \tilde{\mathbf{y}}_\ell^T (B(\mathbf{x}))^{-1} \tilde{\mathbf{y}}_\ell, \tag{3}$$

where

$$B(\mathbf{x}) = \begin{bmatrix} \sigma_z^2 I_n + \sigma_w^2 \Re(U(\mathbf{x})) & -\sigma_w^2 \Im(U(\mathbf{x})) \\ \sigma_w^2 \Im(U(\mathbf{x})) & \sigma_z^2 I_n + \sigma_w^2 \Re(U(\mathbf{x})) \end{bmatrix},$$

and $\tilde{\mathbf{y}}_\ell^T = \begin{bmatrix} \Re(\mathbf{y}_\ell^T) & \Im(\mathbf{y}_\ell^T) \end{bmatrix}$, with $X = \mathrm{diag}(\mathbf{x})$ and

$$U(\mathbf{x}) = AX^2\bar{A}^T.$$

Here, $\Re(\cdot)$ and $\Im(\cdot)$ denote element-wise real and imaginary parts, respectively. (Appendix B presents the derivation of the log likelihood function and its gradient.)

It is important to note that the set $\mathcal{C}$ in (2) is not known explicitly in practice. Hence, in this paper we work with the following hypothesis that was put forward in (Ulyanov et al., 2018; Heckel & Hand, 2018).

- **Deep image prior (DIP) hypothesis** (Ulyanov et al., 2018; Heckel & Hand, 2018): Natural images can be embedded within the range of untrained neural networks that have substantially fewer parameters than the total number of pixels, and use iid noises as inputs.

Inspired by this hypothesis, we define $\mathcal{C}$ as the range of a deep image prior. More specifically, we assume that for every $\mathbf{x} \in \mathcal{C}$, there exists $\boldsymbol{\theta} \in \mathbb{R}^k$ such that $\mathbf{x} \approx g_{\boldsymbol{\theta}}(\mathbf{u})$, where $\mathbf{u}$ is generated iid $\mathcal{N}(0,1)$, and $\boldsymbol{\theta} \in \mathbb{R}^k$ denotes the parameters of the DIP neural network. There are two main challenges that we address:

- Theoretical challenge: Assuming that we can solve the optimization problem (2) under the DIP hypothesis, the following question arises: Can we theoretically characterize the corresponding reconstruction quality? Moreover, what is the relationship between the reconstruction error and key parameters such as k (the number of parameters of the DIP neural network), m, n and L? Specifically, in the scenario where the scene is static, and we can acquire as many looks as necessary, what is the achievable level of accuracy?
- Practical challenge: Given the challenging nature of the likelihood and the DIP hypothesis, can we design a computationally-efficient algorithm for solving (2) under the DIP hypothesis?

Here is a summary of our contributions:

On the theoretical front, we establish the first theoretical result on the performance of multilook coherent imaging systems. These findings unveil intriguing characteristics of such imaging systems. Notably, we demonstrate that with a large number of looks L, these systems deliver highly precise reconstructions when $m^2 = O(k \log n)$. In Section 3, we elaborate on the innovative methodologies underpinning our theoretical results.

On the practical side, we start with vanilla projected gradient descent (PGD) (Lawson & Hanson, 1995), which faces two challenges diminishing its effectiveness on this problem:

Challenge 1: As will be described in Section 4.2, in the PGD, the signal to be projected on the range of $g_{\boldsymbol{\theta}}(\mathbf{u})$ is burried in "noise". Hence, DIPs with large number of parameters will overfit to the noise and will not allow the PGD algorithm to obtain a reliable estimate (Heckel & Hand, 2018; Heckel & Soltanolkotabi, 2019). On the other hand, the low accuracy of simpler DIPs becomes a bottleneck as the algorithm progresses through iterations, limiting the overall performance. To alleviate this issue, we propose **Bagged-DIP**. This is a simple idea with roots in classical literature of ensemble methods (Breiman, 1996). Bagged-DIP idea enables us to use complex DIPs at every iteration and yet obtain accurate results.

Challenge 2: As will be clarified in Section 4.1, PGD requires the inversion of large matrices at every iteration, which is a computationally challenging problem. We alleviate this issue by using the Newton-Schulz algorithm (Schulz, 1933), and empirically demonstrating that **only one** step of this algorithm is sufficient for the PGD algorithm. This significantly reduces the computational complexity of each iteration of PGD.

2. Related Work

Eliminating speckle noise has been extensively explored in the literature (Lim & Nawab, 1980; Gagnon & Jouan, 1997; Tounsi et al., 2019). Current technology relies on gathering enough measurements to ensure the invertibility of matrix A and subsequently inverting A to represent the measurements in the following form: $\mathbf{y}_\ell = X\mathbf{w}_\ell + \mathbf{z}_\ell$. However, as matrix A deviates from the identity, the elements of the vector $\mathbf{z}$ become dependent. In practice, these dependencies are often overlooked, simplifying the likelihood. This simplification allows researchers to leverage various denoising methods, spanning from classical low-pass filtering to application of convolutional neural networks (Tian et al., 2020) and transformers (Fan et al., 2022). A series of papers have considered the impact of the measurement kernel in the

algorithms. By using single-shot digital holography, the authors in (Pellizzari et al., 2017; 2018) develop heuristic method to obtain maximum a posteriori estimate of the real-valued speckle-free object reflectance. They later extend this method to handle multi-shot measurements and incorporate more accurate image priors (Pellizzari et al., 2020; 2022; Bate et al., 2022). While these methods can work with non-identity A's, they still require A to be well-conditioned.

Our paper is different from the existing literature, mainly because we study scenarios where the matrix A is undersampled ($m < n$). In a few recent papers, researchers have explored similar problems (Zhou et al., 2022; Chen et al., 2023). The paper (Chen et al., 2023) aligns closely in scope and approach with our work. The authors addressed a similar problem, albeit assuming real-valued measurements and noises, and advocated for the use of DIP-based PGD. Addressing the concerns highlighted in the last section (further elucidated in Section 5.2), our Bagged-DIP-based PGD employing the Newton-Schulz algorithm significantly outperforms (Chen et al., 2023) in both reconstruction quality and computational complexity. We will provide more information in our simulation studies. Furthermore, we should emphasize that (Chen et al., 2023) did not offer any theoretical results regarding the performance of DIP-based MLE.

The authors in (Zhou et al., 2022) theoretically demonstrated the feasibility of accurate recovery of $\mathbf{x}_o$ even for $m < n$ measurements. While our theoretical results build upon the contributions of (Zhou et al., 2022), our paper extends significantly in two key aspects: (1) We address the multilook problem and investigate the influence of the number of looks on our bounds. To ensure sharp bounds, especially when L is large, we derive sharper bounds than those presented in (Zhou et al., 2022). These require novel technical contributions (such as using decoupling method) as detailed in our proof. (2) In contrast to the use of compression codes' codewords for the set $\mathcal{C}$ in (Zhou et al., 2022), we leverage the range of a deep image prior, inspired by recent advances in machine learning. Despite presenting new challenges in proving our results, this approach enables us to simplify and establish the relationship between Mean Squared Error (MSE) and problem specification parameters such as n, m, k, L.

Given DIP's flexibility, it has been employed for various imaging and (blind) inverse problems, e.g., compressed sensing, phase retrieval etc. (Jagatap & Hegde, 2019; Ongie et al., 2020; Darestani & Heckel, 2021; Ravula & Dimakis, 2022; Zhuang et al., 2022; 2023). To boost the performance of DIP in these applications, researchers have explored several ideas, including, introducing explicit regularization (Mataev et al., 2019), incorporating prior on network weights by introducing a learned regularization method into the DIP structure (Van Veen et al., 2018), combining with pre-trained denoisers in a Plug-and-Play fashion (Zhang et al., 2021; Sun et al., 2021), and exploring the effect of changing DIP structures and input noise settings to speed up DIP training (Li et al., 2023).

Lastly, it's important to note our work can be situated within the realm of compressed sensing (CS) (Donoho, 2006; Candès & Wakin, 2008; Davenport et al., 2012; Bora et al., 2017; Peng et al., 2020; Joshi et al., 2021; Nguyen et al., 2022), where the objective is to derive high-resolution images from lower-resolution measurements. However, notably, the specific challenge of recovery in the presence of speckle noise has not been explored in the literatures before, except in (Zhou et al., 2022) that we discussed before.

3. Main Theoretical Result

As we described in the last section, in our theoretical work, we consider the cases in which $m < n$. m can even be much smaller than n. Furthermore, for notational simplicity, in our theoretical work only, we assume that the measurements and noises are real-valued.[2] Hence, we work with the following likelihood function:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{C}} f(\mathbf{x}), \tag{4}$$

where

$$f(\mathbf{x}) = \log\det\left(\sigma_z^2 I_m + \sigma_w^2 A X^2 A^T\right) + \frac{1}{L}\sum_{\ell=1}^{L} \mathbf{y}_\ell^T \left(\sigma_z^2 I_m + \sigma_w^2 A X^2 A^T\right)^{-1} \mathbf{y}_\ell. \tag{5}$$

Note that we omit subscript L from the likelihood as a way to distinguish between the negative loglikelihood of real-valued measurements from the complex-valued ones. The following theorem is the main theoretical result of the paper. Consider the case of no additve noise, i.e. $\sigma_z = 0$, and that for all i, we have $0 < x_{\min} \le x_{o,i} \le x_{\max}$.

Theorem 3.1. *Let the elements of the measurement matrix A_{ij} be iid $\mathcal{N}(0,1)$. Suppose that $m < n$ and that the function $g_{\boldsymbol{\theta}}(\mathbf{u})$, as a function of $\boldsymbol{\theta} \in [-1,1]^k$, is Lipschitz with Lipschitz constant 1. We have*

$$\frac{1}{n}\|\hat{\mathbf{x}} - \mathbf{x}_o\|_2^2 = O\left(\frac{\sqrt{k\log n}}{m} + \frac{n\sqrt{k\log n}}{m\sqrt{Lm}}\right), \tag{6}$$

with probability $1 - O(e^{-\frac{m}{2}} + e^{-\frac{Ln}{8}} + e^{-k\log n} + e^{k\log n - \frac{n}{2}})$.

[2] For the complex-valued problem, since the phases of the elements of $\mathbf{x}_o$ are not recoverable, we can assume that $\mathbf{x}_o$ is real-valued. Even though in this case, the problem is similar to the problem we study in this paper, given that we have to deal with real and imaginary parts of the measurement matrices and noises, they are notationally more involved.

Before we discuss the proof sketch and the technical novelties of our proof strategy, let us explain some of the conclusions that can be drawn from this theorem and provide some intuition. As is clear in (6), there are two terms in the MSE. One that does not change with L and the other term that decreases with L. To understand these two terms, we provide further explanation in the following remarks.

Remark 3.2. As the number of parameters of DIP, k, increases (while keeping m, n, and L fixed), both error terms in the upper bound of MSE grow. This aligns with intuition, as increasing the number of parameters in $g_{\boldsymbol{\theta}}(\mathbf{u})$ allows the DIP model to generate more intricate images. Consequently, distinguishing between these diverse alternatives based on the measurements becomes more challenging.

Remark 3.3. The main interesting feature of the second term in the MSE, i.e. $\frac{n\sqrt{k\log n}}{m\sqrt{Lm}}$, is the fact that it grows rapidly as a function of n. In imaging systems with only additive noise, the growth is often logarithmic in n (Bickel et al., 2009), contrasting with polynomial growth observed here. This can be associated with the fact that as we increase n, the number of speckle noise elements present in our measurements increases as well. Hence, it is reasonable to expect the error term to grow faster in n compared with additive noise models. But the exact rate at which the error increases is yet unclear. As will be clarified in the proof, most of the upper bounds we derive are expected to be sharp (modulo one step in which we use a union bound, and we do not expect that union bound to loosen our upper bounds). Hence, we believe $\frac{n\sqrt{k\log n}}{m\sqrt{Lm}}$ is sharp too.

Remark 3.4. As $L \to \infty$, the second term in the upper bound of MSE converges to zero, and the dominant term becomes $\sqrt{k\log n}/m$. Note that since we are considering a fixed matrix A accross the looks, even when L goes to infinity, we should not expect to be able to recover $\mathbf{x}_o$ indepedent of the value of m. One heuristic way to see this is to calculate

$$\frac{1}{L}\sum_{\ell=1}^{L}\mathbf{y}_\ell\mathbf{y}_\ell^T = AX_o\frac{1}{L}\sum_{\ell=1}^{L}\mathbf{w}_\ell\mathbf{w}_\ell^T X_o A^T. \tag{7}$$

If we heuristically apply the weak law of large numbers and use the approximation $\frac{1}{L}\sum_\ell \mathbf{w}_\ell\mathbf{w}_\ell^T \approx I$, we see that

$$\frac{1}{L}\sum_{\ell=1}^{L}\mathbf{y}_\ell\mathbf{y}_\ell^T \approx AX_o^2A^T.$$

Under these approximations, the matirx $\frac{1}{L}\sum_\ell \mathbf{y}_\ell\mathbf{y}_\ell^T$ provides $m(m+1)/2$ (due to symmetry) linear measurements of X_o^2. Hence, inspired by classic results in compressed sensing (Candes & Davenport, 2013), intuitively, we expect the accurate recovery of $\mathbf{x}_o^2$ to be possible when $m^2 \gg k\log n$. The first error term in MSE is negligible when $m^2 \gg k\log n$, which is consistent with our conclusion based on the limit of $\frac{1}{L}\sum_\ell \mathbf{y}_\ell\mathbf{y}_\ell^T$.

We next provide a brief sketch of the proof to highlight the technical novelties of our proof and also to enable the readers to navigate through the detailed proof more easily.

Proof sketch of Theorem 3.1. Let $\hat{\mathbf{x}}_o$ denote the minimizer of function f defined as

$$f(\mathbf{x}) = f(\Sigma(\mathbf{x})) = -\log\det\Sigma + \frac{1}{L\sigma_w^2}\sum_{\ell=1}^{L}\mathrm{Tr}(\Sigma\mathbf{y}_\ell\mathbf{y}_\ell^T), \tag{8}$$

with $\Sigma = \Sigma(\mathbf{x}) = (AX^2A^T)^{-1}$, where $X = \mathrm{diag}(\mathbf{x})$. For the reasons that will become clear later, we consider a δ_n-net[3] of the set $[-1,1]^k$ and we call the mapping of the δ_n-net under g, $\mathcal{C}_n$. The choice of δ_n will be discussed later. Define $\tilde{\mathbf{x}}_o$ as the closest vector in $\mathcal{C}_n$ to $\hat{\mathbf{x}}_o$, i.e.,

$$\tilde{\mathbf{x}}_o = \mathrm{argmin}_{\mathbf{x}\in\mathcal{C}_n}\|\hat{\mathbf{x}}_o - \mathbf{x}\|.$$

Let $\tilde{X}_o = \mathrm{diag}(\tilde{\mathbf{x}}_o)$. Define Σ_o, $\hat{\Sigma}_o$ and $\tilde{\Sigma}_o$, as $\Sigma_o = \Sigma(\mathbf{x}_o)$, $\hat{\Sigma}_o = \Sigma(\hat{\mathbf{x}}_o)$, $\tilde{\Sigma}_o = \Sigma(\tilde{\mathbf{x}}_o)$, respectively. Since $\hat{\mathbf{x}}_o$ is the minimizer of (8), we have

$$f(\hat{\Sigma}_o) \le f(\Sigma_o). \tag{9}$$

On the other hand, f can be written as $f(\Sigma) = -\log\det\Sigma + \frac{1}{L\sigma_w^2}\sum_{\ell=1}^{L}\mathrm{Tr}(\Sigma AX_o\mathbf{w}_\ell\mathbf{w}_\ell^T X_o A^T)$. Let $\bar{f}(\Sigma)$ denote the expected value of $f(\Sigma)$ with respect to $\mathbf{w}_1,\dots,\mathbf{w}_\ell$. It is straightforward to show

$$\bar{f}(\Sigma) = -\log\det\Sigma + \mathrm{Tr}(\Sigma AX_o^2A^T). \tag{10}$$

As a function of Σ, $\bar{f}$ achieves its minimum at $\Sigma_o^{-1} = AX_o^2A^T$. We have

$$\begin{aligned}\bar{f}(\tilde{\Sigma}_o) - \bar{f}(\Sigma_o) &= \bar{f}(\tilde{\Sigma}_o) - f(\tilde{\Sigma}_o) + f(\tilde{\Sigma}_o) - f(\hat{\Sigma}_o)\\ &\quad + f(\hat{\Sigma}_o) - f(\Sigma_o) + f(\Sigma_o) - \bar{f}(\Sigma_o)\\ &\le \bar{f}(\tilde{\Sigma}_o) - f(\tilde{\Sigma}_o) + f(\tilde{\Sigma}_o) - f(\hat{\Sigma}_o)\\ &\quad + f(\Sigma_o) - \bar{f}(\Sigma_o),\end{aligned} \tag{11}$$

where to obtain the last inequality we have used (9). The roadmap of the rest of the proof is the following:

1. Obtaining a lower bound for $\bar{f}(\tilde{\Sigma}_o) - \bar{f}(\Sigma_o)$ in terms of $\|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2^2$. Note that since $\bar{f}(\Sigma)$ is a convex function of Σ and is minimized at Σ_o we expect to be able to obtain such bounds. Nevertheless this is the most challenging part of the proof, because of the relatively complicated dependence of $\mathbf{x}$ and Σ, and the dependence of Σ on A in addition to $\mathbf{x}$. Using sharp linear-algebraic bounds combined with the decoupling ideas (De la Pena & Giné, 2012) enabled us to obtain a sharp lower bound for this quantity.

[3]The subscript n of δ_n emphasizes that δ_n depends on n and is very close to zero when n is large.

2. Finding upper bounds for $\bar{f}(\tilde{\Sigma}_o) - f(\tilde{\Sigma}_o)$ and $f(\Sigma_o) - \bar{f}(\Sigma_o)$. Such bounds can be obtained using standard concentration of measure results, such as Hanson-Wright inequality, and the concentration of singular values of iid Gaussian random matrices.

3. Finding an upper bound for $f(\tilde{\Sigma}_o) - f(\hat{\Sigma}_o)$: note that intuitively, we expect $\|\tilde{\Sigma}_o - \hat{\Sigma}_o\|$ to be small as well. Assuming that function f is a smooth function, in the sense that it maps nearby points to nearby points in its range, we expect $f(\tilde{\Sigma}_o) - f(\hat{\Sigma}_o)$ to be small too. However, note that the function f has the randomness of $\mathbf{w}_1, \ldots, \mathbf{w}_L$ and A. Hence, to make our heuristic argument work, we have to first prove that with high probability f is a nice function.

Details of the three steps is presented in Appendix A.2.

4. Main Algorithmic Contributions

4.1. Summary of Projected Gradient Descent and DIP

As discussed in Section 1, we aim to solve the optimization problem (5) under the DIP hypothesis. A popular heuristic for achieving this goal is using projected gradient descent (PGD). At each iteration t, the estimate $\mathbf{x}^t$ is updated as follows:

$$\mathbf{x}^{t+1} = \text{Proj}(\mathbf{x}^t - \mu_t \nabla f_L(\mathbf{x}^t)), \tag{12}$$

where $\text{Proj}(\cdot)$ projects its input onto the range of the function $g_{\boldsymbol{\theta}}(\mathbf{u})$, and μ_t denotes the step size. The details of the calculation of $\nabla f_L(\mathbf{x}^t)$ are outlined in Appendix B.

An outstanding question in the implementation pertains to the nature of the projection operation $\text{Proj}(\cdot)$. If $g_\theta(\mathbf{u})$, in which θ denotes the parameters of the neural network and $\mathbf{u}$ denotes the input Gaussian noise, represents the reconstruction of the DIP, during training, DIP learns to reconstruct images by performing the following two steps:

$$\begin{aligned} \hat{\theta}^t &= \underset{\theta}{\text{argmin}} \, \|g_\theta(\mathbf{u}) - (\mathbf{x}^t - \mu_t \nabla f_L(\mathbf{x}^t))\|, \\ \mathbf{x}^{t+1} &= g_{\hat{\theta}^t}(\mathbf{u}), \end{aligned} \tag{13}$$

where to obtain a local minima in the first optimization problem, we use Adam (Kingma & Ba, 2014) . One of the main challenges in using DIPs in PGD is that the performance of DIP $g_{\boldsymbol{\theta}}(\mathbf{u})$ is affected by the structure choices, training iterations as well as the statistical properties of $\mathbf{x}^t - \mu_t \nabla f_L(\mathbf{x}^t)$ (Heckel & Soltanolkotabi, 2019). We will discuss this issue in the next section.

4.2. Challenges of DIP-based PGD

In this section, we examine two primary challenges encountered by DIP-based PGD and present novel perspectives for addressing them.

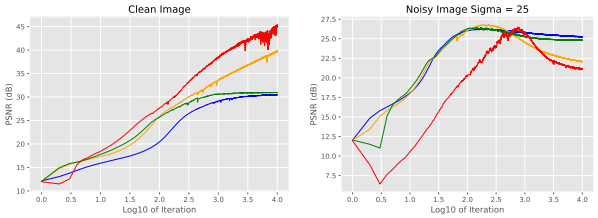

Figure 1. PSNR (averaged over 8 images) versus iteration count is depicted for four DIP models fitted to both clean (left panel) and noisy images with noise level $\sigma = 25$ (right panel). The 4-layer networks are specified as follows: Blue - kernel size=1, channels [100, 50, 25, 10]; Orange - kernel size=3, channels same as Blue; Green - kernel size=1, channels [128, 128, 128, 128]; Red - kernel size=3, channels same as Green.

4.2.1. Challege 1: Right choice of DIP

Designing PGD, as described in Section 4.1, is particularly challenging when it comes to selecting the appropriate network structure for DIP. Figure 1 clarifies the main reason. In this figure, four DIP networks are used for fitting to the clean image (left panel) and an image corrupted by the Gaussian noise (right panel). As is clear, the sophisticated networks fit the clean image very well. However, they are more susceptible to overfitting when the image is corrupted with noise. On the other hand, the networks with simpler structure do not fit to the clean image well, but are less susceptible to the noise than the sophisticated-DIPs. This issue has been observed in previous work (Heckel & Hand, 2018; Heckel & Soltanolkotabi, 2019).

The problem outlined above poses a challenge for the DIP-based PGD. Note that if $\mathbf{x}^t - \mu_t \nabla f_L(\mathbf{x}^t)$ closely approximates $\mathbf{x}_o$, fitting a highly intricate DIP to $\mathbf{x}^t - \mu_t \nabla f_L(\mathbf{x}^t)$ will yield an estimate that remains close to $\mathbf{x}_o$. Conversely, if overly simplistic networks are employed in this scenario, their final estimate may fail to closely approach $\mathbf{x}^t - \mu_t \nabla f_L(\mathbf{x}^t)$, resulting in a low-quality estimate. In the converse scenario, where $\mathbf{x}^t - \mu_t \nabla f_L(\mathbf{x}^t)$ is significantly distant from $\mathbf{x}_o$, a complex network may overfit to the noise. On the contrary, a simpler network, capable of learning only fundamental features of the image, may generate an estimate that incorporates essential image features, bringing it closer to the true image.

The above argument suggests the following approach: initiate DIP-PGD with simpler networks and progressively shift towards more complex structures as the estimate quality improves[4]. However, finding the right complexity level of the DIP for each iteration of PGD, in which the statistics of the error in the estimate $\mathbf{x}^t - \mu_t \nabla f_L(\mathbf{x}^t)$ is not known and

[4] A somewhat weaker approach would be to use intricate networks at every iteration, but then use some regularization approach such as early stopping to control the complexity of the estimates.

may be image dependent, is a challenging problem. In the next section, we propose a new approach for aliviating this problem.

4.2.2. Solution to Challege 1: Bagged-DIP

Our new approach is based on a classical idea in statistics and machine learning: Bagging. Rather than finding the right complexity level for the DIP at each iteration, which is a computationally demanding and statistically challenging problem, we use bagging. The idea of bagging is that in the case of challenging estimation problems, we create several low-bias and hopefully weakly dependent estimates (we are overloading the phrase weakly-dependent to refer to situations in which the cross-correlations of different estimates are not close to 1 or -1) of a quantity and then calculate the average of those quantities to obtain a lower-variance estimate. In order to obtain weakly dependent estimates, a common practice in the literatire is to apply the same learning scheme to multiple datasets, each of which is a random perturbation of the original training set, see e.g. the construction of random forrests.

While there are many ways to create Bagged-DIP estimates, in this paper, we explore a few very simple estimates, leaving other choices for future research. First we select a network that is sophisticated enough to fit well to real-world images. The details of the newtwork we use for this paper can be found in Appendix C. Using the neural network provides our initial estimate of the image from the noisy observation. To generate a new estimate, we begin by selecting an integer number k, partitioning an image of size $(H \times W)$ into non-overlapping patches of sizes $(h_k \times w_k)$. Independent DIPs, with the same structure as the main one, are then employed to reconstruct each of these $(h_k \times w_k)$ patches. Essentially, the estimation of the entire image involves learning $\frac{HW}{h_k w_k}$ DIP models. By placing these $\frac{HW}{h_k w_k}$ patches back into their original positions, we obtain the estimate of the entire image, denoted as $\check{\mathbf{x}}_k$. A crucial aspect of this estimate is that the estimation of a pixel relies solely on the $(h_k \times w_k)$ patch to which the pixel belongs and no other pixel. By iterating this process for K different values of $(h_k \times w_k)$, we derive K estimates denoted as $\check{\mathbf{x}}_1, \ldots, \check{\mathbf{x}}_K$. The final sought-after estimate is obtained by averaging the individual estimates.

The estimation of a pixel in $\check{\mathbf{x}}_k$ is only dependent on the $(h_k \times w_k)$ patch to which the pixel belongs. As our estimates for different values of k utilize distinct regions of the image to derive their pixel estimates, we anticipate these estimates to be weakly-dependent (again in the sense that the cross-correlations are not close to 1 or -1).

4.2.3. Challenge 2: Matrix inversion

As shown in Appendix B, the gradient of $f_L(\mathbf{x})$ defined in (2) can be written as

$$\frac{\partial f_L}{\partial x_j} = 2\mathbf{x}_j\sigma_w^2 \left(\tilde{\mathbf{a}}_{\cdot,j}^{+T} B^{-1} \tilde{\mathbf{a}}_{\cdot,j}^{+} + \tilde{\mathbf{a}}_{\cdot,j}^{-T} B^{-1} \tilde{\mathbf{a}}_{\cdot,j}^{-}\right) - \frac{2\mathbf{x}_j\sigma_w^2}{L} \sum_{\ell=1}^{L} \left[\left(\tilde{\mathbf{a}}_{\cdot,j}^{+T} B^{-1} \tilde{\mathbf{y}}_\ell\right)^2 + \left(\tilde{\mathbf{a}}_{\cdot,j}^{-T} B^{-1} \tilde{\mathbf{y}}_\ell\right)^2\right], \quad (14)$$

where $\tilde{\mathbf{a}}_{\cdot,j}^{+} = \begin{bmatrix} \Re(\mathbf{a}_{\cdot,j}) \\ \Im(\mathbf{a}_{\cdot,j}) \end{bmatrix}$, $\tilde{\mathbf{a}}_{\cdot,j}^{-} = \begin{bmatrix} -\Im(\mathbf{a}_{\cdot,j}) \\ \Re(\mathbf{a}_{\cdot,j}) \end{bmatrix}$, $\tilde{\mathbf{y}}_\ell = \begin{bmatrix} \Re(\mathbf{y}_\ell) \\ \Im(\mathbf{y}_\ell) \end{bmatrix}$, $\mathbf{a}_{\cdot,j}$ denotes the j-th column of matrix A. It's important to highlight that in each iteration of the PGD, the matrix B changes because it depends on the current estimate $\mathbf{x}^t$. This leads to the computation of the inverse of a large matrix $B \in \mathbb{R}^{2m \times 2m}$ at each iteration, posing a considerable computational challenge and a major obstacle in applying DIP-based PGD for this problem. In the next section, we present a solution to address this issue.

4.2.4. Solution to Challenge 2

To address the challenge mentioned in the last section, we propose to use Newton-Schulz algorithm. Newton-Schulz, is an iterative algorithm for obtaining a matrix inverse. The iterations of Newton-Schulz for finding $(B_t)^{-1}$ is given by

$$M^k = M^{k-1} + M^{k-1}(I - B_t M^{k-1}), \quad (15)$$

where M^k is the approximation of $(B_t)^{-1}$ at iteration k. $M^0 = (B_{t-1})^{-1}$. It is shown that if $\sigma_{\max}(I - M^0 B_t) < 1$, the Newton-Schulz converges to B_t^{-1} quadratically fast (Gower & Richtárik, 2017; Stotsky, 2020).

An observation to alleviate the mentioned issue in the previous section is that, given the nature of the gradient descent, we don't anticipate significant changes in the matrix X_t^2 from one iteration to the next. Consequently, we expect B_t and B_{t-1}, as well as their inverses, to be close to each other.

Hence, instead of calculating the full inverse at iteration $t+1$, we can employ the Newton-Schulz algorithm with M^0 set to $(B_t)^{-1}$ from the previous iteration. Our simulations will show that **one** step of the Newton-Schulz algorithm suffices.

5. Simulation Results

5.1. Study of the Impacts of Different Modules

5.1.1. Newton-Schulz iterations

In this section, we aim to answer the following questions: (1) Is the Newton-Schulz algorithm effective in our Bagged-DIP-based PGD? (2) What is the minimum number of iterations for the Newton-Schulz algorithm to have good performance in Bagged-DIP-based PGD? (3) How does the

m/n	#looks	Barbara	Peppers	House	Foreman	Boats	Parrots	Cameraman	Monarch	Average
12.5%	25	19.91/0.443	19.70/0.385	20.15/0.377	19.10/0.355	20.20/0.368	17.61/0.372	18.19/0.426	19.06/0.524	19.24/0.406
	50	20.90/0.567	21.69/0.535	22.27/0.531	20.51/0.577	21.41/0.470	19.23/0.486	19.31/0.492	21.33/0.642	20.83/0.538
	100	21.84/0.633	22.41/0.657	23.96/0.624	20.63/0.638	22.52/0.536	19.68/0.574	20.66/0.512	22.56/0.720	21.78/0.612
25%	25	23.57/0.586	23.17/0.547	24.25/0.520	23.30/0.526	22.77/0.487	21.23/0.522	21.50/0.496	23.13/0.707	22.86/0.549
	50	25.38/0.689	25.12/0.691	26.84/0.652	25.12/0.681	24.50/0.601	23.37/0.636	24.30/0.642	24.93/0.785	24.95/0.672
	100	26.26/0.748	26.14/0.759	28.33/0.717	26.41/0.772	25.72/0.682	24.55/0.720	26.22/0.719	26.28/0.845	26.24/0.745
50%	25	27.30/0.759	27.02/0.724	28.56/0.697	27.56/0.735	26.21/0.669	25.94/0.728	27.95/0.762	27.17/0.845	27.21/0.740
	50	28.67/0.816	28.52/0.804	30.30/0.762	28.88/0.827	27.58/0.739	27.23/0.799	30.21/0.843	28.86/0.898	28.78/0.818
	100	29.40/0.843	29.21/0.849	31.61/0.815	29.74/0.871	28.45/0.785	28.20/0.848	31.58/0.902	30.05/0.932	29.78/0.856

Table 1. PSNR(dB)/SSIM ↑ of 8 test images with $m/n = 12.5\%/25\%/50\%$, $L = 25/50/100$.

computation time differ when using the Newton-Schulz algorithm compared to exact inverse computation?

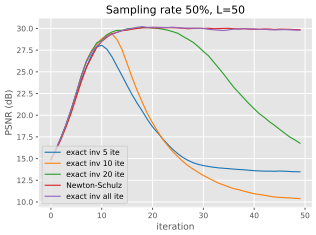

Figure 2. Newton-Schulz approximation (red) compared with computing exact inverse (purple). Blue, orange and green curves correspond to stopping the update of the inverse after the first 5, 10, and 20 iterations respectively.

Figure 2 shows one of the simulations we ran to address the first two questions. In this figure, we have chosen $L = 50$ and $m/n = 0.5$, and the learning rate of PGD is 0.01. The result of Bagged-DIP-based PGD with a **single step** of Newton-Schulz is virtually identical to PGD with the exact inverse. To investigate the impact of the Newton-Schulz algorithm further, we next checked if applying even one step of Newton-Schulz is necessary. Hence, in three different simulations we stopped the matrix inverse update at iterations 5 (blue), 10 (orange), and 20 (green). As is clear from Figure 2, a few iterations after stopping the update, PGD starts diverging. Hence, we conclude that a single step of the Newton-Schulz is necessary and sufficient for PGD.

To address the last question raised above, we evaluated how much time the calculation of the gradient takes if we use one step of the Newton-Schulz compared to the full matrix inversion. Our results are reported in Table 2. Our simulations are for sampling rate 50%, and number of looks $L = 50$ and three different images sizes.[5] As is clear the Newton-Schulz is much faster.

Table 2. Time (in seconds) required for exact matrix inversion and its Newton-Schulz approximation in PGD step.

Image size	32×32	64×64	128×128
GD w/ Newton-Schulz	$\sim$ 7e-5	$\sim$ 8e-5	$\sim$ 1e-4
GD w/o Newton-Schulz	$\sim$ 0.3	$\sim$ 1.2	$\sim$ 52.8

[5]Our algorithm still faces memory limitations on a single GPU when processing 256×256 images. Addressing this issue through approaches like parallelization remains subject for future research.

In our final algorithm, if the difference between $\|\mathbf{x}^t - \mathbf{x}^{t-1}\|_\infty > \delta_\mathbf{x}$, then we use the exact inverse update. $\delta_\mathbf{x}$ is set to 0.12 (please refer to Appendix C for details) in all our simulations. Based on this updating criterion, we observe that the exact matrix inverse is only required for the first 2-3 iterations, and it is adaptive enough to guarantee the convergence of PGD.

5.1.2. Bagged-DIP

Intuitively speaking, the more weakly dependent estimates one generate the better the average estimate will be. In the context of DIPs, there appear to be many different ways to create weakly dependent samples. The goal of this section is not to explore the full-potential of Bagged-DIPs. Instead, we aim to demonstrate that even a few weakly dependent samples can offer noticeable improvements. Hence, unlike the classical applications of bagging in which thousands of bagged samples are generated, to keep the computations managable, we have only considered three bagged estimates. Figure 3 shows one of our simulations. More simulations are in Appendix D.3. In this simulation we have chosen $K = 3$, i.e. we have only three weakly-dependent estimates. These estimates are constructed according to the recipe presented in Section 4.2.2 with the following patch sizes: $h_1 = w_1 = 32$, $h_2 = w_2 = 64$, and $h_3 = w_3 = 128$. As is clear from the left panel of Figure 3, even with these very few samples, Bagged-DIPs has offered between 0.5dB and 1dB over the three estimates it has combined.

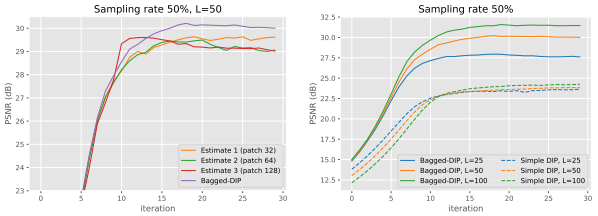

Figure 3. (Left) We compare a Bagged-DIP with three sophisticated DIP estimates. (Right) We compare PGD with simple and Bagged-DIPs across different looks on image "Cameraman".

5.1.3. Simple architectures versus Bagged-DIPs

So far our simulations have been focused on sophisticated networks. Are simpler networks that trade variance for the bias able to offer better performance? The right panel of

Figure 3 compares the performance of Bagged-DIP-based PGD with that of PGD with a simple DIP. Not only this figure shows the major improvement that is offered by using more complicated networks (in addition to bagging), but also it clarifies one of the serious limitations of the simple networks. Note that as L increases, the performance of PGD with simple DIP is not improving. In such cases, the low-accuracy of DIP blocks the algorithm from taking advantage of extra information offered by the new looks. More results about the comparison between Bagged-DIP and simple structured DIP can be found in Table 6 of Appendix D.2.

5.2. Performance of Bagged-DIP-based PGD

In this section, we offer a comprehensive simulaion study to evaluate the performance of the Bagged-DIP-based PGD on several images. We explore the following settings in our simulations:

- Number of looks (L): $L = 25, 50, 100$.
- Undersampling rate ($\frac{m}{n}$): $\frac{m}{n} = 0.125, 0.25, 0.5$.

For each combination of L and m/n, we pick one of the 8, 128×128 images mentioned in Table 1.[6] We then generate the matrix $A \in \mathbb{C}^{m\times n}$ by selecting the first m rows of a matrix that is drawn from the Haar measure on the space of orthogonal matrices. We then generate $\mathbf{w}_1, \dots, \mathbf{w}_L \sim \mathcal{CN}(0, 1)$, and for $\ell = 1, 2, \dots, L$, calculate $\mathbf{y}_\ell = AX_o\mathbf{w}_\ell$.

For our implementation of Bagged-DIP-based PGD, we have made the following choices:

- Initialization: We initialize our algorithm with $\mathbf{x}_0 = \frac{1}{L}\sum_{\ell=1}^{L} |\bar{A}^T\mathbf{y}_\ell|$. However, the final performance of DIP-based PGD does not appear to depend on the initialization. (See Figure 9 in Appendix E for further information.)
- Learning rate: We have selected a learning rate of 0.01 for the gradient desent of the likelihood, and learning rate of 0.001 in the training of DIPs.
- Number of iterations of SGD for training DIP: The details are presented in Table 3 in the appendix.
- Number of iterations of PGD: We run the outer loop (gradient descent of likelihood) for $100, 200, 300$ iterations when $m/n = 0.5, 0.25, 0.125$ respectively.

The peak signal-to-noise-ratio (PSNR) and structural index similarity (SSIM) of our reconstructions are all reported in Table 1. Qualitative results are presented in Figure 4, and Figure 10,11,12 of Appendix F.

[6]Images from the Set11 (Kulkarni et al., 2016) are chosen and cropped to 128×128 for computational manageability in Table 1.

There are no other existing algorithms that are applicable in the undersampled regime ($m < n$) considered in this paper. The only algorithm addressing speckle noise in ill-conditioned and undersampled scenarios prior to our work is the vanilla PGD proposed in (Chen et al., 2023). Although originally designed for real-valued signals and measurements, we have adapted a complex-valued version of this algorithm, with results presented in Appendix D.2. It can be seen that, for $L = 100$, $L = 50$, and $L = 25$ on average (being averaged over $m/n = 0.125, 0.25, 0.5$, and across all images) our algorithm outperforms the one presented in (Chen et al., 2023) by 1.09 dB, 1.47 dB, and 1.27 dB, respectively.

5.3. Sharpness of Our Theoretical Results

In this section, we compare our results with the theoretical findings presented in Theorem 3.1. Note that the dominant term in the MSE is the second term, i.e. $\frac{n\sqrt{k\log n}}{m\sqrt{Lm}}$. There are two features of this term that we would like to confirm with our simulations:

1. The decay of this term in relation to m is $m^{3/2}$. Hence, if we double m, we expect an additional decay factor of $2^{3/2}$ in MSE, or a gain of $15\log 2 \approx 4.5$ dB in PSNR. The average gain in PSNR we have observed is 3.99 dB, which is within the error bounds of the theoretical prediction of 4.5 dB.
2. The decay of this term in terms of L is $L^{1/2}$. Hence, if we double L,we expect an additional decay factor of $\sqrt{2}$ in the MSE, or a gain of $5\log 2 \approx 1.5$dB in PSNR. The average gain in PSNR we have observed is 1.42 dB, which is within the error bounds of the theoretical prediction of 1.5 dB.

5.4. Comparison with $A = I$ Case

The goal of this section is to provide a performance comparison with cases where we have control over the matrix A, allowing us to design it as a well-conditioned kernel. Ideally, we can assume that the measurements are in the form of $\mathbf{y}_\ell = X\mathbf{w}_\ell$, i.e. $A = I$. In this case, the task is transformed into classical despeckling problem. Since there is no wide or ill-conditioned matrix A involved in the measurment process, we expect the imaging systems to outperform for instance the 50% downsampled examples we presented in Table 1. Hence, the main question we aim to address here is:

- What is the PSNR cost incurred due to the undersampling of our measurement matrices?

To address this question, we do the following empirical study: having access to the L measurements in the form of $\mathbf{y}_\ell = X\mathbf{w}_\ell$, $\ell = 1, 2, \dots, L$, we create a sufficient statistic for estimating X. The sufficient statistic is the matrix $S =$

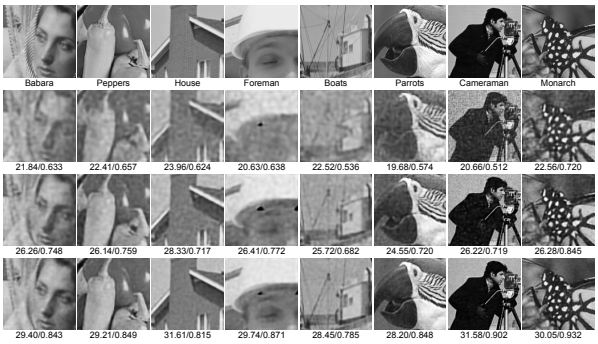

Figure 4. Raw images and reconstructed images from 100 looks of $12.5\%, 25\%, 50\%$ downsampled complex-valued measurements. Row 2-4 are $m/n = 0.125, 0.25, 0.5$ with $L = 100$ respectively. The PSNR/SSIM are reported below the reconstructed images.

$(\frac{1}{L}\sum_{\ell=1}^{L} \mathbf{y}_\ell \mathbf{y}_\ell^T)^{1/2}$. Then, this sufficient statistic is fed to DnCNN neural network (Zhang et al., 2017a;b) (which we term it as DnCNN-UB), and learn the network to achieve the best denoising performance. The results of DnCNN-UB are often between 1-3dB better than the results of our Bagged-DIP based PGD. With the exception of the cameraman, where our Bagged-DIP based PGD seems to be better. Note that the 1-3dB gain is obtained for two reasons: (1) DnCNN approach uses training data, while our Bagged-DIP-based approach does not require any training data. (2) In DnCNN approach, we have controlled the measurement matrix to be very well-conditioned. Details of DnCNN experimental settings and results can be found in Appendix D.1.

6. Conclusion

We explore the theoretical and algorithmic aspects of the problem of signal recovery from multiple sets of measurements, termed as looks, amidst the presence of speckle noise. We established an upper bound on the MSE of such imaging systems, effectively capturing the MSE's dependence on the number of measurements, image complexity, and number of looks. Drawing inspiration from our theoretical framework, we introduce the bagged deep image prior (Bagged-DIP) projected gradient descent (PGD) algorithm. Through extensive experimentation, we demonstrate that our algorithm attains state-of-the-art performance.

Acknowledgements

X.C., S.J., Z.H. and A.M. were supported in part by ONR award no. N00014-23-1-2371. S.J. was supported in part by NSF CCF-2237538. C.A.M. was supported in part by SAAB, Inc., AFOSR Young Investigator Program Award no. FA9550-22-1-0208, and ONR award no. N00014-23-1-2752. We would like to thank the reviewers for their valuable feedback.

Impact Statement

Speckle noise is a prevalent issue in various imaging systems, including synthetic aperture radar, optical coherence tomography (OCT), and ultrasound imaging, with diverse applications from medicine to earth and environmental sciences. The integration of machine learning tools with theoretical analyses, that is presented in this paper, offers a promising direction to improve the performance of such imaging systems, and in turn have a positive impact across a wide range of application areas.

References

Argenti, F., Lapini, A., Bianchi, T., and Alparone, L. A tutorial on speckle reduction in synthetic aperture radar images. *IEEE Geoscience and remote sensing magazine*, 1(3):6–35, 2013.

Bate, T., O'Keefe, D., Spencer, M. F., and Pellizzari, C. J. Experimental validation of model-based digital holographic imaging using multi-shot data. In *Unconventional Imaging and Adaptive Optics 2022*, volume 12239, pp. 83–94. SPIE, 2022.

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. Simultaneous analysis of lasso and dantzig selector. 2009.

Bora, A., Jalal, A., Price, E., and Dimakis, A. G. Compressed sensing using generative models. In *International conference on machine learning*, pp. 537–546. PMLR, 2017.

Breiman, L. Bagging predictors. *Machine learning*, 24: 123–140, 1996.

Candes, E. J. and Davenport, M. A. How well can we estimate a sparse vector? *Applied and Computational Harmonic Analysis*, 34(2):317–323, 2013.

Candès, E. J. and Wakin, M. B. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2): 21–30, 2008.

Chen, X., Hou, Z., Metzler, C., Maleki, A., and Jalali, S. Multilook compressive sensing in the presence of speckle noise. In *NeurIPS 2023 Workshop on Deep Learning and Inverse Problems*, 2023.

Darestani, M. Z. and Heckel, R. Accelerated mri with un-trained neural networks. *IEEE Transactions on Computational Imaging*, 7:724–733, 2021.

Davenport, M. A., Duarte, M. F., Eldar, Y. C., and Kutyniok, G. Introduction to compressed sensing., 2012.

De la Pena, V. and Giné, E. *Decoupling: from dependence to independence*. Springer Science & Business Media, 2012.

Donoho, D. L. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

Fan, C.-M., Liu, T.-J., and Liu, K.-H. Sunet: swin transformer unet for image denoising. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2333–2337. IEEE, 2022.

Gagnon, L. and Jouan, A. Speckle filtering of sar images: a comparative study between complex-wavelet-based and standard filters. In *Wavelet Applications in Signal and Image Processing V*, volume 3169, pp. 80–91. SPIE, 1997.

Gower, R. M. and Richtárik, P. Randomized quasi-newton updates are linearly convergent matrix inversion algorithms. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1380–1409, 2017.

Heckel, R. and Hand, P. Deep decoder: Concise image representations from untrained non-convolutional networks. In *International Conference on Learning Representations*, 2018.

Heckel, R. and Soltanolkotabi, M. Denoising and regularization via exploiting the structural bias of convolutional generators. In *International Conference on Learning Representations*, 2019.

Jagatap, G. and Hegde, C. Algorithmic guarantees for inverse imaging with untrained network priors. *Advances in neural information processing systems*, 32, 2019.

Jalali, S., Maleki, A., and Baraniuk, R. G. Minimum complexity pursuit for universal compressed sensing. *IEEE Transactions on Information Theory*, 60(4):2253–2268, 2014.

Joshi, B., Li, X., Plan, Y., and Yilmaz, O. Plugin-cs: A simple algorithm for compressive sensing with generative prior. In *NeurIPS 2021 Workshop on Deep Learning and Inverse Problems*, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kulkarni, K., Lohit, S., Turaga, P., Kerviche, R., and Ashok, A. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 449–458, 2016.

Lawson, C. L. and Hanson, R. J. *Solving least squares problems*. SIAM, 1995.

Li, T., Wang, H., Zhuang, Z., and Sun, J. Deep random projector: Accelerated deep image prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18176–18185, 2023.

Lim, J. S. and Nawab, H. Techniques for speckle noise removal. In *Applications of speckle phenomena*, volume 243, pp. 35–45. SPIE, 1980.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pp. 416–423. IEEE, 2001.

Mataev, G., Milanfar, P., and Elad, M. Deepred: Deep image prior powered by red. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

Nguyen, T. V., Jagatap, G., and Hegde, C. Provable compressed sensing with generative priors via langevin dynamics. *IEEE Transactions on Information Theory*, 68 (11):7410–7422, 2022.

Ongie, G., Jalal, A., Metzler, C. A., Baraniuk, R. G., Dimakis, A. G., and Willett, R. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

Pellizzari, C. J., Spencer, M. F., and Bouman, C. A. Phase-error estimation and image reconstruction from digital-holography data using a bayesian framework. *JOSA A*, 34(9):1659–1669, 2017.

Pellizzari, C. J., Spencer, M. F., and Bouman, C. A. Optically coherent image reconstruction in the presence of phase errors using advanced-prior models. In *Long-range imaging III*, volume 10650, pp. 68–82. SPIE, 2018.

Pellizzari, C. J., Spencer, M. F., and Bouman, C. A. Coherent plug-and-play: digital holographic imaging through atmospheric turbulence using model-based iterative reconstruction and convolutional neural networks. *IEEE Transactions on Computational Imaging*, 6:1607–1621, 2020.

Pellizzari, C. J., Bate, T. J., Donnelly, K. P., and Spencer, M. F. Solving coherent-imaging inverse problems using deep neural networks: an experimental demonstration. In *Unconventional Imaging and Adaptive Optics 2022*, volume 12239, pp. 57–65. SPIE, 2022.

Peng, P., Jalali, S., and Yuan, X. Solving inverse problems via auto-encoders. *IEEE Journal on Selected Areas in Information Theory*, 1(1):312–323, 2020.

Ravula, S. and Dimakis, A. G. One-dimensional deep image prior for time series inverse problems. In *2022 56th Asilomar Conference on Signals, Systems, and Computers*, pp. 1005–1009. IEEE, 2022.

Rudelson, M. and Vershynin, R. Non-asymptotic theory of random matrices: extreme singular values. In *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures*, pp. 1576–1602. World Scientific, 2010.

Schulz, G. Iterative berechung der reziproken matrix. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 13(1):57–59, 1933.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Stotsky, A. Convergence rate improvement of richardson and newton-schulz iterations. *arXiv preprint arXiv:2008.11480*, 2020.

Sun, Z., Latorre, F., Sanchez, T., and Cevher, V. A plug-and-play deep image prior. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8103–8107. IEEE, 2021.

Tian, C., Xu, Y., Li, Z., Zuo, W., Fei, L., and Liu, H. Attention-guided cnn for image denoising. *Neural Networks*, 124:117–129, 2020.

Tounsi, Y., Kumar, M., Nassim, A., Mendoza-Santoyo, F., and Matoba, O. Speckle denoising by variant nonlocal means methods. *Applied optics*, 58(26):7110–7120, 2019.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9446–9454, 2018.

Van Veen, D., Jalal, A., Soltanolkotabi, M., Price, E., Vishwanath, S., and Dimakis, A. G. Compressed sensing with deep image prior and learned regularization. *arXiv preprint arXiv:1806.06438*, 2018.

Xue, W., Zhang, L., Mou, X., and Bovik, A. C. Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *IEEE transactions on image processing*, 23(2):684–695, 2013.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017a.

Zhang, K., Zuo, W., Gu, S., and Zhang, L. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3929–3938, 2017b.

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

Zhou, W., Jalali, S., and Maleki, A. Compressed sensing in the presence of speckle noise. *IEEE Transactions on Information Theory*, 68(10):6964–6980, 2022.

Zhuang, Z., Yang, D., Hofmann, F., Barmherzig, D., and Sun, J. Practical phase retrieval using double deep image priors. *arXiv preprint arXiv:2211.00799*, 2022.

Zhuang, Z., Li, T., Wang, H., and Sun, J. Blind image deblurring with unknown kernel size and substantial noise. *International Journal of Computer Vision*, pp. 1–30, 2023.

A. Proofs of the main results

A.1. Preliminaries

Before stating the proofs, we present a few lemmas that will be used later in the proof.

Lemma A.1. *Let B and C denote two $n \times n$ symmetric and invertible matrices. Then, if λ_i represents the i^{th} eigenvalue of $B^{-1} - C^{-1}$, we have $|\lambda_i| \in [-\frac{\sigma_{\max}(B-C)}{\sigma_{\min}(B)\sigma_{\min}(C)}, \frac{\sigma_{\max}(B-C)}{\sigma_{\min}(B)\sigma_{\min}(C)}]$.*

Proof. Suppose λ_i is the i^{th} eigenvalue of $B^{-1} - C^{-1}$. Then, there exists a norm 1 vector $\boldsymbol{v} \in \mathbb{R}^n$ such that

$$(B^{-1} - C^{-1})\boldsymbol{v} = \lambda_i \boldsymbol{v}.$$

Multiplying both sides by B, we have

$$(I - BC^{-1})\boldsymbol{v} = \lambda_i B\boldsymbol{v}.$$

Define $\boldsymbol{u} = C^{-1}\boldsymbol{v}$. Then, we have $(C - B)\boldsymbol{u} = \lambda_i BC\boldsymbol{u}$, or equivalently

$$\lambda_i \boldsymbol{u} = (BC)^{-1}(C - B)\boldsymbol{u}.$$

Hence,

$$|\lambda_i| \leq \frac{\sigma_{\max}(C - B)}{\sigma_{\min}(B)\sigma_{\min}(C)}.$$

□

Lemma A.2. *(Rudelson & Vershynin, 2010) Let the elements of an $m \times n$ ($m < n$) matrix A be drawn independently from $\mathcal{N}(0, 1)$. Then, for any $t > 0$,*

$$\mathbb{P}(\sqrt{n} - \sqrt{m} - t \leq \sigma_{\min}(A) \leq \sigma_{\max}(A) \leq \sqrt{n} + \sqrt{m} + t) \geq 1 - 2\mathrm{e}^{-\frac{t^2}{2}}. \tag{16}$$

Lemma A.3 (Concentration of χ^2 (Jalali et al., 2014))**.** *Let $Z_1, Z_2, \ldots, Z_n$ denote a sequence of independent $\mathcal{N}(0, 1)$ random variables. Then, for any $t \in (0, 1)$, we have*

$$\mathbb{P}(\sum_{i=1}^{n} Z_i^2 \leq n(1 - t)) \leq \mathrm{e}^{\frac{n}{2}(t + \log(1-t))}.$$

Also, for any $t > 0$,

$$\mathbb{P}(\sum_{i=1}^{n} Z_i^2 \geq n(1 + t)) \leq \mathrm{e}^{-\frac{n}{2}(t - \log(1+t))}.$$

Define

Theorem A.4 (Hanson-Wright inequality)**.** *Let $\mathbf{X} = (X_1, ..., X_n)$ be a random vector with independent components with $\mathbb{E}[X_i] = 0$ and $\|X_i\|_{\Psi_2} \leq K$. Let A be an $n \times n$ matrix. Then, for $t > 0$,*

$$\mathbb{P}\Big(|\mathbf{X}^T A\mathbf{X} - \mathbb{E}[\mathbf{X}^T A\mathbf{X}]| > t\Big) \leq 2\exp\left(-c\min\left(\frac{t^2}{K^4\|A\|_{\text{HS}}^2}, \frac{t}{K^2\|A\|_2}\right)\right), \tag{17}$$

where c is a constant, and $\|X\|_{\psi_2} = \inf\{t > 0 : \mathbb{E}(\exp(X^2/t^2)) \leq 2\}$.

Theorem A.5 (Decoupling of U-processes, Theorem 3.4.1. of (De la Pena & Giné, 2012))**.** *Let $X_1, X_2, \ldots, X_n$ denote random variables with values in measurable space $(S, \mathcal{S})$. Let $(\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_n)$ denote an independent copy of $X_1, X_2, \ldots, X_n$. For $i \neq j$ let $h_{i,j} : S^2 \to \mathbb{R}$. Then, there exists a constant C such that for every $t > 0$ we have*

$$\mathbb{P}\Big(|\sum_{i \neq j} h_{i,j}(X_i, X_j)| > t\Big) \leq C\mathbb{P}\Big(C|\sum_{i \neq j} h_{i,j}(X_i, \tilde{X}_j)| > t\Big).$$

A.2. Main steps of the proof

As we discussed in Section 3, the proof has three main steps:

1. Obtaining a lower bound for $\bar{f}(\tilde{\Sigma}_o) - \bar{f}(\Sigma_o)$ in terms of $\|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2^2$: This will be presented in Section A.2.1.
2. Finding upper bounds for $\bar{f}(\tilde{\Sigma}_o) - f(\tilde{\Sigma}_o)$ and $f(\Sigma_o) - \bar{f}(\Sigma_o)$: This will be presented in Section A.2.2.
3. Finding an upper bound for $f(\tilde{\Sigma}_o) - f(\hat{\Sigma}_o)$: this will be presented in Section A.2.3.

We combine these steps and finish the proof in Section A.2.4. For notational simplicty we drop the subscript of δ_n in our proof. However, whenever we need the interpretation of the results we should not that δ depends on n and behaves like $O(1/n^5)$. It becomes more clear why we have picked this choice later. Many other choices of δ_n will work as well.

A.2.1. Obtaining lower bound for $\bar{f}(\tilde{\Sigma}_o) - \bar{f}(\Sigma_o)$

Define $\Delta\Sigma$ as

$$\Delta\Sigma = \tilde{\Sigma}_o - \Sigma_o,$$

and let λ_m denote the maximum eigenvalue of $\Sigma_o^{-\frac{1}{2}} \Delta\Sigma \Sigma_o^{-\frac{1}{2}}$. As the first step the following lemma obtains a lower bound $\bar{f}(\tilde{\Sigma}_o) - \bar{f}(\Sigma_o)$ in terms of $\tilde{\Sigma} - \Sigma_o$.

Lemma A.6. *(Zhou et al., 2022) For $\tilde{\mathbf{x}} \in \mathbb{R}^n$ and $\mathbf{x}_o \in \mathbb{R}^n$, let $\tilde{X} = \mathrm{diag}(\tilde{\mathbf{x}})$, $X_o = \mathrm{diag}(\mathbf{x}_o)$. Assume that $A\tilde{X}^2A^T$ and $AX_o^2A^T$ are both invertible, and define $\tilde{\Sigma} = (A\tilde{X}^2A^T)^{-1}$, and $\Sigma_o = (AX_o^2A^T)^{-1}$. Then,*

$$\bar{f}(\tilde{\Sigma}) - \bar{f}(\Sigma_o) \geq \frac{1}{2(1+\lambda_m)^2} \mathrm{Tr}(\Sigma_o^{-1}\Delta\Sigma\Sigma_o^{-1}\Delta\Sigma), \tag{18}$$

The next step of the proof is to connect the quantity $\mathrm{Tr}(\Sigma_o^{-1}\Delta\Sigma\Sigma_o^{-1}\Delta\Sigma)$ appearing in (18) to the difference $\tilde{\mathbf{x}} - \mathbf{x}_o$. The subsequent lemma brings us one step closer to this goal.

Lemma A.7. *(Zhou et al., 2022) Consider two $m \times m$ matrices $\tilde{\Sigma} = (A\tilde{X}^2A^T)^{-1}$ and $\Sigma = (AX^2A^T)^{-1}$ and define $\Delta\Sigma = \tilde{\Sigma} - \Sigma$. Then,*

$$\mathrm{Tr}(\Sigma^{-1}\Delta\Sigma\Sigma^{-1}\Delta\Sigma) \geq \frac{x_{\min}^4 \lambda_{\min}^2(AA^T)}{x_{\max}^8 \lambda_{\max}^4(AA^T)} \|A(\tilde{X}^2 - X^2)A^T\|_{\mathrm{HS}}^2 \tag{19}$$

$$\mathrm{Tr}(\Sigma^{-1}\Delta\Sigma\Sigma^{-1}\Delta\Sigma) \leq \frac{x_{\max}^4 \lambda_{\max}^2(AA^T)}{x_{\min}^8 \lambda_{\min}^4(AA^T)} \|A(\tilde{X}^2 - X^2)A^T\|_{\mathrm{HS}}^2. \tag{20}$$

Combining (18) and (19) enables us to obtain a lower bound for $\bar{f}(\tilde{\Sigma}_o) - \bar{f}(\Sigma_o)$ in terms of $\|A(\tilde{X}_o^2 - X_o^2)A^T\|_{\mathrm{HS}}^2$ and $\lambda_{\min}(AA^T)$ and $\lambda_{\max}(AA^T)$. Lower bounding the quantity $\frac{\lambda_{\min}^2(AA^T)}{\lambda_{\max}^4(AA^T)}$ is straightforward. Define the event:

$$\mathcal{E}_4 = \{\sqrt{n} - 2\sqrt{m} \leq \sigma_{\min}(A) \leq \sigma_{\max}(A) \leq \sqrt{n} + 2\sqrt{m}\},$$

From Lemma A.2, we have

$$\mathbb{P}(\mathcal{E}_4^c) \leq 2\exp(-\frac{m}{2}). \tag{21}$$

Hence,

$$\mathbb{P}\left(\frac{\lambda_{\min}^2(AA^T)}{\lambda_{\max}^4(AA^T)} \leq \frac{(\sqrt{n} - 2\sqrt{m})^4}{(\sqrt{n} + 2\sqrt{m})^8}\right) \leq 2\exp(-\frac{m}{2}). \tag{22}$$

The only remaining step in obtaining the lower bound we are looking for is that, the term $\|A(\tilde{X}_o^2 - X_o^2)A^T\|_{\mathrm{HS}}^2$ in the lower bound is not in the form of $\|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2^2$ yet. Hence, the final step in obtaining the lower bound is to obtain a lower bound of the form $\|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2^2$ for $\|A(\tilde{X}_o^2 - X_o^2)A^T\|_{\mathrm{HS}}^2$. Towards this goal, for $\gamma > 0$ define $\mathcal{E}_1(\gamma)$ as the event that

$$\|A(\tilde{X}_o^2 - X_o^2)A^T\|_{\mathrm{HS}}^2 \geq m(m-1)\|\mathbf{x}_o^2 - \tilde{\mathbf{x}}_o^2\|_2^2 - m^2n\gamma,$$

Our goal is to show that for an appropriate value of γ this event holds with high probability. Towards this goal we use the following lemma:

Lemma A.8. *Let the elements of $m \times n$ matrix A be drawn i.i.d. $\mathcal{N}(0,1)$. For any given $\mathbf{d} \in \mathbb{R}^{\mathbf{n}}$, define $D = \mathrm{diag}(\mathbf{d})$. Then,*

$$\mathbb{P}(\|ADA^T\|_{HS}^2 \leq m(m-1)\sum_{i=1}^{n} d_i^2 - t) \leq 2C\exp\left(-c\min\left(\frac{t^2}{C^2\|\mathbf{d}\|_\infty^4 q_{m,n}}, \frac{t}{C\|\mathbf{d}\|_\infty^2 \tilde{q}_{m,n}}\right)\right) + 2\mathrm{e}^{-\frac{n}{2}}, \tag{23}$$

where C and c are the constants that appeared in Lemmas A.5 and A.4, and

$$\begin{aligned} q_{m,n} &\triangleq m^2(2\sqrt{n}+\sqrt{m})^4, \\ \tilde{q}_{m,n} &\triangleq (2\sqrt{n}+\sqrt{m})^2. \end{aligned} \tag{24}$$

The proof of this lemma uses decoupling and is presented in Section A.3.1. Using this lemma, we have

$$\begin{aligned} \mathbb{P}(\mathcal{E}_1^c) &\overset{(a)}{\leq} 2Ce^{k\log\frac{2k}{\delta}}\exp\left(-c\min\left(\frac{\check{\alpha}_{m,n}\gamma^2}{x_{\max}^8}, \frac{\check{\beta}_{m,n}\gamma}{x_{\max}^4}\right)\right) + 2\mathrm{e}^{k\log\frac{2k}{\delta}-n/2} \\ &\leq 2Ce^{k\log\frac{2k}{\delta}}\left(\mathrm{e}^{-c\frac{\check{\alpha}_{m,n}\gamma^2}{x_{\max}^8}} + \mathrm{e}^{-c\frac{\check{\beta}_{m,n}\gamma}{x_{\max}^4}}\right) + 2\mathrm{e}^{k\log\frac{2k}{\delta}-\frac{n}{2}} \\ &= 2C\mathrm{e}^{k\log\frac{2k}{\delta}}\mathrm{e}^{-\frac{cm^2\gamma^2}{C^2x_{\max}^8(2+\sqrt{m/n})^4}} + 2C\mathrm{e}^{k\log\frac{2k}{\delta}}\mathrm{e}^{-\frac{cm^2\gamma}{Cx_{\max}^4(2+\sqrt{m/n})^2}} + 2\mathrm{e}^{k\log\frac{2k}{\delta}-n/2}, \end{aligned} \tag{25}$$

where

$$\begin{aligned} \check{\alpha}_{m,n} &\triangleq \frac{m^4n^2}{C^2m^2(2\sqrt{n}+\sqrt{m})^4} = \frac{m^2n^2}{C^2(2\sqrt{n}+\sqrt{m})^4}, \\ \check{\beta}_{m,n} &\triangleq \frac{m^2n}{C(2\sqrt{n}+\sqrt{m})^2}. \end{aligned} \tag{26}$$

In order to obtain Inequality (a), we should first note that by a simple counting argument we conclude that $|\mathcal{C}_\delta| \leq (\frac{2k}{\delta})^k$ (Shalev-Shwartz & Ben-David, 2014). Hence, by combining this result, the union bound on the choice of $\tilde{x}_o$ and Lemma A.8 we reach Inequality (a).

By setting

$$\gamma = 2C\frac{x_{\max}^4(2+\sqrt{m/n})^2}{m\sqrt{c}}\sqrt{k\log\frac{2k}{\delta}}, \tag{27}$$

we have

$$\begin{aligned} \|A(\tilde{X}_o^2 - X_o^2)A^T\|_{HS}^2 &\geq m(m-1)\sum_i^n(\tilde{x}_{o,i}^2 - x_{o,i}^2)^2 - \tilde{C}mn\sqrt{k\log\frac{2k}{\delta}} \\ &= m(m-1)\sum_i^n(\tilde{x}_{o,i} - x_{o,i})^2(\tilde{x}_{o,i} + x_{o,i})^2 - \tilde{C}mn\sqrt{k\log\frac{2k}{\delta}} \\ &\geq 4m(m-1)x_{\min}^2\sum_i^n(\tilde{x}_{o,i} - x_{o,i})^2 - \tilde{C}mn\sqrt{k\log\frac{2k}{\delta}} \\ &= 4m(m-1)x_{\min}^2\|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2^2 - \tilde{C}mn\sqrt{k\log\frac{2k}{\delta}} \end{aligned} \tag{28}$$

with probability

$$\mathbb{P}(\mathcal{E}_1^c) \quad \leq \quad O(e^{-k\log\frac{k}{\delta}} + \mathrm{e}^{k\log\frac{k}{\delta}-\frac{n}{2}}), \tag{29}$$

In the above equations $\tilde{C}$ is a constant that does not depend on m, n or δ. Furthermore, in (29) we have assumed that m is large enough (and hence γ is small enough) to make the inequality $\frac{m^2\gamma^2}{C^2x_{\max}^8(2+\sqrt{m/n})^4} < \frac{m^2\gamma}{Cx_{\max}^4(2+\sqrt{m/n})^2}$ true.

Combining (18), (19), (22), (28), and (29) we conclude the lower bound we were looking for:

$$\bar{f}(\tilde{\Sigma}_o) - \bar{f}(\Sigma_o) \geq \frac{\check{C}}{(1+\lambda_m)^2} \frac{(\sqrt{n}-2\sqrt{m})^4}{(\sqrt{n}+2\sqrt{m})^8} \left(4m(m-1)x_{\min}^2 \|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2^2 - \tilde{C}mn\sqrt{k\log\frac{2k}{\delta}} \right) \tag{30}$$

with probability

$$P(\mathcal{E}_1^c \cap \mathcal{E}_4^c) \geq 1 - O\left(e^{-\frac{m}{2}} + e^{-k\log\frac{k}{\delta}} + \mathrm{e}^{k\log\frac{k}{\delta} - \frac{n}{2}} \right).$$

Our last step for the lower bound is to simplify the expression of (30). Recall that λ_m is defined as the maximum eigenvalue of $\Sigma_o^{-\frac{1}{2}} \Delta\Sigma \Sigma_o^{-\frac{1}{2}}$. On the other hand, $\lambda_m = \|\Sigma_o^{-\frac{1}{2}} \Delta\Sigma \Sigma_o^{-\frac{1}{2}}\|_2 \leq \|\Delta\Sigma\|_2 \|\Sigma_o^{-\frac{1}{2}}\|_2^2 = \|\Delta\Sigma\|_2 \|\Sigma_o^{-1}\|_2$. But, $\|\Sigma_o^{-1}\|_2 = \|AX_o^2A^T\|_2 \leq x_{\max}^2 \lambda_{\max}(AA^T)$. Similarly, $\|\Delta\Sigma\|_2 = \|\hat{\Sigma}_o - \Sigma_o\|_2 \leq \|\hat{\Sigma}_o\|_2 + \|\Sigma_o\|_2 \leq \frac{1}{\|AX_o^2A^T\|_2} + \frac{1}{\|A\hat{X}_o^2A^T\|_2} \leq \frac{2}{x_{\min}^2\lambda_{\min}(AA^T)}$. So overall, $\lambda_m \leq \frac{2x_{\max}^2\lambda_{\max}(AA^T)}{x_{\min}^2\lambda_{\min}(AA^T)}$, and conditioned on $\mathcal{E}_4$, we have

$$\lambda_m \leq \frac{2x_{\max}^2(1+2\sqrt{m/n})^2}{x_{\min}^2(1-2\sqrt{m/n})^2}. \tag{31}$$

Hence, since $\lambda_m > 1$, we have

$$(1+\lambda_m)^2 \leq \frac{16x_{\max}^4(1+2\sqrt{m/n})^4}{x_{\min}^4(1-2\sqrt{m/n})^4} = O(1). \tag{32}$$

Hence,

$$\frac{\check{C}}{(1+\lambda_m)^2} \frac{(\sqrt{n}-2\sqrt{m})^4}{(\sqrt{n}+2\sqrt{m})^8} = \frac{\check{C}}{n^2(1+\lambda_m)^2} \frac{(1-2\sqrt{m/n})^4}{(1+2\sqrt{m/n})^8} \geq \frac{\check{\check{C}}}{n^2}. \tag{33}$$

In summary, there exist two absolute constants $\bar{C}$ and $\bar{\bar{C}}$ such that

$$\bar{f}(\tilde{\Sigma}_o) - \bar{f}(\Sigma_o) \geq \frac{\bar{C}m(m-1)}{n^2} \|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2^2 - \bar{\bar{C}} \frac{m\sqrt{k\log\frac{2}{\delta}}}{n} \tag{34}$$

with probability

$$P(\mathcal{E}_1^c \cap \mathcal{E}_4^c) \geq 1 - O\left(e^{-\frac{m}{2}} + e^{-k\log\frac{1}{\delta}} + \mathrm{e}^{k\log\frac{k}{\delta} - \frac{n}{2}} \right).$$

A.2.2. Finding upper bounds for $\bar{f}(\tilde{\Sigma}_o) - f(\tilde{\Sigma}_o)$ and $f(\Sigma_o) - \bar{f}(\Sigma_o)$

Given $t_2 > 0$ define events $\mathcal{E}_2$ and $\mathcal{E}_3$

$$\mathcal{E}_2(t_2) = \{|\delta f(\Sigma_o)| \leq t_2\}, \quad \mathcal{E}_3 = \{|\delta f(\tilde{\Sigma}_o)| \leq t_2\}.$$

The following Lemma enables to calculate the probability of $\mathcal{E}_2 \cap \mathcal{E}_3$.

Lemma A.9. *Given $\Sigma = (AX^2A^T)^{-1}$, let $\delta f(\Sigma) = f(\Sigma) - \bar{f}(\Sigma)$. Then, for $t > 0$, there exists a constant c independent of $m, n, x_{\min}$, and $x_{\max}$, such that*

$$\mathbb{P}(|\delta f(\Sigma)| \geq t|A) \leq 2\exp\left(-\frac{cLt}{2x_{\max}^2\|A^T\Sigma A\|_2} \min\left(1, \frac{t}{2mx_{\max}^2\|A^T\Sigma A\|_2}\right)\right).$$

Also, $\|A^T\Sigma A\|^2 \leq \frac{\lambda_{\max}^2(AA^T)}{\lambda_{\min}^2(AA^T)x_{\min}^4}$.

The proof of this lemma is presented in Section A.3.2.

Our goal is to use Lemma A.9 for obtaining $\mathbb{P}((\mathcal{E}_2 \cap \mathcal{E}_3)^c)$.

First note that we have

$$\mathbb{P}((\mathcal{E}_2 \cap \mathcal{E}_3)^c) = \mathbb{P}((\mathcal{E}_2 \cap \mathcal{E}_3)^c \cap \mathcal{E}_4) + \mathbb{P}((\mathcal{E}_2 \cap \mathcal{E}_3)^c \cap \mathcal{E}_4^c) \leq \mathbb{P}((\mathcal{E}_2 \cap \mathcal{E}_3)^c \cap \mathcal{E}_4) + \mathbb{P}(\mathcal{E}_4^c). \tag{35}$$

Furthermore, using Lemma A.9, for $t_2 \leq 2mx_{\max}^2 \|A^T \Sigma A\|_2$,

$$\begin{aligned}\mathbb{P}((\mathcal{E}_2 \cap \mathcal{E}_3)^c \cap \mathcal{E}_4) &\leq e^{k \log \frac{2k}{\delta}} \exp(-c_3 L t_2^2/2m) \\ &= \exp((k \log \frac{2k}{\delta}) - c_3 L t_2^2/2m).\end{aligned} \tag{36}$$

To obtain the first inequality we have used Lemma A.9 and used a union bound on all the possible choices of $\tilde{X}_o \in \mathcal{C}_n$. As discussed before, $|\mathcal{C}_\delta| \leq \mathrm{e}^{k \log \frac{2k}{\delta}}$. Let $t_2 = \sqrt{4mk \log(\frac{2k}{\delta})/Lc_3}$. Note that since we are interested in the regime that m is much bigger than $k \log \frac{2k}{\delta}$, we can conclude that $t_2 \leq 2mx_{\max}^2 \|A^T \Sigma A\|_2$. Hence,

$$\mathbb{P}((\mathcal{E}_2 \cap \mathcal{E}_3)^c \cap \mathcal{E}_4) \leq e^{-k \log \frac{2k}{\delta}}. \tag{37}$$

Hence, combining (35) and (37) we have

$$\mathbb{P}((\mathcal{E}_2 \cap \mathcal{E}_3)^c) \leq e^{-k \log \frac{2k}{\delta}} + 2e^{-\frac{m}{2}}. \tag{38}$$

In summary, we have that for $t_2 = \sqrt{4mk \log(\frac{2k}{\delta})/Lc_3}$,

$$\mathbb{P}((\mathcal{E}_2 \cap \mathcal{E}_3)^c) = O(e^{-k \log \frac{k}{\delta}} + e^{-\frac{m}{2}}). \tag{39}$$

A.2.3. Finding an upper bound for $f(\tilde{\Sigma}_o) - f(\hat{\Sigma}_o)$:

The following lemma help us obtain an upper bound for $f(\tilde{\Sigma}_o) - f(\hat{\Sigma}_o)$.

Lemma A.10. *Assume that $A\tilde{X}_o^2 A^T$ and $A\hat{X}_o^2 A^T$ are both invertible. Define $\Delta\Sigma = \tilde{\Sigma}_o - \hat{\Sigma}_o$. Then,*

$$|\lambda_i(\tilde{\Sigma}_o^{-\frac{1}{2}} \Delta\Sigma \tilde{\Sigma}_o^{-\frac{1}{2}})| \in [0, \frac{x_{\max}^2 \lambda_{\max}^2(AA^T) \|\hat{\mathbf{x}}_o^2 - \tilde{\mathbf{x}}_o^2\|_\infty}{x_{\min}^4 \lambda_{\min}^2(AA^T)}]. \tag{40}$$

Furthermore, if all the eigenvalues of $\tilde{\Sigma}_o^{-\frac{1}{2}} \Delta\Sigma \tilde{\Sigma}_o^{-\frac{1}{2}}$ fall in the range $[-0.5, 0.5]$. Then,

$$f(\tilde{\Sigma}_o) - f(\hat{\Sigma}_o) \leq \frac{x_{\max}^2 \lambda_{\max}^2(AA^T) \|\hat{\mathbf{x}}_o^2 - \tilde{\mathbf{x}}_o^2\|_\infty}{x_{\min}^4 \lambda_{\min}^2(AA^T)} \Big(2m + \frac{1}{L\sigma_w^2} \sum_{\ell=1}^{L} \mathbf{w}_\ell^T \mathbf{w}_\ell\Big). \tag{41}$$

The proof of this lemma can be found in Section A.3.3. The main objective of this section is to obtain upper bound for the following three terms:

- $\frac{1}{L\sigma_w^2} \sum_\ell \mathbf{w}_\ell^T \mathbf{w}_\ell$:
  Consider the following event:
  $$\mathcal{E}_5 = \left\{ \frac{1}{L\sigma_w^2} \sum_{\ell=1}^{L} \mathbf{w}_\ell^T \mathbf{w}_\ell \geq 2n \right\}$$
  It is straightforward to use Lemma A.3 to see that
  $$\mathbb{P}(\mathcal{E}_5) \leq \mathrm{e}^{-\frac{Ln}{8}}.$$
  Hence we conclude that
  $$\frac{1}{L\sigma_w^2} \sum_{\ell=1}^{L} \mathbf{w}_\ell^T \mathbf{w}_\ell \leq 2n. \tag{42}$$
  with probability
  $$\mathbb{P}(\mathcal{E}_5^c) \geq 1 - \mathrm{e}^{-\frac{Ln}{8}}. \tag{43}$$
  .

- $\frac{\lambda^2_{\max}(AA^T)}{\lambda^2_{\min}(AA^T)}$:

  Based on the calculation of $\mathbb{P}(\mathcal{E}_4)$ in (21), it is straightforward to see that

  $$\frac{\lambda^2_{\max}(AA^T)}{\lambda^2_{\min}(AA^T)} \leq \frac{(\sqrt{n}+2\sqrt{m})^2}{(\sqrt{n}-2\sqrt{m})^2},$$

  with probability larger than

  $$\mathbb{P}(\mathcal{E}_4) \geq 1 - 2\exp(-\frac{m}{2}). \tag{44}$$

- $\|\hat{\mathbf{x}}_o^2 - \tilde{\mathbf{x}}_o^2\|_\infty$: Before we start let us prove a simple claim. For every $\tilde{\mathbf{x}}$ in the range of $g_{\boldsymbol{\theta}}(\mathbf{u})$, there exists a vector $\mathbf{v}$ such that,

  $$\|\tilde{\mathbf{x}} - \mathbf{v}\|_2 \leq \delta.$$

  To prove this claim, let's assume that $\tilde{\mathbf{x}} = g_{\tilde{\boldsymbol{\theta}}}(\mathbf{u})$. Suppose that $\hat{\boldsymbol{\theta}}$ is a vector in the δ_n-net of $[-1,1]^k$that is closest to $\tilde{\boldsymbol{\theta}}$. By the definition of δ_n-net, we have

  $$\|\tilde{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}\|_2 \leq \delta_n.$$

  Hence, it is straightforward to use Lipschitzness of $g_{\tilde{\boldsymbol{\theta}}}(\mathbf{u})$ to prove that

  $$\|\tilde{\mathbf{x}} - \hat{\mathbf{x}}\|_2 \leq \delta, \tag{45}$$

  and

  $$\|\hat{\mathbf{x}}_o^2 - \tilde{\mathbf{x}}_o^2\|_\infty \leq 2x_{\max}\|\hat{\mathbf{x}}_o - \tilde{\mathbf{x}}_o\|_\infty \leq 2x_{\max}\|\hat{\mathbf{x}}_o - \tilde{\mathbf{x}}_o\|_2 \leq 2x_{\max}\delta,$$

Summarizing the discusions of this section, we can conclude that there exist a constant $\tilde{C}$ such that

$$|f(\tilde{\Sigma}_o) - f(\hat{\Sigma}_o)| \leq \tilde{C}n\delta \tag{46}$$

with probability larger than

$$\mathbb{P}(\mathcal{E}_4 \cap \mathcal{E}_5) \geq 1 - O(e^{-\frac{m}{2}} + e^{-\frac{Ln}{8}}).$$

A.2.4. Summary of the bounds

As mentioned in (11) we have

$$\bar{f}(\tilde{\Sigma}_o) - \bar{f}(\Sigma_o) \leq \bar{f}(\tilde{\Sigma}_o) - f(\tilde{\Sigma}_o) + f(\tilde{\Sigma}_o) - f(\hat{\Sigma}_o) + f(\Sigma_o) - \bar{f}(\Sigma_o), \tag{47}$$

Furthermore, we proved the following:

- According to (34) there exist two constants $\bar{C}$ and $\bar{\bar{C}}$ such that

  $$\bar{f}(\tilde{\Sigma}_o) - \bar{f}(\Sigma_o) \geq \frac{\bar{C}m(m-1)}{n^2}\|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2^2 - \bar{\bar{C}}\frac{m\sqrt{k\log\frac{2k}{\delta}}}{n} \tag{48}$$

  with probability

  $$P(\mathcal{E}_1^c \cap \mathcal{E}_4^c) \geq 1 - O\left(e^{-\frac{m}{2}} + e^{-k\log\frac{k}{\delta}} + \mathrm{e}^{k\log\frac{k}{\delta}-\frac{n}{2}}\right).$$

- According to the discussion of Section A.2.2, we have

  $$|\bar{f}(\tilde{\Sigma}_o) - f(\tilde{\Sigma}_o)| \leq \sqrt{4mk\log(\frac{2k}{\delta})/Lc_3},$$
  $$|\bar{f}(\Sigma_o) - f(\Sigma_o)| \leq \sqrt{4mk\log(\frac{2k}{\delta})/Lc_3}, \tag{49}$$

  with probability $1 - O(e^{-k\log\frac{k}{\delta}} + e^{-\frac{m}{2}})$.

- According to the discussion of Section A.2.3, there exists a constant $\tilde{C}$ such that

$$|f(\tilde{\Sigma}_o) - f(\hat{\Sigma}_o)| \leq \tilde{C}n\delta \tag{50}$$

with probability larger than

$$\mathbb{P}(\mathcal{E}_4 \cap \mathcal{E}_5) \geq 1 - O(e^{-\frac{m}{2}} + e^{-\frac{Ln}{8}}).$$

Combining all these three results with (47) we notice that

$$\frac{\bar{C}m(m-1)}{n^2}\|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2^2 \leq \bar{\bar{C}}\frac{m\sqrt{k\log\frac{2}{\delta}}}{n} + 2\sqrt{4mk\log(\frac{2}{\delta})/Lc_3} + \tilde{C}n\delta, \tag{51}$$

with probability $1 - O(e^{-\frac{m}{2}} + e^{-\frac{Ln}{8}} + e^{-k\log\frac{k}{\delta}} + \mathrm{e}^{k\log\frac{k}{\delta}-\frac{n}{2}})$. Hence, we can conclude that

$$\frac{1}{n}\|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2^2 \leq O\left(\frac{\sqrt{k\log\frac{k}{\delta}}}{m} + \frac{n\sqrt{k\log\frac{k}{\delta}}}{m\sqrt{Lm}} + \frac{n^2\delta}{m^2}\right), \tag{52}$$

with probability $1 - O(e^{-\frac{m}{2}} + e^{-\frac{Ln}{8}} + e^{-k\log\frac{1}{\delta}} + \mathrm{e}^{k\log\frac{1}{\delta}-\frac{n}{2}})$.

Set $\delta = \frac{1}{n^5}$, we have

$$\frac{1}{n}\|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2^2 = O\left(\frac{\sqrt{k\log n}}{m} + \frac{n\sqrt{k\log n}}{m\sqrt{Lm}}\right), \tag{53}$$

with probability $1 - O(e^{-\frac{m}{2}} + e^{-\frac{Ln}{8}} + e^{-k\log n} + \mathrm{e}^{k\log n-\frac{n}{2}})$.

So far, we have found an upper bound for the error between $\|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2$. However, our final estimate is $\hat{\mathbf{x}}_o$. Note that we have

$$\|\hat{\mathbf{x}}_o - \mathbf{x}_o\|_2 \leq \|\hat{\mathbf{x}}_o - \tilde{\mathbf{x}}_o\|_2 + \|\tilde{\mathbf{x}}_o - \mathbf{x}_o\|_2 \leq \delta + \|\tilde{x}_o - x_o\|_2. \tag{54}$$

Hence, we have

$$\frac{1}{n}\|\hat{\mathbf{x}}_o - \mathbf{x}_o\|_2^2 = O\left(\frac{\sqrt{k\log n}}{m} + \frac{n\sqrt{k\log n}}{m\sqrt{Lm}}\right), \tag{55}$$

with probability $1 - O(e^{-\frac{m}{2}} + e^{-\frac{Ln}{8}} + e^{-k\log n} + \mathrm{e}^{k\log n-\frac{n}{2}})$.

A.3. Proof of auxiliary lemmas

A.3.1. Proof of Lemma A.8

Let $\boldsymbol{a}_i^T$ denote the i^{th} row of matrix A. We have

$$\|ADA^T\|_{HS}^2 = \sum_{i=1}^m\sum_{j=1}^m |\boldsymbol{a}_i^T D\boldsymbol{a}_j|^2 \geq \sum_{i=1}^m\sum_{j\neq i} |\boldsymbol{a}_i^T D\boldsymbol{a}_j|^2. \tag{56}$$

Note that

$$\mathbb{E}(\sum_{i=1}^m\sum_{j\neq i} |\boldsymbol{a}_i^T D\boldsymbol{a}_j|^2) = m(m-1)\sum_{i=1}^n d_i^2.$$

Using Theorem A.5 we conclude that there exists a constant C such that

$$\begin{aligned}
&\mathbb{P}(|\sum_{i=1}^m\sum_{j\neq i} |\boldsymbol{a}_i^T D\boldsymbol{a}_j|^2 - m(m-1)\sum_{i=1}^n d_i^2| > t)\\
&\leq C\mathbb{P}(C|\sum_{i=1}^m\sum_{j\neq i} |\boldsymbol{a}_i^T D\tilde{\boldsymbol{a}}_j|^2 - m(m-1)\sum_{i=1}^n d_i^2| > t)\\
&= C\mathbb{P}(C|\sum_{i=1}^m \boldsymbol{a}_i^T D\sum_{j\neq i}\tilde{\boldsymbol{a}}_j\tilde{\boldsymbol{a}}_j^T D\boldsymbol{a}_i - m(m-1)\sum_{i=1}^n d_i^2| > t),
\end{aligned} \tag{57}$$

where $\tilde{\boldsymbol{a}}_1, \tilde{\boldsymbol{a}}_2, \ldots, \tilde{\boldsymbol{a}}_m$ denote independent copies of $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m$. Define $\tilde{A}$ as the matrix whose rows are $\tilde{\boldsymbol{a}}_1^T, \tilde{\boldsymbol{a}}_2^T, \ldots, \tilde{\boldsymbol{a}}_m^T$. Also, let $\tilde{A}_{\setminus i}$ denote the matrix that is constructed by removing the i^{th} row of $\tilde{A}$. Define

$$F \triangleq \begin{bmatrix} D\tilde{A}_{\setminus 1}^T \tilde{A}_{\setminus 1} D & 0 & \ldots & 0 \\ 0 & D\tilde{A}_{\setminus 2}^T \tilde{A}_{\setminus 2} D & \ldots & 0 \\ 0 & 0 & \ldots & D\tilde{A}_{\setminus m}^T \tilde{A}_{\setminus m} D \end{bmatrix}.$$

and

$$\boldsymbol{v}^T = [\boldsymbol{a}_1^T, \boldsymbol{a}_2^T, \ldots, \boldsymbol{a}_m^T].$$

Using Theorem A.4 we have

$$\begin{aligned} &\mathbb{P}(C|\sum_{i=1}^m \boldsymbol{a}_i^T D \sum_{j\neq i} \tilde{\boldsymbol{a}}_j \tilde{\boldsymbol{a}}_j^T D \boldsymbol{a}_i - m(m-1)\sum_{i=1}^n d_i^2| > t \mid \tilde{A}) \\ &= \mathbb{P}(C|\boldsymbol{v}^T F \boldsymbol{v} - \mathbb{E}\boldsymbol{v}^T F \boldsymbol{v}| > t \mid \tilde{A}) \\ &\leq 2\exp\left(-c\min(\frac{t^2}{C^2\|F\|_{HS}^2}, \frac{t}{C\|F\|_2})\right) \end{aligned} \tag{58}$$

Hence, in order to obtain a more explicit upper bound, we have to find upper bounds for $\|F\|_2$ and $\|F\|_{HS}^2$. First note that

$$\begin{aligned} \lambda_{\max}(F) &= \max_i(\lambda_{\max}(D\tilde{A}_{\setminus i}^T \tilde{A}_{\setminus i} D)) \\ &\leq \lambda_{\max}(D\tilde{A}^T \tilde{A} D) \leq \|\mathbf{d}\|_\infty^2 \lambda_{\max}(\tilde{A}^T \tilde{A}). \end{aligned} \tag{59}$$

Similarly,

$$\begin{aligned} \|F\|_{HS}^2 &= \sum_{i=1}^m \|D\tilde{A}_{\setminus i}^T \tilde{A}_{\setminus i} D\|_{HS}^2 \\ &\overset{(a)}{\leq} \sum_{i=1}^m m\lambda_{\max}^2(D\tilde{A}_{\setminus i}^T \tilde{A}_{\setminus i} D) \\ &\overset{(b)}{\leq} m^2 \|\mathbf{d}\|_\infty^4 \lambda_{\max}^2(\tilde{A}^T \tilde{A}), \end{aligned} \tag{60}$$

where Inequality (a) uses the fact that the rank of matrix $D\tilde{A}_{\setminus i}^T \tilde{A}_{\setminus i} D$ is $m-1$, and Inequality (b) uses (59). Finally, using Lemma A.2 we have

$$\mathbb{P}(\sigma_{\max}(\tilde{A}) > 2\sqrt{n} + \sqrt{m}) \leq 2\mathrm{e}^{-\frac{n}{2}}, \tag{61}$$

and hence

$$\mathbb{P}(\lambda_{\max}(\tilde{A}^T A) > (2\sqrt{n} + \sqrt{m})^2) \leq 2\mathrm{e}^{-\frac{n}{2}}. \tag{62}$$

By combining (57) and (58) we obtain

$$\begin{aligned} &\mathbb{P}(|\sum_{i=1}^m \sum_{j\neq i} |\boldsymbol{a}_i^T D \boldsymbol{a}_j|^2 - m(m-1)\sum_{i=1}^n d_i^2| > t \mid \tilde{A}) \\ &\leq 2C\mathbb{E}\left(\exp\left(-c\min\left(\frac{t^2}{C^2\|F\|_{HS}^2}, \frac{t}{C\|F\|_2}\right)\right)\right), \end{aligned} \tag{63}$$

where the expected value is with respect to the randomness in F or equivalently $\tilde{A}$.

Let the event $\mathcal{E}$ denote the event of $\sigma_{\max}(\tilde{A}) \leq 2\sqrt{n} + \sqrt{m}$, and $\mathbb{I}_{\mathcal{E}}$ denote the indicator function of the event $\mathcal{E}$. Then,

using (63) we have

$$\begin{aligned}
&\mathbb{P}(|\sum_{i=1}^{m}\sum_{j\neq i}|\boldsymbol{a}_i^T D\boldsymbol{a}_j|^2 - m(m-1)\sum_{i=1}^{n} d_i^2| > t)\\
&= \mathbb{P}(\{|\sum_{i=1}^{m}\sum_{j\neq i}|\boldsymbol{a}_i^T D\boldsymbol{a}_j|^2 - m(m-1)\sum_{i=1}^{n} d_i^2| > t\} \cap \mathcal{E})\\
&+ \mathbb{P}(\{|\sum_{i=1}^{m}\sum_{j\neq i}|\boldsymbol{a}_i^T D\boldsymbol{a}_j|^2 - m(m-1)\sum_{i=1}^{n} d_i^2| > t\} \cap \mathcal{E}^c)\\
&\leq \mathbb{E}\left(\mathbb{P}(|\sum_{i=1}^{m}\sum_{j\neq i}|\boldsymbol{a}_i^T D\boldsymbol{a}_j|^2 - m(m-1)\sum_{i=1}^{n} d_i^2| > t \mid \tilde{A})\mathbb{I}_{\mathcal{E}}\right) + \mathbb{P}(\mathcal{E}^c)\\
&\leq 2C\mathbb{E}\left(\exp\left(-c\min\left(\frac{t^2}{C^2\|F\|_{HS}^2}, \frac{t}{C\|F\|_2}\right)\right)\mathbb{I}_{\mathcal{E}}\right) + \mathbb{P}(\mathcal{E}^c)\\
&\leq 2C\exp\left(-c\min\left(\frac{t^2}{C^2\|\mathbf{d}\|_\infty^4 q_{m,n}}, \frac{t}{C\|\mathbf{d}\|_\infty^2 \tilde{q}_{m,n}}\right)\right) + 2\mathrm{e}^{-\frac{n}{2}}.
\end{aligned} \tag{64}$$

A.3.2. Proof of Lemma A.9

By definition,

$$\begin{aligned}
\delta f(\Sigma) &= f(\Sigma) - \bar{f}(\Sigma)\\
&= \frac{1}{L\sigma_w^2}\sum_{\ell=1}^{L}\mathbf{w}_\ell^T X_o A^T \Sigma A X_o \mathbf{w}_\ell - \mathrm{Tr}(\Sigma A X_o^2 A^T).
\end{aligned} \tag{65}$$

Define matrix $B \in \mathbb{R}^{Ln\times Ln}$ as $B = X_o A^T \Sigma A X_o$ and $\tilde{B}$ as

$$\tilde{B} = \begin{bmatrix} B & 0 & \cdots & 0\\ 0 & B & \cdots & 0\\ 0 & 0 & \cdots & 0\\ 0 & 0 & \cdots & B \end{bmatrix}.$$

Furthermore, define

$$\tilde{\mathbf{w}}^\top = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L].$$

Then, by the Hanson-Wright inequality (Theorem A.4), we have

$$\begin{aligned}
&\mathbb{P}(|\frac{1}{L\sigma_w^2}\tilde{\mathbf{w}}^T\tilde{B}\tilde{\mathbf{w}} - \mathrm{Tr}(\Sigma A X_o^2 A^T)| > t)\\
&\leq 2\exp\Big(-c\min(\frac{L^2t^2}{4\|\tilde{B}\|_{\mathrm{HS}}^2}, \frac{Lt}{2\|\tilde{B}\|_2})\Big).
\end{aligned} \tag{66}$$

First note that,

$$\|\tilde{B}\|_2 = \|B\|_2. \tag{67}$$

Furthermore,

$$\begin{aligned}
\|\tilde{B}\|_{\mathrm{HS}}^2 &= L\mathrm{Tr}(B^2) = L\sum_{i=1}^{m}\lambda_i^2(B)\\
&\leq Lm\lambda_{\max}^2(B) = Lm\|B\|_2^2,
\end{aligned} \tag{68}$$

where the second equality is due to the fact that $\text{rank}(B) = m$. On the other hand, $\|B\|_2 = \|X_o A^T \Sigma A X_o\|_2 \leq x_{\max}^2 \|A^T \Sigma A\|_2$. Moreover,

$$\begin{aligned}\|A^T\Sigma A\|_2^2 &= \max_{\mathbf{u}\in\mathbb{R}^n} \frac{\mathbf{u}^T A^T \Sigma A A^T \Sigma A \mathbf{u}}{\|\mathbf{u}\|_2^2} \\ &\leq \lambda_{\max}(A^TA)\lambda_{\max}(AA^T)\lambda_{\max}^2(\Sigma) \end{aligned} \tag{69}$$

But $\Sigma = (AX^2A^T)^{-1}$ and $X = \text{diag}(\mathbf{x})$. Therefore, $\lambda_{\max}(\Sigma) = (\lambda_{\min}(AX^2A^T))^{-1} \leq (\lambda_{\min}(AA^T)x_{\min}^2)^{-1}$ and

$$\|A^T\Sigma A\|_2^2 \leq \frac{\lambda_{\max}(AA^T)\lambda_{\max}(A^TA)}{\lambda_{\min}^2(AA^T)x_{\min}^4}.$$

A.3.3. Proof of Lemma A.10

To prove $|\lambda_i(\tilde{\Sigma}_o^{-\frac{1}{2}}\Delta\Sigma\tilde{\Sigma}_o^{-\frac{1}{2}})| \in [0, \frac{x_{\max}^2\lambda_{\max}^2(AA^T)\|\hat{\mathbf{x}}_o^2 - \tilde{\mathbf{x}}_o^2\|_\infty}{x_{\min}^4\lambda_{\min}^2(AA^T)}]$, first note that

$$\begin{aligned}|\lambda_i| &\leq \frac{\sigma_{\max}(\Delta\Sigma)}{\sigma_{\min}(\tilde{\Sigma}_o)} \\ &= \frac{\sigma_{\max}((A\hat{X}_o^2A^T)^{-1} - (A\tilde{X}_o^2A^T)^{-1})}{\sigma_{\min}(\tilde{\Sigma}_o)} \\ &\overset{(a)}{\leq} \frac{\sigma_{\max}((A\hat{X}_o^2A^T) - (A\tilde{X}_o^2A^T))}{\sigma_{\min}(\tilde{\Sigma}_o)\sigma_{\min}(A\hat{X}_o^2A^T)\sigma_{\min}(A\tilde{X}_o^2A^T)} \\ &= \frac{\sigma_{\max}(A\tilde{X}_o^2A^T)\sigma_{\max}((A\hat{X}_o^2A^T) - (A\tilde{X}_o^2A^T))}{\sigma_{\min}(A\hat{X}_o^2A^T)\sigma_{\min}(A\tilde{X}_o^2A^T)}. \\ &\leq \frac{x_{\max}^2\lambda_{\max}^2(AA^T)\|\hat{\mathbf{x}}_o^2 - \tilde{\mathbf{x}}_o^2\|_\infty}{x_{\min}^4\lambda_{\min}^2(AA^T)}. \end{aligned} \tag{70}$$

To obtain inequality (a) we have used Lemma A.1.

To prove (41) we start with

$$f(\hat{\Sigma}_o) - f(\tilde{\Sigma}_o) \leq |\log\det(\hat{\Sigma}_o) - \log\det(\tilde{\Sigma}_o)| + \frac{1}{L\sigma_w^2}|\sum_{\ell=1}^{L}\mathbf{y}_\ell^T((A\hat{X}_o^2A^T)^{-1} - (A\tilde{X}_o^2A^T)^{-1})\mathbf{y}_\ell|. \tag{71}$$

We have

$$\begin{aligned}&|\log\det(\hat{\Sigma}_o) - \log\det(\tilde{\Sigma}_o)| \\ &= |\log\det(I + \tilde{\Sigma}_o^{-\frac{1}{2}}\Delta\Sigma\tilde{\Sigma}_o^{-\frac{1}{2}})| \\ &\leq \sum_{i=1}^{n}|\log(1 + \lambda_i(\tilde{\Sigma}_o^{-\frac{1}{2}}\Delta\Sigma\tilde{\Sigma}_o^{-\frac{1}{2}}))| \\ &\overset{(b)}{\leq} 2\sum_{i=1}^{m}|\lambda_i(\tilde{\Sigma}_o^{-\frac{1}{2}}\Delta\Sigma\tilde{\Sigma}_o^{-\frac{1}{2}})| \\ &\leq 2m\frac{x_{\max}^2\lambda_{\max}^2(AA^T)\|\hat{\mathbf{x}}_o^2 - \tilde{\mathbf{x}}_o^2\|_\infty}{x_{\min}^4\lambda_{\min}^2(AA^T)}, \end{aligned} \tag{72}$$

where Inequality (b) uses the assumption that $\lambda_i(\tilde{\Sigma}_o^{-\frac{1}{2}}\Delta\Sigma\tilde{\Sigma}_o^{-\frac{1}{2}}) \in [-0.5, 0.5]$ and $|\log(1+x)| <= 2|x|$ when $x \in [0.5, +\infty)$.

Furthermore, note that

$$\begin{aligned}
&\frac{1}{L\sigma_w^2}|\sum_{\ell=1}^{L}\mathbf{y}_\ell^T((A\hat{X}_o^2A^T)^{-1}-(A\tilde{X}_o^2A^T)^{-1})\mathbf{y}_\ell| \\
&\leq \sigma_{\max}((A\hat{X}_o^2A^T)^{-1}-(A\tilde{X}_o^2A^T)^{-1})\frac{1}{L\sigma_w^2}\sum_{\ell=1}^{L}\mathbf{y}_\ell^T\mathbf{y}_\ell \\
&\leq \frac{\lambda_{\max}(AA^T)\|\hat{\mathbf{x}}_o^2-\tilde{\mathbf{x}}_o^2\|_\infty}{x_{\min}^4\lambda_{\min}^2(AA^T)}\frac{1}{L\sigma_w^2}\sum_{\ell=1}^{L}\mathbf{y}_\ell^T\mathbf{y}_\ell \\
&\leq \frac{\lambda_{\max}(AA^T)\|\hat{\mathbf{x}}_o^2-\tilde{\mathbf{x}}_o^2\|_\infty}{L\sigma_w^2x_{\min}^4\lambda_{\min}^2(AA^T)}\sum_{\ell=1}^{L}\mathbf{y}_\ell^T\mathbf{y}_\ell \\
&\leq \frac{x_{\max}^2\lambda_{\max}^2(AA^T)\|\hat{\mathbf{x}}_o^2-\tilde{\mathbf{x}}_o^2\|_\infty}{L\sigma_w^2x_{\min}^4\lambda_{\min}^2(AA^T)}\sum_{\ell=1}^{L}\mathbf{w}_\ell^T\mathbf{w}_\ell.
\end{aligned}$$

B. Likelihood function and its gradient

B.1. Caculation of the likelihood function

The aim of this section is to derive the loglikelihood for our model,

$$\mathbf{y}_\ell = AX\mathbf{w}_\ell + \mathbf{z}_\ell, \qquad \text{for} \quad \ell = 1,\dots,L,$$

where $\mathbf{w}_1,\mathbf{w}_2,\dots,\mathbf{w}_L$, and $\mathbf{z}_1,\mathbf{z}_2,\dots,\mathbf{z}_L$ are independent and identically distributed $\mathcal{CN}(0,\sigma_w^2I_n)$ and $\mathcal{CN}(0,\sigma_z^2I_p)$ respectively. Since the noises are indepednet across the looks, we can write the loglikelihood for one of the looks, and then add the loglikelihoods to obtain the likelihood for all the looks. For notational simplicity, we write the measurements of one of the looks as:

$$\mathbf{y} = AX\mathbf{w} + \mathbf{z}$$

Note that $\mathbf{y}$ is a linear combination of two Gaussian random vectors and is hence Gaussian. Hence, by writing the real and imaginary parts of $\mathbf{y}$ seperately we will have

$$\Re(\mathbf{y}) + \Im(\mathbf{y}) = (\Re(AX) + i\Im(AX))(\mathbf{w}^{(1)} + i\mathbf{w}^{(2)}) + (\mathbf{z}^{(1)} + i\mathbf{z}^{(2)}),$$

and

$$\tilde{\mathbf{y}} \triangleq \begin{bmatrix}\Re(\mathbf{y})\\ \Im(\mathbf{y})\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, B\right),$$

where

$$B = \begin{bmatrix}\sigma_z^2I_n + \sigma_w^2\Re(AX^2\bar{A}^T) & -\sigma_w^2\Im(AX^2\bar{A}^T)\\ \sigma_w^2\Im(AX^2\bar{A}^T) & \sigma_z^2I_n + \sigma_w^2\Re(AX^2\bar{A}^T)\end{bmatrix}.$$

Hence, the log-likelihood of our data $\mathbf{y}$ as a function of $\mathbf{x}$ is

$$\ell(\mathbf{x}) = -\frac{1}{2}\log\det(B) - \frac{1}{2}\begin{bmatrix}\Re(\mathbf{y}^T) & \Im(\mathbf{y}^T)\end{bmatrix}(B)^{-1}\begin{bmatrix}\Re(\mathbf{y})\\ \Im(\mathbf{y})\end{bmatrix} + C. \tag{73}$$

Note that equation (73) is for a single look. Hence the loglikelihood of $\mathbf{y}_1,\mathbf{y}_2,\dots,\mathbf{y}_L$ as a function of $\mathbf{x}$ is:

$$\ell(\mathbf{x}) = -\frac{L}{2}\log\det(B) - \frac{1}{2}\sum_{\ell=1}^{L}\tilde{\mathbf{y}}_\ell^TB^{-1}\tilde{\mathbf{y}}_\ell + C, \tag{74}$$

Since we would like to maximize $\ell(\mathbf{x})$ as a function of $\mathbf{x}$, for notational simplicty we define the cost function $f_L(\mathbf{x}):\mathbb{R}^n\to\mathbb{R}$:

$$f_L(\mathbf{x}) = \log\det(B) + \frac{1}{L}\sum_{\ell=1}^{L}\tilde{\mathbf{y}}_\ell^TB^{-1}\tilde{\mathbf{y}}_\ell, \tag{75}$$

that we will minimize to obtain the maximum likelihood estimate.

B.2. Calculation of the gradient of the likelihood function

As discussed in the main text, to execute the projected gradient descent, it is necessary to compute the gradient of the negative log-likelihood function ∂f_L. The derivatives of f_L with respect to each element $\mathbf{x}_j$ of $\mathbf{x}$ is given by:

$$\begin{aligned}\frac{\partial f_L}{\partial \mathbf{x}_j} =& 2\mathbf{x}_j\sigma_w^2\left(\begin{bmatrix}\Re(\mathbf{a}_{\cdot,j}^T) & \Im(\mathbf{a}_{\cdot,j}^T)\end{bmatrix} B^{-1}\begin{bmatrix}\Re(\mathbf{a}_{\cdot,j})\\ \Im(\mathbf{a}_{\cdot,j})\end{bmatrix} + \begin{bmatrix}-\Im(\mathbf{a}_{\cdot,j}^T) & \Re(\mathbf{a}_{\cdot,j}^T)\end{bmatrix} B^{-1}\begin{bmatrix}-\Im(\mathbf{a}_{\cdot,j})\\ \Re(\mathbf{a}_{\cdot,j})\end{bmatrix}\right)\\ &- \frac{2\mathbf{x}_j\sigma_w^2}{L}\sum_{\ell=1}^{L}\left[\left(\begin{bmatrix}\Re(\mathbf{a}_{\cdot,j}^T) & \Im(\mathbf{a}_{\cdot,j}^T)\end{bmatrix} B^{-1}\begin{bmatrix}\Re(\mathbf{y}_\ell)\\ \Im(\mathbf{y}_\ell)\end{bmatrix}\right)^2 + \left(\begin{bmatrix}-\Im(\mathbf{a}_{\cdot,j}^T) & \Re(\mathbf{a}_{\cdot,j}^T)\end{bmatrix} B^{-1}\begin{bmatrix}\Re(\mathbf{y}_\ell)\\ \Im(\mathbf{y}_\ell)\end{bmatrix}\right)^2\right]\\ =& 2\mathbf{x}_j\sigma_w^2\left(\tilde{\mathbf{a}}_{\cdot,j}^{+T}B^{-1}\tilde{\mathbf{a}}_{\cdot,j}^{+} + \tilde{\mathbf{a}}_{\cdot,j}^{-T}B^{-1}\tilde{\mathbf{a}}_{\cdot,j}^{-}\right) - \frac{2\mathbf{x}_j\sigma_w^2}{L}\sum_{\ell=1}^{L}\left[\left(\tilde{\mathbf{a}}_{\cdot,j}^{+T}B^{-1}\tilde{\mathbf{y}}_\ell\right)^2 + \left(\tilde{\mathbf{a}}_{\cdot,j}^{-T}B^{-1}\tilde{\mathbf{y}}_\ell\right)^2\right],\end{aligned} \tag{76}$$

where $\mathbf{a}_{\cdot,j}$ denotes the j-th column of matrix A, $\tilde{\mathbf{a}}_{\cdot,j}^{+} = \begin{bmatrix}\Re(\mathbf{a}_{\cdot,j})\\ \Im(\mathbf{a}_{\cdot,j})\end{bmatrix}$ and $\tilde{\mathbf{a}}_{\cdot,j}^{-} = \begin{bmatrix}-\Im(\mathbf{a}_{\cdot,j})\\ \Re(\mathbf{a}_{\cdot,j})\end{bmatrix}$.

B.3. More simplification of the gradient

The special form of the matrix B enables us to do the calculations more efficiently. To see this point, define:

$$U + iV \triangleq \left(\sigma_z^2 I_n + \sigma_w^2 A X^2 \bar{A}^T\right)^{-1},$$

where $U, V \in R^{m\times m}$. These two matrices should satisfy:

$$\begin{aligned}\left(\sigma_z^2 I_n + \sigma_w^2\Re(AX^2\bar{A}^T)\right)U - \sigma_w^2\Im(AX^2\bar{A}^T)V = I_n\\ \sigma_w^2\Im(AX^2\bar{A}^T)U + \left(\sigma_z^2 I_n + \sigma_w^2\Re(AX^2\bar{A}^T)\right)V = 0.\end{aligned}$$

These two equations imply that:

$$B^{-1} = \begin{bmatrix}U & -V\\ V & U\end{bmatrix}. \tag{77}$$

This simple observation, enables us to reduce the number of multiplications required for the Newton-Schulz algorithm. More specifically, instead of requiring to multiply two $2m \times 2m$ matrices, we can do 4 multiplications of $m \times m$ matrices. This helps us have a factor of 2 reduction in the cost of matrix-matrix multiplication in our Newton-Schulz algorithm.

In cases the exact inverse calculation is required, again this property enables us to reduce the inversion of matrix $B \in \mathbb{R}^{2m\times 2m}$ to the inversion of two $m \times m$ matrices (albeit a few $m \times m$ matrix multiplications are required as well).

Plugging (77) into (76), we obtain a simplified form for the gradient of $f_L(\mathbf{x})$:

$$\begin{aligned}\frac{\partial f_L}{\partial \mathbf{x}_j} =& 4\mathbf{x}_j\sigma_w^2\Re\left(\bar{\mathbf{a}}_{\cdot,j}^T(U+iV)\mathbf{a}_{\cdot,j}\right) - \frac{2\mathbf{x}_j\sigma_w^2}{L}\sum_{\ell=1}^{L}\left[\Re^2\left(\bar{\mathbf{a}}_{\cdot,j}^T(U+iV)\mathbf{y}_\ell\right) + \Im^2\left(\bar{\mathbf{a}}_{\cdot,j}^T(U+iV)\mathbf{y}_\ell\right)\right]\\ =& 4\mathbf{x}_j\sigma_w^2\Re\left(\bar{\mathbf{a}}_{\cdot,j}^T(U+iV)\mathbf{a}_{\cdot,j}\right) - \frac{2\mathbf{x}_j\sigma_w^2}{L}\sum_{\ell=1}^{L}\left\|\bar{\mathbf{a}}_{\cdot,j}^T(U+iV)\mathbf{y}_\ell\right\|_2^2.\end{aligned} \tag{78}$$

C. Details of our Bagged-DIP-based PGD

Algorithm 1 shows a detailed version of the final algorithm we execute for recovering images from their multilook, speckle-corrupted, undersampled measurements. In one of the steps of the algorithm we ensure that all the pixel values of our estimate are within the range $[0, 1]$. This is because we have assumed that the image pixels take values within $[0, 1]$.

The only remaining parts of the algorithm we need to clarify are, (1) our hyperparameter choices, and (2) the implementation details of the Bagged-DIP module. As described in the main text, in each (outer) iteration of PGD, we learn three DIPs and then take the average of their outputs. Let us now consider one of these DIPs that is applied to one of the $h_k \times w_k$ patches.

Algorithm 1 Iterative PGD algorithm

Input: $\{\mathbf{y}_l\}_{l=1}^L, A, \mathbf{x}_0 = \frac{1}{L}\sum_{l=1}^L |A^T\mathbf{y}_l|, g_\theta(\cdot)$.
Output: Reconstructed $\hat{\mathbf{x}}$.
for $t = 1, \ldots, T$ **do**
  [Gradient Descent Step]
  if $t = 1$ **or** $\|\mathbf{x}^t - \mathbf{x}^{t-1}\|_\infty > \delta_\mathbf{x}$ **then**
    Calculate exact $B_t = (AX_t^2A^T)^{-1}$.
  else
    Approx $\tilde{B}_t = B_{t-1} + B_{t-1}(I_m - AX_t^2A^TB_{t-1})$.
  end if
  Gradient calculation at coordinate j as $\nabla f_L(\mathbf{x}_{t-1,j})$ using B_t or $\tilde{B}_t$, and update $\mathbf{x}_{t,j}^G$: $\mathbf{x}_{t,j}^G \leftarrow \mathbf{x}_{t-1,j} - \mu_t \nabla f_L(\mathbf{x}_{t-1,j})$.
  Save matrix inverse B_t or $\tilde{B}_t$.
  Truncate $\mathbf{x}_t^G$ into range $(0, 1)$, $\mathbf{x}_t^G = \text{clip}(\mathbf{x}_t^G, 0, 1)$.
  [Bagged-DIPs Projection Step]
  Generate random image given randomly generated noise $\mathbf{u} \sim \mathcal{N}(0, 1)$ as $g_\theta(\mathbf{u})$.
  Update θ_t by optimizing over $\|g_\theta(\mathbf{u}) - \mathbf{x}_t^G\|_2^2$: $\theta_t \leftarrow \text{argmin}_\theta \|g_\theta(\mathbf{u}) - \mathbf{x}_t^G\|_2^2$ till converges.
  Generate $\mathbf{x}_t^P$ using trained $g_{\hat{\theta}_t}(\cdot)$ as $\mathbf{x}_t^P \leftarrow g_{\theta_t}(\mathbf{u})$.
  Obtain $\mathbf{x}_t = \mathbf{x}_t^P$.
end for
Reconstruct image as $\hat{\mathbf{x}} = \mathbf{x}_T$.

Inspired by the deep decoder paper (Heckel & Hand, 2018), we construct our neural network, using four blocks: we call the first three blocks DIP-blocks and the last one output block. The strctures of the blocks are shown in Figure 5. As is clear from the figrue, each DIP block is composed of the following components:

- Up sample: This unit increases the hight and width of the datacube that receives by a factor of 2. To interpolate the missing elements, it uses the simple bilinear interpolation. Hence, if the size of the image is 128×128, then the height and width of the input to DIP-block3 will be 64×64, the input of DIP-Block2 will be 32×32, and so on.
- ReLU: this module is quite standard and does not require further explanation.
- Convolution: For all our simulations we have either used 1×1 or 3×3 convolutions. Additionally, we provide details on the number of channels for the data cubes entering each block in our simulations. The channel numbers are [128, 128, 128, 128] for the four blocks.

The output block is simpler than the other three blocks. It only have a 2D convolution that uses the same size as the convolutions of the other DIP blocks. The nonlinearity used here is sigmoid, since we assume that the pixel values are between $[0, 1]$.

Finally, we should mention that each element of the input noise $\mathbf{u}$ of DIP (as described before DIP function is $g_{\boldsymbol{\theta}}(\mathbf{u})$) is generated independently from Normal distribution $\mathcal{N}(0, 1)$.

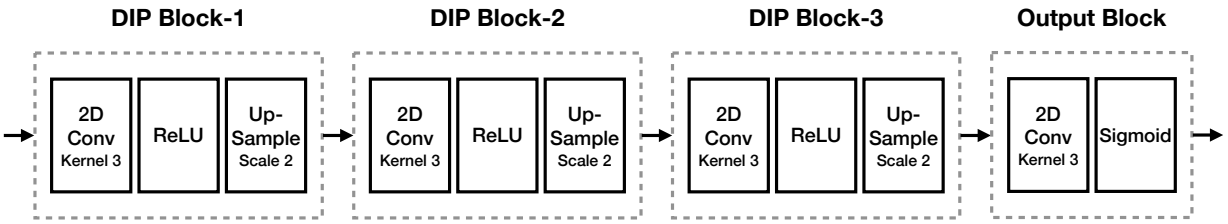

Figure 5. The structure of DIP and Output Blocks.

The other hyperparameters that are used in the DIP-based PGD algorithm are set in the following way: The learning rate of the loglikelihood gradient descent step (in PGD) is set to $\mu = 0.01$. For training the Bagged-DIPs, we use Adam (Kingma & Ba, 2014) with the learning rate set to 0.001 and weight decay set to 0. The number of iterations used for training Bagged-DIPs for different estimates on images are mentioned in Table 3. We run the outer loop (gradient descent of likelihood) for

Patch size of estimates	Barbara	Peppers	House	Foreman	Boats	Parrots	Cameraman	Monarch
128	400	400	400	400	400	800	4k	800
64	300	300	300	300	300	600	2k	600
32	200	200	200	200	200	400	1k	400

Table 3. Number of iterations used in training Bagged-DIPs for different estimates.

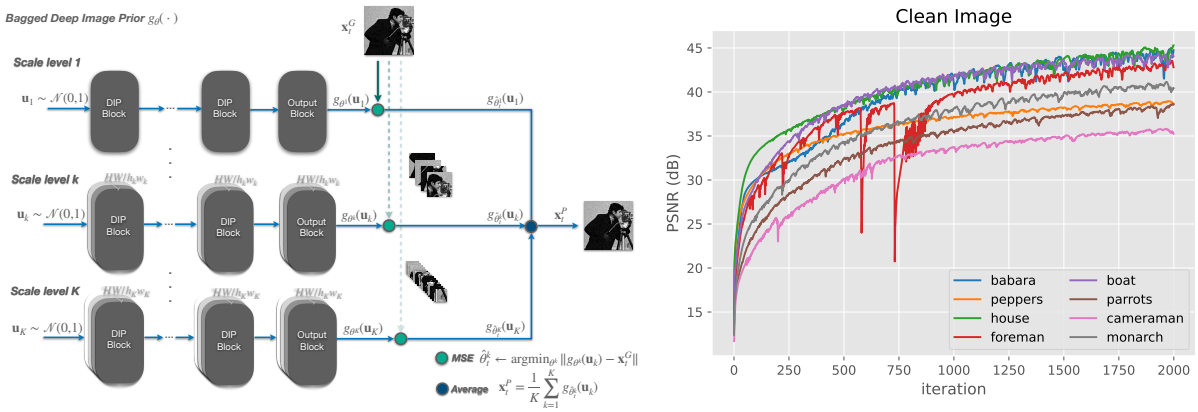

Figure 6. (Left) The structure of Bagged-DIPs with K estimates. (Right) Performance of fitting Bagged-DIP to clean images.

$100, 200, 300$ iterations when $m/n = 0.5, 0.25, 0.125$ respectively. For "Cameraman" only, when $m/n = 0.125$, since the convergence rate is slow, we run 800 outer iterations.

The Newton-Schulz algorithm, utilized for approximating the inverse of matrix B_t, has a quadratic convergence when the maximum singular value $\sigma_{\max}(I - M^0 B_t) < 1$. Hence, ideally, if this condition does not hold, we do not want to use the Newton-Schulz algorithm, and may prefer the exact inversion. Unfortunately, checking the condition $\sigma_{\max}(I - M^0 B_t) < 1$ is also computationally demanding. However, the special form of B_t enables us to have an easier heuristic evaluation of this condition.

For our problems, we establish an empirical sufficient condition for convergence: $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_\infty < \delta_\mathbf{x}$, where $\delta_\mathbf{x}$ is a predetermined constant. To determine the most robust value for $\delta_\mathbf{x}$, we conducted simple experiments. We set $n = 128 \times 128$ and $m/n = 0.5$. The sensing matrix A is generated as described in the main part of the paper (see Section 5.2). Each element of $\mathbf{x}_o$ is independently drawn from a uniform distribution $U[0.001, 1]$. Furthermore, each element of $\Delta\mathbf{x}_o$ is independently sampled from a two-point distribution. In this distribution, the probability of the variable X being $\delta_\mathbf{x}$ is equal to the probability of X being $-\delta_\mathbf{x}$, both with a probability of 0.5, ensuring $\|\Delta\mathbf{x}_o\|_\infty = \delta_\mathbf{x}$. We define B as $A(X + \Delta X_o)^2 \bar{A}^T$, and M^0 as $(AX^2\bar{A}^T)^{-1}$. We then assess the convergence of the Newton-Schulz algorithm for calculating B^{-1}. For various values of $\delta_\mathbf{x}$, we ran the simulation 100 times each, recording the convergence success rate. As indicated in Table 4, the algorithm demonstrates instability when $\delta_\mathbf{x} \geq 0.13$. Consequently, we set $\delta_\mathbf{x}$ to 0.12 in all our simulations to ensure the reliable convergence of the Newton-Schulz algorithm.

$\delta_\mathbf{x}$	convergence success rate
0.1	100%
0.11	100%
0.12	100%
0.13	38%
0.14	0%
0.15	0%

Table 4. Convergence success rate for different thresholds.

D. Additional experiments.

D.1. Comparison with classical despeckling.

We use DnCNN (Zhang et al., 2017b;a) as the neural networks for despeckling task. The DnCNN structure we use consists of one input block, eight DnCNN block and one output block. The details are shown in Figure 7. The number of channels

for each convolutional layer is 64.

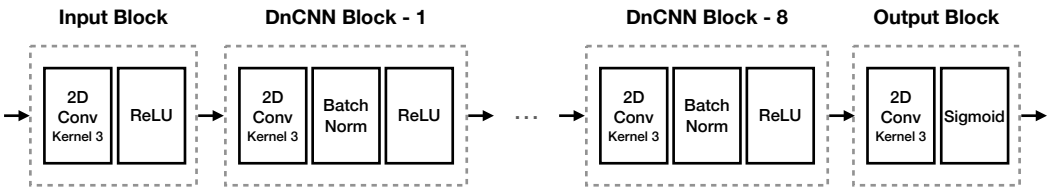

Figure 7. The structure of DnCNN, Input and Output Blocks.

Since there is no sensing matrix A in this simulation, we consider real-valued speckle noise $\mathbf{w}$ (similar to what we considered in our theoretical work) to make the number of measurements the same (with the same number of looks), and make the comparisons simpler. The training set we use for DnCNN is BSD400 (Martin et al., 2001), we divide the images in training set to be 128×128, with stride step 32. The learning rate is 1e-4, batch size is 64, it takes 20 epochs for the training to converge. Table 5 compares the results of this simulation, with the results of the simulation we presented in the main text for fifty percent downsampled measurement matrix. The results of DnCNN-UB are often between 1-3dB better than the results of our Bagged-DIPs-based PGD. With the exception of the cameraman, where our Bagged-DIPs-based PGD seems to be better. It should be noted that the gain obtained by DnCNN here should not be only associated to the fact that the matrix A is undersampled. There is one major difference that may be contributing to the improvements that we see in Table 5, and that is we are also using a training set, while our DIP-based method does not use any training data.

m/n	#looks	Barbara	Peppers	House	Foreman	Boats	Parrots	Cameraman	Monarch	Average
50%	25	27.30/0.759	27.02/0.724	28.56/0.697	27.56/0.735	26.21/0.669	25.94/0.728	27.95/0.762	27.17/0.845	27.21/0.740
	50	28.67/0.816	28.52/0.804	30.30/0.762	28.88/0.827	27.58/0.739	27.23/0.799	30.21/0.843	28.86/0.898	28.78/0.818
	100	29.40/0.843	29.21/0.849	31.61/0.815	29.74/0.871	28.45/0.785	28.20/0.848	31.58/0.902	30.05/0.932	29.78/0.856
DnCNN-UB	25	28.77/0.832	29.42/0.830	29.90/0.799	31.26/0.820	28.51/0.788	28.19/0.858	28.77/0.911	29.40/0.925	29.28/0.845
	50	30.30/0.873	30.93/0.868	30.92/0.832	31.54/0.859	29.87/0.839	29.07/0.888	29.84/0.925	30.63/0.940	30.40/0.878
	100	32.18/0.903	32.38/0.899	32.45/0.866	34.28/0.900	31.59/0.865	29.77/0.902	29.83/0.896	31.27/0.955	31.72/0.898

Table 5. PSNR(dB)/SSIM ↑ of $m/n = 50\%$, $L = 25/50/100$. DnCNN despeckling tasks are used for showing the performance gap between our method with 50% downsampled complex valued measurements and the corresponding empirical uppper bound.

D.2. Comparison with (Chen et al., 2023)

As discussed before, a recent paper has also considered the problem of recovering an image from undersampled/ill-conditioned measurement kernels in the presence of the speckle noise (Chen et al., 2023) and they also considered a DIP-based approach. The goal of this section is to provide some comparison in the performance of our DIP-based method and the one presented in (Chen et al., 2023).

While (Chen et al., 2023) considered real-valued speckle noises and measurements, we adapt their approch to the complex-valued settings under which we have run our experiments. The results of our comparisons are presented in Table 6. As is clear in this table (Chen et al., 2023) considered two different algorithms DIP-simple and DIP-M^3. DIP-simple is the DIP-based PGD with filter size = 1 and the number of channels were chosen as $[100, 50, 25, 10]$. In DIP-M^3, the same DIP was chosen. But the authors also used λ residual connection to balance the contribution from gradient descent and projection outputs as follows:

$$\mathbf{x}_t = \lambda \mathbf{x}_t^P + (1-\lambda)\mathbf{x}_t^G,$$

where $\mathbf{x}_t^P$ and $\mathbf{x}_t^G$ are gradient descent and projection results respectively. Similar to the setting of that paper we consider, where smaller λ is used when L increase, so we set the hyperparameter $\lambda = 0.3, 0.2, 0.1$ for $L = 25, 50, 100$.

We should note that choosing optimal λ is tricky for different m/n and L. Setting a small λ means that a large portion of projection has been bypassed, which indicates the limit on learning capacity of simple DIP. To verify the statement that a simple DIP has very limited learning capacity on some images, we also provide the baseline, DIP-simple, which distingushes from DIP-M^3 by setting $\lambda = 1.0$. This means we use the projection results fully from the DIP-simple, and it fails on several tasks as we can see from Table 6, and especially the case we mention in the right panel of Figure 3.

It's important to highlight that Bagged-DIPs do not incorporate residual connections to bypass the projection, essentially representing the case where $\lambda = 1.0$. Specifically, we solely rely on the projection from Bagged-DIPs. The outcomes presented in Table 6 demonstrate the robust projection capabilities of Bagged-DIPs, leading to superior performance compared to DIP-simple and DIP-M^3.

m/n	#looks	DIP-simple	DIP-M^3	Bagged-DIPs
12.5%	25	18.03/0.336	17.81/0.316	**19.24/0.406**
	50	19.25/0.408	18.96/0.382	**20.83/0.538**
	100	20.15/0.497	19.94/0.464	**21.78/0.612**
25%	25	22.00/0.493	21.69/0.474	**22.86/0.549**
	50	23.42/0.572	23.39/0.551	**24.95/0.672**
	100	24.79/0.656	25.08/0.629	**26.24/0.745**
50%	25	25.62/0.683	26.01/0.668	**27.21/0.740**
	50	26.81/0.749	27.81/0.733	**28.78/0.818**
	100	27.53/0.799	29.52/0.779	**29.78/0.856**

Table 6. Average PSNR(dB)/SSIM ↑ comparison of baseline methods, $m/n = 12.5\%/25\%/50\%$, $L = 25/50/100$.

D.3. Bagging performance.

We show the comparison of Bagged-DIPs with three sophisticated DIP estimates in Figure 8. In most of the test images, the bagging of three estimates yields a performance improvement over all individual estimates. As described in the main part of the paper, this is expected when all the estimates are low-bias and are weakly-dependent.

However, there are exceptions, such as the case of "Foreman." In such instances, one of the estimates appears to surpass our average estimate. This occurs when certain individual estimates are affected by large biases. While these biases are mitigated to some extent in our average estimate, a residual portion persists, affecting the overall performance of the average estimate. There are a few directions one can explore to resolve this issue and we leave them for future research. For instance, we can create more bagging samples, and then use more complicated networks without worrying about the overfitting. That will alleviate the issue of high bias that exists in a few images.

Since averaging the estimates may results in blurring issues on the image, we further quantitatively measure the sharpness of the images by using gradient magnitude similarity deviation (GMSD) (Xue et al., 2013). Table 7 below reports PSNR, GMSD, and SSIM of the averaged (i.e., bagged) output, and compares it against the performance achieved by individual estimates. The test image here is Cameraman with $m/n = 0.5$, and $L = 50$. It can be observed that throughout the iterations, the averaged output, i.e., the output of Bagged-DIP, achieves the best performance in terms of PSNR, GMSD and SSIM.

Estimate	ite 5	ite 10	ite 15	ite 20	ite 25	ite 30
estimate 1	20.58/0.0793/0.684	27.72/0.0728/0.859	29.17/0.0851/0.844	29.64/0.0888/0.842	29.58/0.0905/0.838	29.62/0.0927/0.837
estimate 2	20.41/0.0825/0.660	27.64/0.0797/0.840	29.25/0.0836/0.826	29.46/0.0900/0.823	29.22/0.0957/0.813	29.06/0.1017/0.809
estimate 3	20.68/0.0851/0.644	27.76/0.0809/0.818	29.57/0.0909/0.791	29.20/0.0999/0.770	29.14/0.1014/0.765	29.02/0.1054/0.762
estimate avg	**20.68/0.0788/0.683**	**27.95/0.0715/0.858**	**29.76/0.0799/0.845**	**30.12/0.0843/0.841**	**30.14/0.0875/0.838**	**30.00/0.0916/0.835**

Table 7. PNSR(dB)/GMSD/SSIM of the reconstruction of the Cameraman at the different iterations of the PGD algorithm. estimates 1, 2 and 3 denote the estimates obtained with 32×32, 64×64 and 128×128 patch sizes respectively. Estimate avg, exhibits the performance of the averaged signal.

E. Time cost of PGD algorithm.

We provide the timing details of training the Bagged-DIPs-based PGD algorithm in Table 8. The time cost of each iteration in PGD is affected by m/n, and the iterations needed for training Bagged-DIPs. The experiments are performed on Nvidia RTX 6000 GPUs, and we record the time it uses accordingly for different tasks.

Bagged-DIPs training iterations	12.5%	25%	50%
200, 300, 400	~ 65	~ 75	~ 105
400, 600, 800	~ 115	~ 125	~ 155
1k, 2k, 4k	~ 330	~ 340	~ 370

Table 8. Time (in seconds) required for each iteration of the PGD with different sampling rate and iterations for training Bagged-DIPs.

We also find that, compared with initializing $\mathbf{x}_0$ with fixed values, using initialization $\mathbf{x}_0 = \frac{1}{L}\sum_{l=1}^{L} |\bar{A}^T \mathbf{y}_l|$ helps improve the convergence rate. But the final reconstructed performance does not depend on the initialization methods we compare. The effect of input initialization in PGD algorithm is shown in Figure 9.

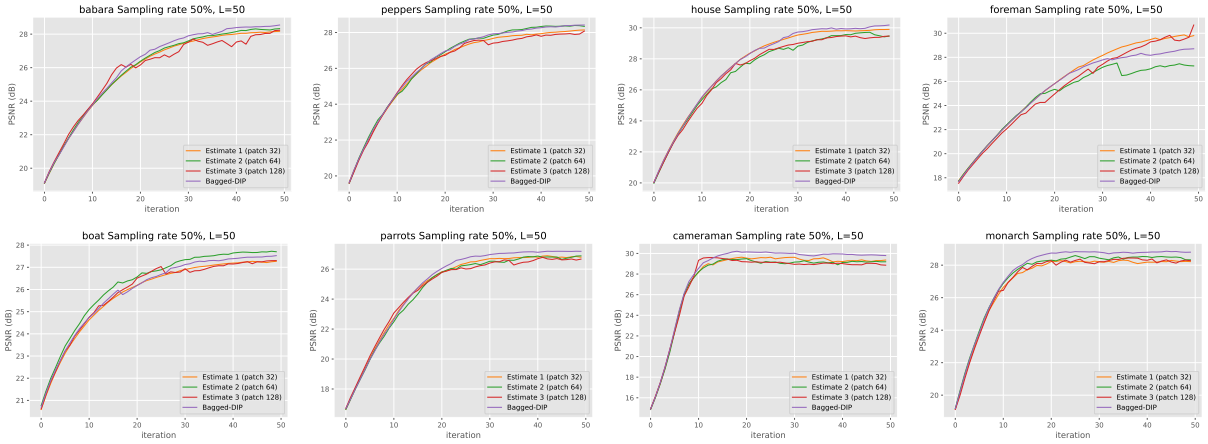

Figure 8. Comparison of Bagged-DIPs and three sophisticated DIP estimates on 8 images, $m/n = 0.5$, $L = 50$.

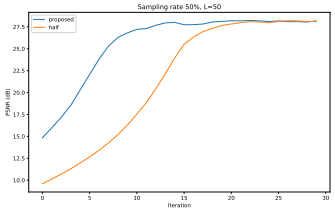

Figure 9. Comparison between two initializations: (1) proposed: our inititalization method; (2) half: a vector whose elements are all 0.5.

We should also underscore that there exist several straightforward computational optimizations that can enhance the runtime of our algorithm. For instance, in our simulations, we opted for a conservative and small step size (μ_t) in our PGD algorithm to ensure convergence across all scenarios and all images. By employing adaptive step sizes, one can significantly reduce the runtime. Supporting this assertion, the results in Table 9 for the test image Barbara ($m/n = 0.5, L = 50$) demonstrate that similar PSNR performance can be achieved using much larger values of μ_t. The required number of iterations for these larger values is substantially smaller. For example, setting $\mu_t =$ **0.1** results in the algorithm converging within **5** outer iterations, thereby reducing the runtime by a factor of 10.

Step size for PGD	**ite 1**	**ite 2**	**ite 3**	**ite 4**	**ite 5**	**ite 10**	**ite 20**	**ite 30**	**ite 40**	**ite 50**
$\mu_t = 0.01$	19.13	19.76	20.33	20.77	21.26	23.41	26.46	27.77	28.35	**28.54**
$\mu_t = 0.02$	19.81	20.97	22.03	22.93	23.70	26.78	**28.52**	28.67	28.69	28.66
$\mu_t = 0.05$	22.04	24.24	25.87	27.10	27.73	**28.58**	28.51	28.47	28.45	28.45
$\mu_t = 0.1$	24.81	27.57	28.19	**28.33**	28.30	28.28	28.22	28.22	28.22	28.14

Table 9. Each row of the table displays the PSNRs of the estimates obtained at various iterations of Bagged-DIP PGD with a specific step-size indicated in the first column.

F. Qualitative results of Bagged-DIPs-based PGD

We show the qualitative results of the Bagged-DIPs-based PGD algorithm that are reconstructed from $L = 25, 50, 100$ looks of $12.5, 25, 50\%$ downsampled complex-valued measurements. Row 1-3 in Figure 10 are $m/n = 0.125$ with $L = 25, 50, 100$ respectively, row 1-3 in Figure 11 are $m/n = 0.25$ with $L = 25, 50, 100$ respectively, row 1-3 in Figure 12 are $m/n = 0.5$ with $L = 25, 50, 100$ respectively. We also report the PSNR/SSIM under each reconstructed image.

An evident advantage observed in the reconstructed images is that utilizing bagging estimates with various patch sizes helps alleviate the blocking issue that may arise when relying solely on a single patch size (e.g., patch size = 32). This is because boundaries in smaller patch sizes do not necessarily align with boundaries in larger patch sizes, and aggregating estimates

from different patch sizes through averaging can effectively mitigate the blocking issue. Additionally, we visualize the reconstructed images from individual estimates and bagged estimate in Figure 13. We observe that there is no blocking issue in the bagged estimate.

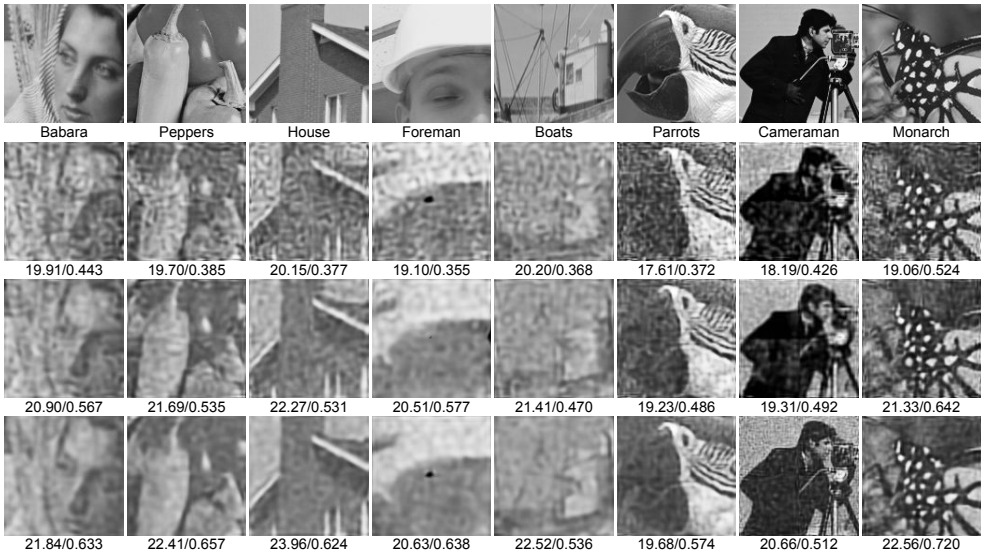

Figure 10. Reconstructed images from $L = 25, 50, 100$ looks of 12.5% downsampled complex-valued measurements. Row 1-3 are $m/n = 0.125$ with $L = 25, 50, 100$ respectively.

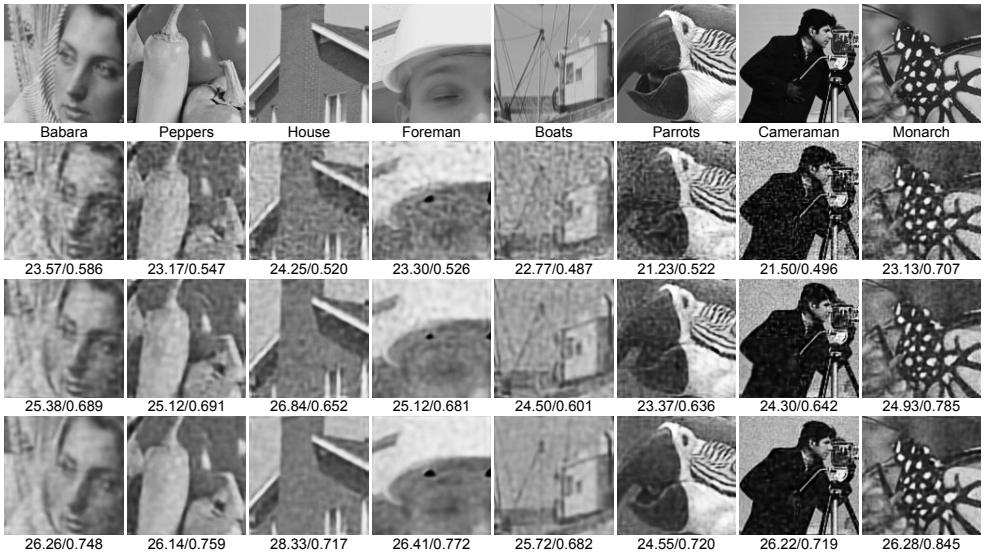

Figure 11. Reconstructed images from $L = 25, 50, 100$ looks of 25% downsampled complex-valued measurements. Row 1-3 are $m/n = 0.25$ with $L = 25, 50, 100$ respectively.

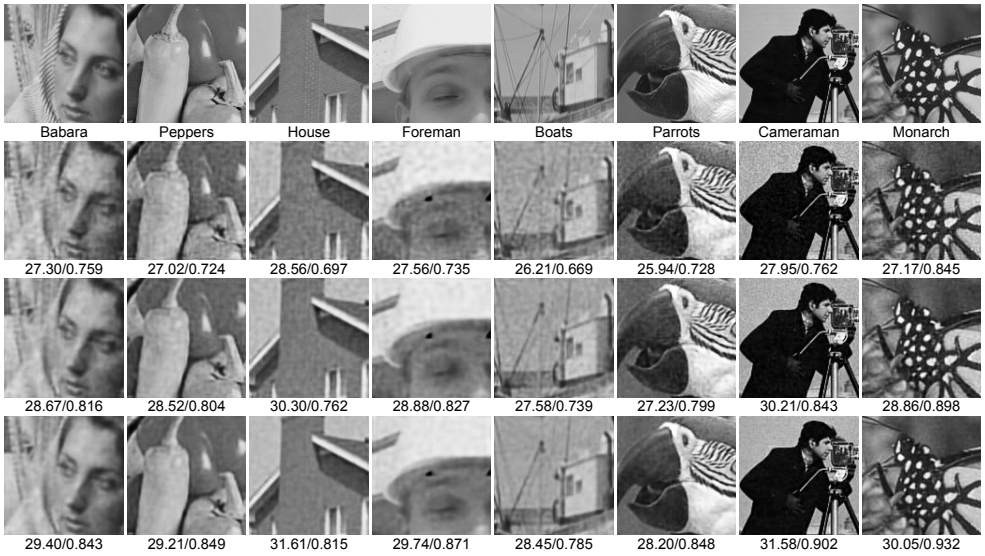

Figure 12. Reconstructed images from $L = 25, 50, 100$ looks of 50% downsampled complex-valued measurements. Row 1-3 are $m/n = 0.5$ with $L = 25, 50, 100$ respectively.

Figure 13. We visualize the reconstructed images Cameraman with $m/n = 0.5, L = 50$. From left to right: estimate 1 (patch size 32), estimate 2 (patch size 64), estimate 3 (patch size 128), bagged estimate.