Inexact Alternating Optimization for Phase Retrieval In the Presence of Outliers

Cheng Qian, *Student Member, IEEE,* Xiao Fu, *Member, IEEE*, Nicholas D. Sidiropoulos, *Fellow, IEEE*,
Lei Huang, *Senior Member, IEEE*, and Junhao Xie, *Senior Member, IEEE*

Abstract

Phase retrieval has been mainly considered in the presence of Gaussian noise. However, the performance of the algorithms proposed under the Gaussian noise model severely degrades when grossly corrupted data, i.e., outliers, exist. This paper investigates techniques for phase retrieval in the presence of heavy-tailed noise – which is considered a better model for situations where outliers exist. An ℓ_p-norm ($0 < p < 2$) based estimator is proposed for fending against such noise, and two-block inexact alternating optimization is proposed as the algorithmic framework to tackle the resulting optimization problem. Two specific algorithms are devised by exploring different local approximations within this framework. Interestingly, the core conditional minimization steps can be interpreted as iteratively reweighted least squares and gradient descent. Convergence properties of the algorithms are discussed, and the Cramér-Rao bound (CRB) is derived. Simulations demonstrate that the proposed algorithms approach the CRB and outperform state-of-the-art algorithms in heavy-tailed noise.

Index Terms

Phase retrieval, iterative reweighted least squares (IRLS), gradient descent, impulsive noise, Cramér-Rao bound (CRB).

I. Introduction

Phase retrieval aims at recovering a signal $\mathbf{x} \in \mathbb{C}^N$ from only the magnitude of linear measurements. This is an old problem [2], [3] that has recently attracted renewed and growing interest. Phase retrieval arises in many fields, such as X-ray crystallography, coherent diffraction imaging, and optical imaging and astronomy, where the detectors only record the intensity information, because phase is very difficult and expensive to measure.

The mathematical description of the phase retrieval problem is simple: given the measuring matrix $\mathbf{A} = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_M]^H \in \mathbb{C}^{M \times N}$ and the measurement vector $\mathbf{y} \in \mathbb{R}^M$, where $\mathbf{a}_m \in \mathbb{C}^N$, and

$$\mathbf{y} = |\mathbf{A}\mathbf{x}|,$$

Original manuscript submitted to *IEEE Trans. on Signal Processing*, April 23, 2016. Preliminary conference version of part of this work has been submitted to *EUSIPCO 2016* [1]. C. Qian (e-mail: qianc@umn.edu) and J. Xie (e-mail: xj@hit.edu.cn) are with the Department of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, 150001 China. C. Qian is currently a long-term visitor in the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, USA, as a Ph.D. exchange student supported by the Chinese Scholarship Council.

X. Fu and N. D. Sidiropoulos are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: xfu@umn.edu, nikos@umn.edu).

L. Huang is with the College of Information Engineering, Shenzhen University, Shenzhen, 518060 China (e-mail: dr.lei.huang@ieee.org).

The work of N. Sidiropoulos was supported by NSF CIF-1525194.

find $\mathbf{x} \in \mathbb{C}^N$. Early attempts to solve the phase retrieval problem can be traced back to the 1970s, where techniques such as Gerchberg-Saxton (GS) [2], Fienup [3] and their variants were proposed. These algorithms have been empirically shown to work well under certain conditions, although little had been known regarding their convergence properties from a theoretical point of view. Recently, some new algorithms along this line of work were proposed in [4], where the convergence issue is better studied.

In recent years, more modern optimization-based approaches for phase retrieval have been proposed. For example, Candès proposed a semidefinite relaxation approach known as the *PhaseLift* algorithm [8] and proved that exact recovery is possible with high probability in the noiseless case. Wirtinger-Flow (WF) [5] is a more recent approach that combines a good statistical initialization with a computationally light gradient-type refinement algorithm. The combination works very well when $\mathbf{A}$ is i.i.d. Gaussian. In our recent work [7], we proposed a least-squares *feasible point pursuit* (LS-FPP) approach that aims to solve the same optimization problem as LS PhaseLift in the presence of noise. Simulations in [6] indicate that LS-FPP approaches the Cramér-Rao bound for the Gaussian measurement model considered in [6].

Some phase retrieval algorithms were originally developed under an exact measurement model, and subsequently treated the noisy case by replacing equality constraints with relaxed inequality constraints [8]-[9]. Most of the existing algorithms were explicitly or implicitly developed under a Gaussian noise model. In certain applications, a subset of the measurements may be corrupted much more significantly than the others, and heavy-tailed noise may be encountered as well [11]-[15]. One representative example is high energy coherent X-ray imaging using a charge-coupled device (CCD), where the impulsive noise originates from X-ray radiation effects on the CCD, and the density of impulses increases with the intensity of X-ray radiation or CCD exposure time [15]. Under such circumstance, modeling the noise as Gaussian is no longer appropriate.

In recent years, robust phase retrieval algorithms, e.g., [11], [14], have been proposed to handle outliers. The framework in [13], [14] considered an undersampled phase retrieval model corrupted with Laplacian-distributed outliers, but the approach was designed specifically for sparse $\mathbf{x}$. It is worth noting that Chen and Candés [16] suggested a truncated WF (TWF) algorithm to handle Poisson noise, which also exhibits robustness to outliers under certain conditions. However, this method is sensitive to the choice of the measuring matrix $\mathbf{A}$ in practice, as will be shown in simulations in Section IV.

Another important aspect of phase retrieval is how noise enters the measurement model. Our previous work [6] considered the noise model $\mathbf{y} = |\mathbf{A}\mathbf{x}|^2 + \mathbf{n}$. Another noise model that is frequently considered in the literature [2]-[4], [10] is

$$\mathbf{y} = |\mathbf{A}\mathbf{x}| + \mathbf{n}. \tag{1}$$

Notice the subtle but important difference between the two models: whether noise is added to the magnitude or the squared magnitude. The choice hinges on the experimental setup, including the measurement apparatus; but (1) is more widely adopted by experimentalists.

Contributions: We consider the phase retrieval problem under the model in (1) in the presence of impulsive noise, and focus on designing robust algorithms to handle it. To fend against impulsive noise, we adopt the ℓ_p-fitting ($0 < p < 2$) based estimator that is known to be effective in dealing with outliers, and devise two optimization algorithms using two-block inexact alternating optimization. Specifically, the two algorithms both solve local majorization-based approximate subproblems for one block, instead of exactly solving the conditional block minimization problem to optimality. Interestingly, starting from different majorizations, the resulting solutions turn out equivalent to iteratively reweighted least squares and gradient descent, respectively. Unlike the existing inexact and exact alternating optimization frameworks that mostly operate with convex constraints, the proposed algorithms work with a unit-modulus nonconvex constraint, so convergence analysis seems difficult. Nevertheless, we prove convergence of the proposed algorithms to a Karush-Kuhn-Tucker (KKT) point by exploiting the two-block structure of the problem. We also derive computationally light implementations using Nesterov-type and stochastic gradient updates. In order to assist in performance analysis and experimental design, we derive the Cramér-Rao bound (CRB) for the model in (1) and under different parameterizations, in the presence of Laplacian and Gaussian noise. Curiously, although related CRBs for different noisy measurement models have been previously derived in [6], [17]-[21], to the best of our knowledge, there is no available CRB for the model in (1) – and our work fills this gap. The proposed algorithms are validated by extensive simulations. The simulations show that our approaches outperform the state-of-the-art algorithms in the presence of impulsive noise.

A preliminary conference version of part of this work has been submitted to EUSIPCO 2016 [1]. The conference version includes one of the two basic algorithms, the Laplacian CRB, and limited simulations. The second algorithm, Nesterov acceleration and stochastic gradient-type updates, additional CRB results, and, most importantly, proof of convergence of the iterative algorithms are all provided only in this journal version, which naturally also includes more comprehensive simulation results.

Notation: Throughout the paper, we use boldface lowercase letters for vectors and boldface uppercase letters for matrices. Superscripts $(\cdot)^T$, $(\cdot)^*$, $(\cdot)^H$, $(\cdot)^{-1}$ and $(\cdot)^\dagger$ represent transpose, complex conjugate, conjugate transpose, matrix inverse and pseudo-inverse, respectively. The Re$\{\cdot\}$ and Im$\{\cdot\}$ denote the real part and imaginary part. $E[\cdot]$ is the expectation operator, $|\cdot|$ is the absolute value operator, $||\cdot||_F$ is the Frobenius norm, $||\cdot||_p$ is the vector ℓ_p-norm, whose definition is $||\mathbf{x}||_p = (\sum_{i=1}^N |x_i|^p)^{1/p}$, $\odot$ is the element-wise product, and diag$(\cdot)$ is a diagonal matrix with its argument on the diagonal. δ_{ij} denotes the Kronecker delta function, and $\angle(\cdot)$ takes the phase of its argument. $\mathbf{0}_m$, $\mathbf{1}_m$, and $\mathbf{I}_m$ stand for the $m \times 1$ all-zero vector, $m \times 1$ all-one vector, and $m \times m$ identity matrix, respectively. Furthermore, tr$(\cdot)$ and $\partial a/\partial x$ denote the trace and partial derivative operators, respectively.

II. Proposed Algorithms

A. AltIRLS

Let us first consider the noiseless case where $\mathbf{y} = |\mathbf{A}\mathbf{x}|$. Effectively, it can also be written as

$$\mathbf{y} \odot \mathbf{u} = \mathbf{A}\mathbf{x} \tag{2}$$

where $\mathbf{u} = e^{j\angle(\mathbf{A}\mathbf{x})}$ is an auxiliary vector of unknown unit-modulus variables with its mth component being $u_m = e^{j\angle(\mathbf{a}_m^H\mathbf{x})}$. In the presence of impulsive noise, ℓ_p-(quasi)-norm has be recognized as an effective tool for promoting sparsity and fending against outliers [22]-[26]. Therefore, we propose to employ an ℓ_p-fitting based estimator instead of using the ℓ_2-norm, which has the form of

$$\min_{|\mathbf{u}|=\mathbf{1},\mathbf{x}} \sum_{m=1}^{M} \left(|y_m u_m - \mathbf{a}_m^H\mathbf{x}|^2 + \epsilon\right)^{p/2} \tag{3}$$

where $0 < p < 2$ is chosen to down-weigh noise impulses (i.e., outliers), and $\epsilon > 0$ is a small regularization parameter that keeps the cost function within its differentiable domain when $p < 1$, which will prove handy in devising an effective algorithm later. When $1 < p < 2$, we may simply let $\epsilon = 0$.

To handle Problem (3), we follow the rationale of alternating optimization, i.e., we first update $\mathbf{x}$ fixing $\mathbf{u}$, and then we do the same for $\mathbf{u}$.

Assume that after some iterations, the current solution at iteration r is $(\mathbf{x}^{(r)}, \mathbf{u}^{(r)})$. At step $(r+1)$, the subproblem with respect to (w.r.t.) $\mathbf{x}$ is

$$\mathbf{x}^{(r+1)} = \arg\min_{\mathbf{x}} \sum_{m=1}^{M} \left(|y_m u_m^{(r)} - \mathbf{a}_m^H \mathbf{x}|^2 + \epsilon \right)^{p/2} \tag{4}$$

which is still difficult to handle. Particularly, when $p < 1$, the subproblem itself is still non-convex; when $p \geq 1$, the subproblem is convex but has no closed-form solution. Under such circumstances, we propose to employ the following lemma [27]:

Lemma 2.1: Assume $0 < p < 2$, $\epsilon \geq 0$, and $\phi_p(w) := \frac{2-p}{2}\left(\frac{2}{p}w\right)^{\frac{p}{p-2}} + \epsilon w$. Then, we have

$$\left(x^2 + \epsilon\right)^{p/2} = \min_{w \geq 0} \; w x^2 + \phi_p(w), \tag{5}$$

and the unique minimizer is

$$w_{\text{opt}} = \frac{p}{2}\left(x^2 + \epsilon\right)^{\frac{p-2}{2}}. \tag{6}$$

By Lemma 2.1, an upper bound of $\sum_{m=1}^{M} \left(|y_m u_m^{(r)} - \mathbf{a}_m^H \mathbf{x}|^2 + \epsilon \right)^{p/2}$ that is tight at the current solution $\mathbf{x}^{(r)}$ can be easily found:

$$\begin{aligned} &\sum_{m=1}^{M} \left(|y_m u_m^{(r)} - \mathbf{a}_m^H \mathbf{x}|^2 + \epsilon \right)^{p/2} \\ &\qquad \leq \sum_{m=1}^{M} \left(w_m^{(r)} \left| y_m u_m^{(r)} - \mathbf{a}_m^H \mathbf{x} \right|^2 + \phi_p\left(w_m^{(r)}\right) \right) \end{aligned} \tag{7}$$

where

$$w_m^{(r)} := \frac{p}{2}\left(\left| y_m u_m^{(r)} - \mathbf{a}_m^H \mathbf{x}^{(r)} \right|^2 + \epsilon \right)^{\frac{p-2}{2}}, \tag{8}$$

and the equality holds if and only if $\mathbf{x} = \mathbf{x}^{(r)}$. Instead of directly dealing with Problem (4), we solve a surrogate problem using the right hand side (RHS) of (7) at each iteration to update $\mathbf{x}$. Notice that the RHS of (7) is convex w.r.t. $\mathbf{x}$ and the corresponding problem can be solved in closed-form:

$$\mathbf{x}^{(r+1)} = \left(\mathbf{W}^{(r)}\mathbf{A}\right)^{\dagger} \mathbf{W}^{(r)} \left(\mathbf{y} \odot \mathbf{u}^{(r)}\right) \tag{9}$$

where

$$\mathbf{W}^{(r)} = \operatorname{diag}\left(\sqrt{w_1^{(r)}} \;\; \cdots \;\; \sqrt{w_M^{(r)}} \right). \tag{10}$$

The conditional problem w.r.t. $\mathbf{u}$ is

$$\mathbf{u}^{(r+1)} = \arg\min_{|\mathbf{u}|=\mathbf{1}} \sum_{m=1}^{M} \left(|y_m u_m - \mathbf{a}_m^H \mathbf{x}^{(r+1)}|^2 + \epsilon \right)^{p/2}. \tag{11}$$

Although the problem is non-convex, it can be easily solved to optimality. Specifically, the first observation is that the partial minimization w.r.t. $\mathbf{u}$ is insensitive to the value of p; i.e., given a fixed $\mathbf{x}$, for any $p > 0$, the solutions w.r.t. $\mathbf{u}$ are identical. Second, for all $p > 0$, the solution is simply to align the angle of $y_m u_m$ to that of $\mathbf{a}_m^H \mathbf{x}^{(r+1)}$, which is exactly

$$u_m^{(r+1)} = e^{j\angle\left(\mathbf{a}_m^H \mathbf{x}^{(r+1)}\right)}, \quad m = 1, \ldots, M. \tag{12}$$

We update $\mathbf{x}$ and $\mathbf{u}$ alternately, until some convergence criterion is met. We see that the way that we construct the upper bound of the partial problem w.r.t. $\mathbf{x}$ is in fact the same as the procedure in iteratively reweighted least squares (IRLS) [27], [29]. The difference is that we 'embed' this IRLS step into an alternating optimization algorithm. We therefore call this algorithm alternating IRLS (AltIRLS), which is summarized in Algorithm 1.

Algorithm 1 AltIRLS for phase retrieval

```
function AltIRLS ($\mathbf{y}, \mathbf{A}, \mathbf{x}^{(0)}$)
    Initialize $\mathbf{u}^{(0)} = \exp(\angle(\mathbf{A}\mathbf{x}^{(0)}))$ and $\mathbf{W}^{(0)} = \mathbf{W}^{(0)} = \mathrm{diag}\left(\sqrt{w_1^{(0)}} \cdots \sqrt{w_M^{(0)}}\right)$ with $w_m^{(0)} = \frac{p}{2}\left(\left|y_m u_m^{(0)} - \mathbf{a}_m^H \mathbf{x}^{(0)}\right|^2 + \epsilon\right)^{\frac{p-2}{2}}$
    while stopping criterion has not been reached do
        $\mathbf{x}^{(r)} = (\mathbf{W}^{(r-1)}\mathbf{A})^{\dagger}\mathbf{W}^{(r-1)}(\mathbf{y} \odot \mathbf{u}^{(r-1)})$.
        $\mathbf{u}^{(r)} = \exp(j\angle(\mathbf{A}\mathbf{x}^{(r)}))$
        $w_m^{(r)} = \frac{p}{2}\left(\left|y_m u_m^{(r)} - \mathbf{a}_m^H \mathbf{x}^{(r)}\right|^2 + \epsilon\right)^{\frac{p-2}{2}}, \forall m$
        $\mathbf{W}^{(r)} = \mathrm{diag}\left(\sqrt{w_1^{(r)}} \cdots \sqrt{w_M^{(r)}}\right)$
        $r = r + 1$
    end while
end function
```

A relevant question regarding Algorithm 1 is whether or not this algorithm converges to a meaningful point, e.g., a stationary or KKT point. Note that the block variable $\mathbf{u}$ is constrained to a non-convex set, and we do not compute the optimal solution for the block variable $\mathbf{x}$ at each iteration of Algorithm 1. For such a type of algorithm, there is no existing theoretical framework that establishes convergence. We therefore need careful custom convergence analysis. We have the following result:

Proposition 2.1: Every limit point of the solution sequence produced by Algorithm 1 is a KKT point of Problem (3).

Proof: See Appendix A. ■

The result in Proposition 2.1 is interesting: Although the algorithm that we propose to compute the ℓ_p fitting-based estimator solves non-convex subproblems, it ensures convergence to a KKT point. Such a nice property is proven by exploiting the two-block structure of the algorithm. Note that the proof itself does not rely on the specific form of the optimization problem in (3), and thus can be easily extended to other two-block alternating optimization cases, which we believe is of much broader interest.

B. AltGD

Algorithm 1 uses a simple update strategy, but its complexity may still become an issue when the problem size grows. Specifically, the bottleneck of the AltIRLS algorithm lies in solving the subproblem

$$\mathbf{x}^{(r+1)} = \arg\min_{\mathbf{x}} \left|\left|\mathbf{W}^{(r)}(\mathbf{y} \odot \mathbf{u}) - \mathbf{W}^{(r)}\mathbf{A}\mathbf{x}\right|\right|_2^2. \tag{13}$$

Although the above problem is merely a quadratic program with closed-form solution, i.e.,

$$\mathbf{x}^{(r+1)} = \left(\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A}\right)^{-1}\mathbf{A}^H(\mathbf{W}^{(r)})^2\left(\mathbf{y} \odot \mathbf{u}^{(r)}\right), \tag{14}$$

the matrix inversion part requires $\mathcal{O}(N^3)$ flops to compute and also $\mathcal{O}(N^2)$ memory to store, both of which are not efficient for high-dimensional $\mathbf{x}$ – e.g., when $\mathbf{x}$ is a vectorized image, N is usually very large (more specifically, for a 100×100 image, N is 10,000).

To circumvent this difficulty, we propose to deal with problem (13) using a first-order approach. Specifically, let us denote

$$f^{(r)}(\mathbf{x}) = \left|\left|\mathbf{W}^{(r)}(\mathbf{y} \odot \mathbf{u}^{(r)}) - \mathbf{W}^{(r)}\mathbf{A}\mathbf{x}\right|\right|_2^2 \tag{15}$$

and approximate (13) using the following function

$$\begin{aligned}\mathbf{g}^{(r)}(\mathbf{x}) = f^{(r)}(\mathbf{x}^{(r)}) + \mathrm{Re}\{(\nabla f^{(r)}(\mathbf{x}^{(r)}))^H(\mathbf{x} - \mathbf{x}^{(r)})\} \\ + \frac{\mu^{(r)}}{2}\left\|\mathbf{x} - \mathbf{x}^{(r)}\right\|_2^2.\end{aligned} \tag{16}$$

where $\mu^{(r)} \geq 0$ is a pre-specified parameter. Instead of optimizing $f^{(r)}(\mathbf{x})$, we optimize $\mathbf{g}^{(r)}(\mathbf{x})$. By rearranging terms, the subproblem becomes

$$\mathbf{x}^{(r+1)} = \arg\min_{\mathbf{x}} \left\|\mathbf{x} - \left(\mathbf{x}^{(r)} - \frac{1}{\mu^{(r)}}\nabla f(\mathbf{x}^{(r)})\right)\right\|_2^2 \tag{17}$$

and the solution is

$$\mathbf{x}^{(r+1)} = \mathbf{x}^{(r)} - \frac{1}{\mu^{(r)}}\nabla f(\mathbf{x}^{(r)}) \tag{18}$$

i.e., a gradient step with step-size $1/\mu^{(r)}$, where the gradient is

$$\nabla f^{(r)}(\mathbf{x}^{(r)}) = \mathbf{A}^H\big(\mathbf{W}^{(r)}\big)^2\big(\mathbf{A}\mathbf{x}^{(r)} - \mathbf{y} \odot \mathbf{u}^{(r)}\big) \tag{19}$$

which does not require any matrix inversion operation. Also, the $N \times N$ matrix does not need to be stored, if we take the order $\mathbf{A}\mathbf{x}^{(r)} \rightarrow \big(\mathbf{W}^{(r)}\big)^2\mathbf{A}\mathbf{x}^{(r)} \rightarrow \mathbf{A}^H\big(\mathbf{W}^{(r)}\big)^2\mathbf{A}\mathbf{x}^{(r)}$ to compute the update of $\mathbf{x}$. Therefore, using this approach, both memory and complexity requirements are less demanding. We summarize the algorithm in Algorithm 2. We call this algorithm alternating gradient descent (AltGD).

In terms of convergence, it is easy to see that

$$f^{(r)}(\mathbf{x}) \leq g^{(r)}(\mathbf{x}),$$

if $\mu^{(r)} \geq \lambda_{\max}(\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A})$, and the equality holds if and only if $\mathbf{x} = \mathbf{x}^{(r)}$. Therefore, the algorithmic structure of Algorithm 2 is the same as that of Algorithm 1, except that the majorizing functions of the $\mathbf{x}$-block are different – which means that the proof of convergence of Algorithm 1 also applies here:

Corollary 2.1: If $\mu^{(r)} \geq \lambda_{\max}(\mathbf{A}^T(\mathbf{W}^{(r)})^2\mathbf{A})$ for all r, then, every limit point of Algorithm 1 is a KKT point.

Remark 2.1: Exactly computing $\lambda_{\max}(\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A})$ may be time consuming in practice. Many practically easier ways can be employed, e.g., the Armijo rule-based line search [30]. In our simulations, we use a simple heuristic: we let $\mu^{(r)} = \text{trace}((\mathbf{W}^{(r)})^2)$ instead of $\lambda_{\max}(\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A})$. The rationale behind is that we observe that the energy of $\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A}$ is usually dominated by $\mathbf{W}^{(r)}$, and using $\mu^{(r)} = \text{trace}((\mathbf{W}^{(r)})^2)$ is a good approximation of $\text{trace}(\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A})$ that is an upper bound of $\lambda_{\max}(\mathbf{A}^H(\mathbf{W}^{(r)})^2\mathbf{A})$. We should mention that this step-size choice is a heuristic, but works well in practice, as will be shown in the simulations.

Algorithm 2 AltGD for phase retrieval

```
function $\textsc{AltGD}(\mathbf{y}, \mathbf{A}, \mathbf{x}^{(0)})$
    Initialize $\mathbf{u}^{(0)} = \exp(\angle(\mathbf{A}\mathbf{x}^{(0)}))$
    while stopping criterion has not been reached do
        Choose $\mu^{(r-1)}$ as the leading eigenvalue of $\mathbf{A}^H(\mathbf{W}^{(r-1)})^2\mathbf{A}$ or the trace of $(\mathbf{W}^{(r-1)})^2$
        $\mathbf{x}^{(r)} = \mathbf{x}^{(r-1)} - 1/\mu^{(r-1)}\nabla f(\mathbf{x}^{(r-1)})$
        $\mathbf{u}^{(r)} = \exp(j\angle(\mathbf{A}\mathbf{x}^{(r)}))$
        $r = r + 1$
    end while
end function
```

C. Further Reducing Complexity

1) Extrapolation: Algorithm 2 is easier to compute than Algorithm 1 in terms of per-iteration complexity. However, first-order methods are known to tend to require more iterations in total. One way to alleviate this effect is to incorporate Nesterov's "optimal first-order" method, i.e., for each update, we set

$$\mathbf{z}^{(r)} = \mathbf{x}^{(r)} + \frac{t^{(r-1)} - 1}{t^{(r)}}\left(\mathbf{x}^{(r)} - \mathbf{x}^{(r-1)}\right) \tag{20}$$

$$t^{(r)} = \frac{1 + \sqrt{1 + 4(t^{(r-1)})^2}}{2} \tag{21}$$

$$\mathbf{x}^{(r+1)} = \mathbf{z}^{(r)} - \frac{1}{\mu^{(r)}}\nabla f^{(r)}(\mathbf{z}^{(r)}) \tag{22}$$

In practice, the above 'extrapolation' technique greatly expedites the whole process in various applications [31], [32]. Some numerical evidence can be seen in Fig. 1, where a simple comparison between the plain Algorithm 2 and the extrapolated version is presented. We choose SNR= 20 dB, $N = 16$ and $M = 128$. Each element in the signal and measurement vectors is independently drawn from the complex circularly symmetric Gaussian distribution with mean zero and variance one. The noise is generated from a symmetric α stable (SαS distribution) which will be described in detail in Section IV. Fig. 1 shows the convergence of AltGD and accelerated AltGD using extrapolation when $p = 1.3$. As we can see, the accelerated AltGD converges after 40 iterations which is much faster than AltGD that converges after 200 iterations.

2) Block Incremental / Stochastic Gradient: When N is very large (thus $M > N$ is larger), even gradient computation is too much of a burden. In such cases, a pragmatic way is to "break down" the problem to pieces and do (block) incremental or stochastic gradient. We divide the measurement matrix into L blocks Γ_l. Then it is straightforward to get the gradient for the lth block as

$$\nabla f_{\Gamma_l}(\mathbf{x}^{(r)}) = \mathbf{A}_{\Gamma_l}^H\big(\mathbf{W}_{\Gamma_l}^{(r)}\big)^2\big(\mathbf{A}_{\Gamma_l}\mathbf{x}^{(r)} - \mathbf{y}_{\Gamma_l} \odot \mathbf{u}_{\Gamma_l}^{(r)}\big). \tag{23}$$

The estimate of $\mathbf{x}$ is updated according to

$$\mathbf{x}^{(r+1)} = \mathbf{x}^{(r)} - \frac{1}{\mu_{\Gamma_l}^{(r)}}\nabla f_{\Gamma_l}(\mathbf{x}^{(r)}) \tag{24}$$

where $\mu_{\Gamma_l}^{(r)}$ is chosen as the leading eigenvalue of $\mathbf{A}_{\Gamma_l}^H\mathbf{A}_{\Gamma_l}$ or $\mathrm{trace}\big(\big(\mathbf{W}_{\Gamma_l}^{(r)}\big)^2\big)$. The algorithm can proceed block by block with revisits, or by randomly picking blocks, resulting in block incremental gradient and stochastic gradient versions, respectively.

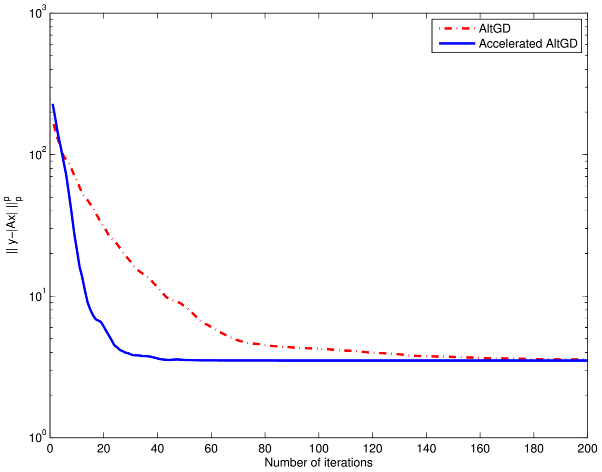


Fig. 1. Cost function value versus number of iterations.

A subtle point here is that, to maintain robustness, one should choose a block size larger than one. The reason is that the robustness of the algorithm is brought by treating different y_m with different weights (more specifically, by downweighting the outliers). When using only one y_m for updating, this ability vanishes.

III. Cramér-Rao Bound Analysis

In this section, the CRB on the accuracy of retrieving $\mathbf{x}$ in (1) is presented. The CRB provides a lower bound on the MSE of unbiased estimators and is expected to be a good predictor of MSE performance for large values of M (i.e., a sufficient number of measurements). For phase retrieval under Gaussian noise, over the past few years, several CRBs have been derived for different models, e.g., 2-D Fourier-based measurements [17], noise added prior to taking the magnitude [18] and noise added after taking the magnitude square [6], [19]-[20]. To the best of our knowledge, there is no available CRB formula for the signal model in (1). Here, we present the CRBs for two particular types of noise: Laplacian and Gaussian noise. Although our main interest here lies in evaluating performance of robust algorithms and the Laplacian CRB serves this purpose, the Gaussian CRB is of interest in other application contexts. Note that we use subscripts *r, c,* L and G to stand for real, complex, Laplacian and Gaussian, respectively.

To get started, we should note that our derivations are based on the assumption that $\mathbf{x}$ is nonzero. That is because the term $|\mathbf{A}\mathbf{x}|$ is non-differentiable at $\mathbf{x} = \mathbf{0}_N$ where $\mathbf{0}_N$ is the $N \times 1$ zero vector, the Fisher information only exists at $\mathbf{x} \neq \mathbf{0}_N$. With this caveat, we have the following theorem:

Theorem 3.1: In Laplacian noise, the variance of any unbiased estimate of $\mathbf{x}$ is bounded below by

$$\mathrm{CRB}_{L,c} = \mathrm{trace}\left(\mathbf{F}_{L,c}^{\dagger}\right) \tag{25}$$

where

$$\mathbf{F}_{L,c} = \frac{4}{\sigma_n^2}\mathbf{G}_{L,c}\,\mathrm{diag}(|\mathbf{A}\mathbf{x}|^{-2})\,\mathbf{G}_{L,c}^{T} \tag{26}$$

with

$$\mathbf{G}_{L,c} = \begin{bmatrix} \mathrm{Re}\{\mathbf{A}^H\mathrm{diag}(\mathbf{A}\mathbf{x})\} \\ \mathrm{Im}\{\mathbf{A}^H\mathrm{diag}(\mathbf{A}\mathbf{x})\} \end{bmatrix}. \tag{27}$$

Proof: See Appendix B. ■

The CRB for real $\mathbf{x}$ is a special case of the complex case, which can be easily derived from Theorem 3.1:

Theorem 3.2: In Laplacian noise, the variance of any unbiased estimate of real-valued $\mathbf{x}$ is bounded below by

$$\mathrm{CRB}_{L,r} = \mathrm{trace}\left(\mathbf{F}_{L,r}^{-1}\right) \tag{28}$$

where

$$\mathbf{F}_{L,r} = \frac{4}{\sigma_n^2}\mathbf{G}_{L,r}\,\mathrm{diag}(|\mathbf{A}\mathbf{x}|^{-2})\,\mathbf{G}_{L,r}^T \tag{29}$$

with

$$\mathbf{G}_{L,r} = \mathrm{Re}\{\mathbf{A}^H\mathrm{diag}(\mathbf{A}\mathbf{x})\}. \tag{30}$$

In Appendix C, we show that when $\mathbf{A}$ has full column rank, $\mathbf{F}_{L,c}$ is singular with rank $(2N-1)$ while $\mathbf{F}_{L,r}$ is always nonsingular. Thus, for complex $\mathbf{x}$, we adopt its pseudo-inverse to compute an optimistic (looser) CRB, which is still a valid lower bound that can be used to benchmark the efficiency of any unbiased estimator [39]-[42]. If $\mathbf{x}$ is close to zero, then the CRB is not tight at low SNRs, and more measurements should be used to approach the CRB.

Theorem 3.3: In Gaussian noise case, the CRB is four times larger than the CRB in Theorem 1, i.e.,

$$\mathrm{CRB}_G = 4\mathrm{CRB}_L. \tag{31}$$

Proof: See Appendix D. ■

In certain applications of phase retrieval, we may be more interested in the performance bound for retrieving the phase of $\mathbf{x}$. The following theorem provides the CRB on both the phase and amplitude of $\mathbf{x}$ under Laplacian noise. Note that in the Gaussian noise case, it is straightforward to apply Theorem 3.3 to compute the lower bound. Similar results can also be found in [43].

Theorem 3.4: In Laplacian noise, the variance of any unbiased estimate of the amplitude of $\mathbf{x}$ is bounded below by

$$\mathrm{CRB}_{L,|\mathbf{x}|} = \sum_{i=1}^{N} d_i \tag{32}$$

and the variance of any unbiased estimate of the phase of $\mathbf{x}$ is bounded below by

$$\mathrm{CRB}_{L,\angle(\mathbf{x})} = \sum_{i=N+1}^{2N} d_i \tag{33}$$

where $d = [d_1 \;\cdots\; d_{2N}]$ contains the main diagonal elements of $\mathbf{F}_L^{\dagger}$ which in this case is defined as

$$\mathbf{F}_L = \frac{4}{\sigma_n^2}\mathbf{G}_L\,\mathrm{diag}(|\mathbf{A}\mathbf{x}|^{-2})\,\mathbf{G}_L^T \tag{34}$$

with

$$\mathbf{G}_L = \begin{bmatrix} \mathrm{diag}(|\mathbf{x}|)^{-1} & \\ & \mathbf{I}_N \end{bmatrix}\begin{bmatrix} \mathrm{Re}\left\{\mathrm{diag}\left(\mathbf{x}^*\right)\mathbf{A}^H\mathrm{diag}(\mathbf{A}\mathbf{x})\right\} \\ \mathrm{Im}\{\mathrm{diag}(\mathbf{x}^*)\mathbf{A}^H\mathrm{diag}(\mathbf{A}\mathbf{x})\} \end{bmatrix}. \tag{35}$$

Proof: See Appendix E. ■

Remark 3.1: In deriving the CRB in Theorem 3.4, we use no additional assumptions on $\mathbf{x}$. Therefore, Theorem 3.4 works for both real and complex $\mathbf{x}$. Specifically, in the real case, the phase is actually the sign of $\mathbf{x}$. Here, $\mathbf{F}_L$ is also singular with rank deficit equal to one, so we adopt its pseudo-inverse to compute the CRB. We omit the proof of rank-1 deficiency of $\mathbf{F}_L$, since it follows the line of argument in Appendix C.

IV. Simulation Results

In this section, we evaluate the performance of the proposed methods and compare them with some existing algorithms, namely, PhaseCut [10], TWF [16], WF [5], GS [2] and PRIME[1] [37] in terms of MSE performance, where the MSE is computed after removing the global phase ambiguity between the true and estimated $\mathbf{x}$. In the simulations, we consider an exponential signal - $\mathbf{x} = \exp(j0.16\pi t)$ comprising 16 samples, i.e., $t = 1, \cdots, 16$. The measurement vectors are generated from a masked Fourier transformation, which has the form of

$$\mathbf{A} = \left[(\mathbf{D}\mathbf{\Lambda}_1)^T \quad \cdots \quad (\mathbf{D}\mathbf{\Lambda}_K)^T \right]^T$$

where $K = M/N$ is the number of masks, $\mathbf{D}$ is a $N \times N$ discrete Fourier transform (DFT) matrix with $\mathbf{D}\mathbf{D}^H = N\mathbf{I}_N$ and $\mathbf{\Lambda}_k$ is a $N \times N$ diagonal masking matrix with its diagonal entries generated by $b_1 b_2$, where b_1 and b_2 are independent and distributed as [8]

$$b_1 = \begin{cases} 1 & \text{w. prob. } 0.25 \\ -1 & \text{w. prob. } 0.25 \\ -j & \text{w. prob. } 0.25 \\ j & \text{w. prob. } 0.25 \end{cases} \quad \text{and } b_2 = \begin{cases} \sqrt{2}/2 & \text{w. prob. } 0.8 \\ \sqrt{3} & \text{w. prob. } 0.2. \end{cases}$$

All results are obtained using a computer with 3.6 GHz i7-4790 CPU and 8 GB RAM. AltGD, AltIRLS, WF, TWF, GS, and PRIME are all initialized from the same starting point that is computed by picking the principal eigenvector of $\sum_{i=1}^{M} y_i^2 \mathbf{a}_i \mathbf{a}_i^H$, and the stopping criterion is

$$\frac{\left| \, ||y - |\mathbf{A}\mathbf{x}^{(r)}|||_2^2 - ||y - |\mathbf{A}\mathbf{x}^{(r-1)}|||_2^2 \, \right|}{||y - |\mathbf{A}\mathbf{x}^{(r-1)}|||_2^2} \leq 10^{-7}$$

or the number of iterations reaching 1000. Furthermore, we use AltGD with extrapolation for the simulations.

A. Selection of p

Before we do the performance comparison, let us study how p affects the performance of the proposed methods. In this example, SNR is fixed at 20 dB, where the SNR is computed via

$$\text{SNR} = 10 \log 10 \left(\frac{||\mathbf{A}\mathbf{x}||^2}{||\mathbf{n}||^2} \right).$$

We assume that 10% of the data are corrupted by outliers that are generated from the Gaussian distribution with mean zero and variance 100, and the remaining elements in $\mathbf{n}$ are zero. Eight masks are employed to generate the measurements. Fig. 2 shows the MSE as a function of p. Note that, when p is smaller than 1, in order to achieve convergence in a non-convex setting, we initialize our methods in a two-step way. Specifically, for each of the proposed methods, when $0.6 \leq p < 1$, we use the spectrum output to initialize our methods with $p = 1.3$ and 100 iterations, and then choose the corresponding output to do another 100 iterations with $p = 1$ and use the output as a final starting point. When $p \leq 0.6$, we take an additional intermediate step with $p = 0.7$ and 100 iterations to gradually stage the initialization process for our approaches, since in this p-regime the subproblem is strictly non-convex. It is observed that generally, $p \leq 1$ provides better performance than $p > 1$, especially for AltIRLS. When $p < 0.5$, AltGD suffers severe performance degradation while AltIRLS still performs well. Our

[1] Here, PRIME stands for PRIME-Modulus-Both-Terms in [37].

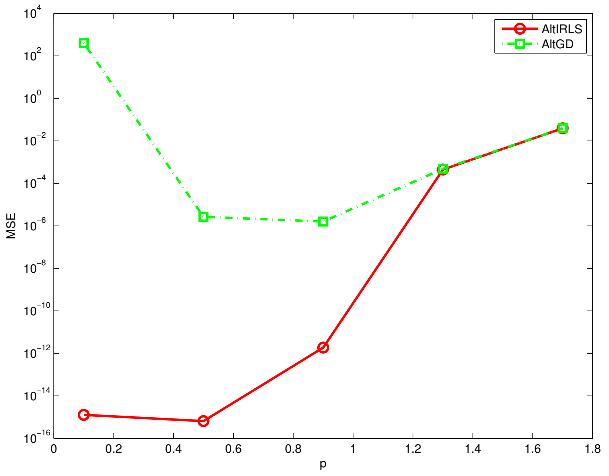


Fig. 2. MSE versus p.

understanding is that first-order algorithms are in general more sensitive to problem conditioning, and a small p can lead to badly-conditioned $\mathbf{W}^{(r)}$. Working with $p < 1$ usually requires much more careful initialization. In our experience, $p \approx 1.3$ strikes a good compromise between numerical stability and estimation accuracy.

To illustrate the performance provided by our proposed methods with $p < 1$ over the state-of-art algorithms including TWF, WF, PhaseCut, GS, and PRIME, we present the following example, where the parameters are the same as for Fig. 2, except that a relative small number of masks is used, say, $K = 6$. Note that among these competitors, PhaseCut, GS and PRIME share the same signal model as ours, but WF is designed after a model with noise added to the squared magnitude of the linear measurements, and TWF adopts a Poisson model with $\lambda = |\mathbf{a}_i^H \mathbf{x}|^2$. Hence, $\mathbf{y}^2$ is fed to WF and TWF, where the squaring is element-wise. We set $p = 0.7$ for the proposed methods, and perform Monte-Carlo trials to estimate MSE $= 10 \log 10(||\hat{\mathbf{x}} - \mathbf{x}||_2^2)$. It is shown in Fig. 3 that the MSE samples of WF, PhaseCut, GS and PRIME are located around -10 dB, and TWF is slightly better than them. However, our methods produce much smaller MSEs. In particular, for AltGD, most of its MSEs are located between -100 dB and -20 dB, while for AltIRLS, its MSEs are concentrated around -145 dB, which indicates that our proposed methods are efficient in suppressing outliers.

B. Statistical Performance Comparison

We now compare the MSE performance as a function of SNR, using 500 Monte-Carlo trials. We use Laplacian and α-stable distributions, which are considered suitable for modeling heavy-tailed distributions that generate impulsive noise. The probability density function (PDF) of the Laplacian distribution is given in (46). The PDF of an α-stable distribution is generally not available, but its characteristic function can be written in closed-form as

$$\phi(t; \alpha, \beta, c, \mu) = \exp\left(jt\mu - \gamma^{\alpha}|t|^{\alpha}\left(1 - j\beta \text{sgn}(t)\Phi(\alpha)\right)\right)$$

where $\Phi(\alpha) = \tan(\alpha\pi/2)$, $0 < \alpha \leq 2$ is the stability parameter, $-1 \leq \beta \leq 1$ is a measure of asymmetry, $\gamma > 0$ is a scale factor which measures the width of the distribution and μ is a shift parameter. There are two special cases that admit closed-form PDF expression, that is, $\alpha = 1$ and $\alpha = 2$ which respectively yield the Cauchy and Gaussian distributions. For $\alpha < 2$, $\phi(t; \alpha, \beta, \gamma, \mu)$ possesses heavy tails, thus is impulsive. Generally, α controls

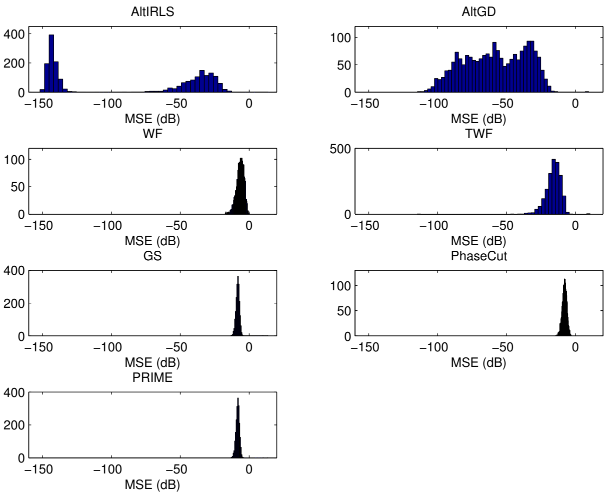


Fig. 3. Signal recovery performance comparison.

the density of impulses. In the following, we set $\alpha = 0.8$, and the other parameters are $\beta = 0$, $\gamma = 2$ and $\mu = 0$, resulting in a symmetric α-stable (SαS) distribution with zero-shift.

In the simulations, the number of measurements is $M = 8N$. Figs. 4 and 5 show the MSEs of amplitude and phase of $\mathbf{x}$ in SαS and Laplacian noise, respectively, where

$$\text{MSE on amplitude} = \frac{1}{500}\sum_{i=1}^{500} ||\, |\hat{\mathbf{x}}|_i - |\mathbf{x}|\, ||_2^2$$

$$\text{MSE on phase} = \frac{1}{500}\sum_{i=1}^{500} ||\, \angle(\hat{\mathbf{x}})_i - \angle(\mathbf{x})\, ||_2^2 .$$

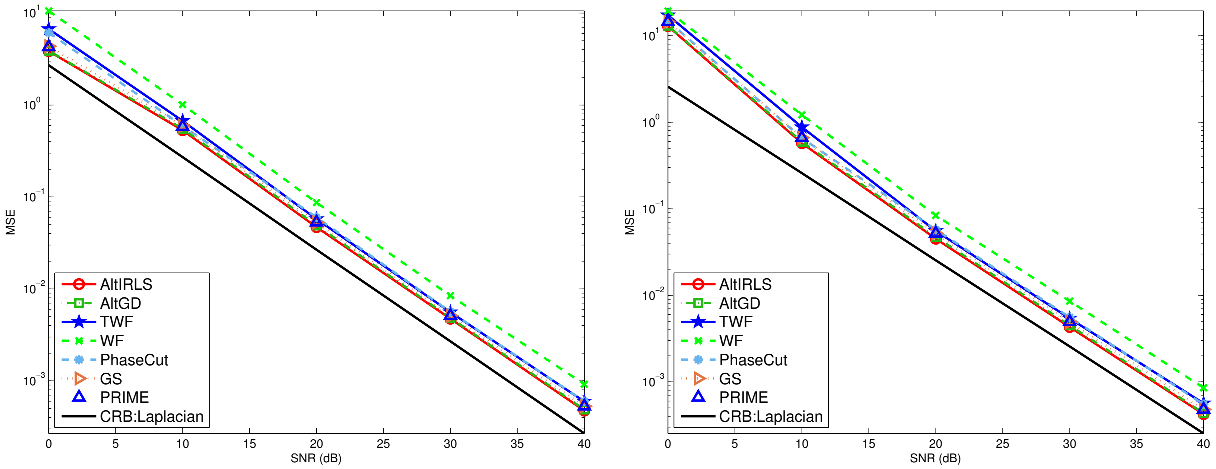
Note that for SαS noise, we set $p = 1.3$ for AltIRLS and AltGD, while for Laplacian noise, $p = 1$. In the Laplacian noise case, we include the CRB derived in Theorem 3.4 as a performance benchmark. It is observed from Fig. 4 that our approaches slightly outperform the AltMinPhase algorithm and yield the MSEs which are closest to the CRB. The performance gap between the proposed methods and their competitors becomes even larger in SαS noise; see Fig. 5. This indicates that the proposed algorithms work better in more critical situations, i.e., when more severe outliers exist. As a reference, we also plot the MSE performance versus SNR in Gaussian noise in Fig. 6, where we set $p = 2$ for having a maximum likelihood estimator. We see that the performance of the algorithms are similar while the proposed algorithms are still the closest to the CRB.

C. CPU Time Comparison

In this example, we compare the CPU times as a function of N. We set SNR at 30 dB, vary the signal length from 8 to 256, and keep the number of masks $K = 8$. We set $p = 1.3$ for the proposed schemes. The simulation results are shown in Fig. 7, from which we find that TWF is faster than our methods. However, it has larger MSE than ours.

D. Performance Comparison on Caffeine Molecule Image Data

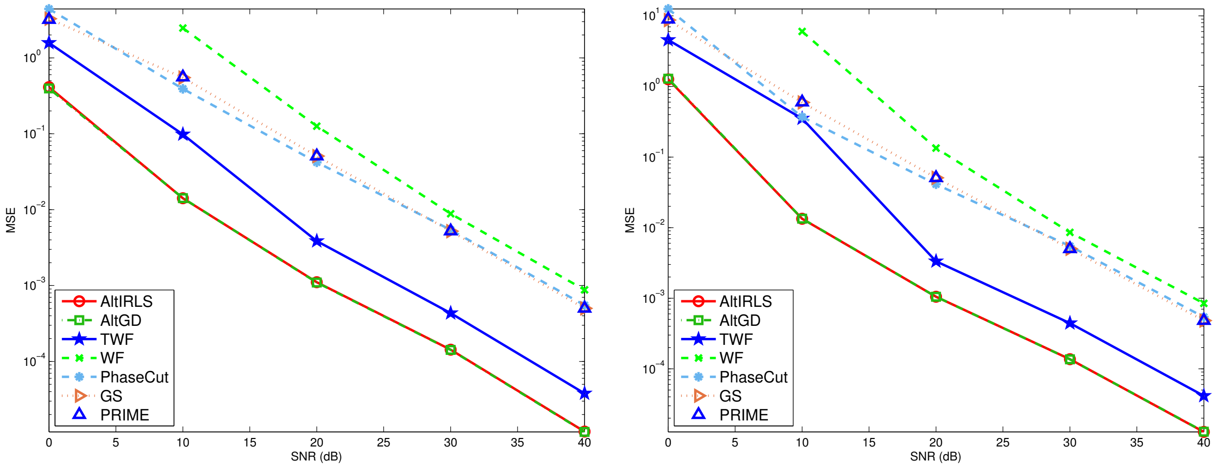
In this subsection, we showcase the performance using real caffeine molecule data that is often used to test phase retrieval algorithms [5], [38], [45]. The caffeine molecule data with size 128×128 is the projection of electron



(a) MSE on amplitude

(b) MSE on phase

Fig. 4. MSE versus SNR in Laplacian noise.

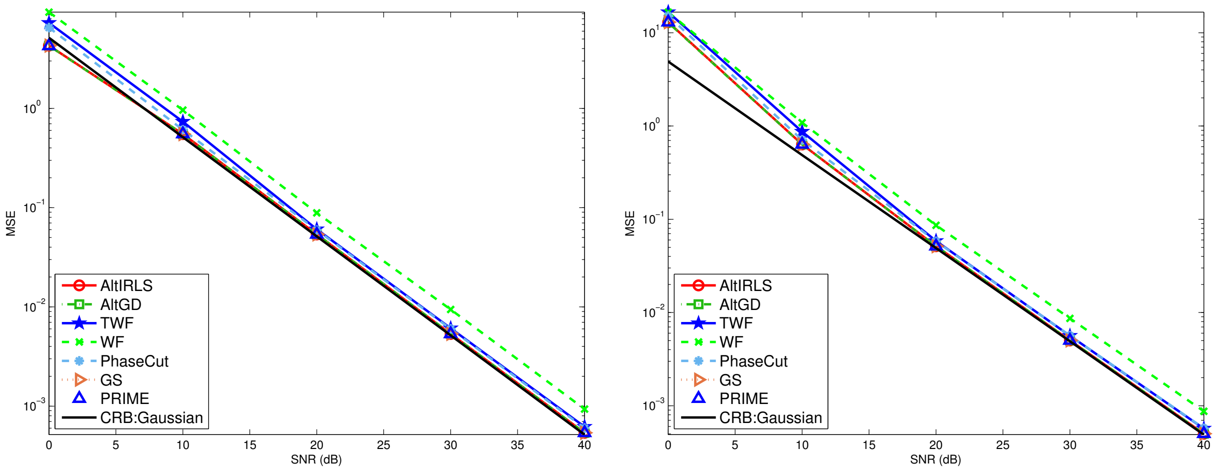


(a) MSE on amplitude

(b) MSE on phase

Fig. 5. MSE versus SNR in SαS noise.

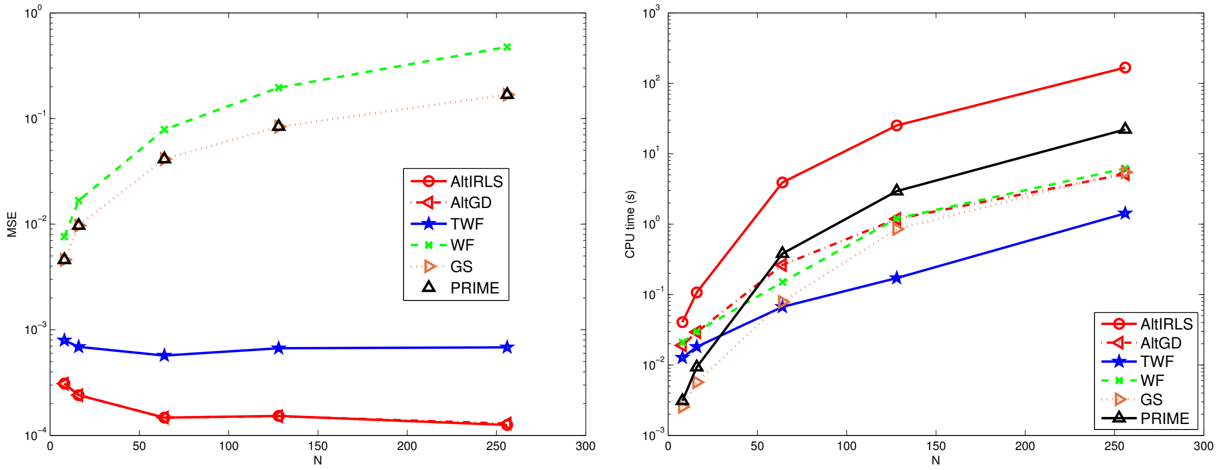
density of a 3-D Caffeine molecule's density map onto the xy-plane. The objective is to reconstruct the data from the magnitude of its masked Fourier transform. Here, we use the same measuring process as in the previous simulations and K = 6 masks are used to generate measurements. We add impulsive noise that is SαS distributed. We compare the performance of AltGD and TWF, which are the most scalable robust algorithms. We also include the block incremental (BI) implementation of AltGD (cf. Section II.C2), which is referred to as *BI-AltGD*. The performance is measured by relative MSE, which is defined as $\text{MSE} = 1/500\sum_{i=1}^{500}\|\hat{\mathbf{X}}_i - \mathbf{X}\|_F^2/\|\mathbf{X}\|_F^2$. Fig. 8 plots the retrieved molecule's density map using AltGD, BI-AltGD with $p = 1.3$ and TWF where SNR$= -5$ dB. It is seen in Fig. 8 that TWF yields a blurred image, while our schemes still work well under such a low SNR. It should be noted that BI-AltGD takes 4.991 seconds and hits a relative MSE as small as 1.945×10^{-8}, that directly knocks out AltGD and TWF, which take 12.605 seconds and 37.391 seconds, resulting in MSE 2.993×10^{-2} and 0.4118, respectively. The reason why TWF takes much longer time than our methods is that it usually requires about 500 iterations to converge, while the AltGD and BI-AltGD only need 100 and 40 (outer) iterations, respectively. Using the same parameter setting but changing SNR from -20 dB to 40 dB, we compare the probability of resolution



(a) MSE on amplitude

(b) MSE on phase

Fig. 6. MSE versus SNR in Gaussian noise.



(a) MSE on amplitude

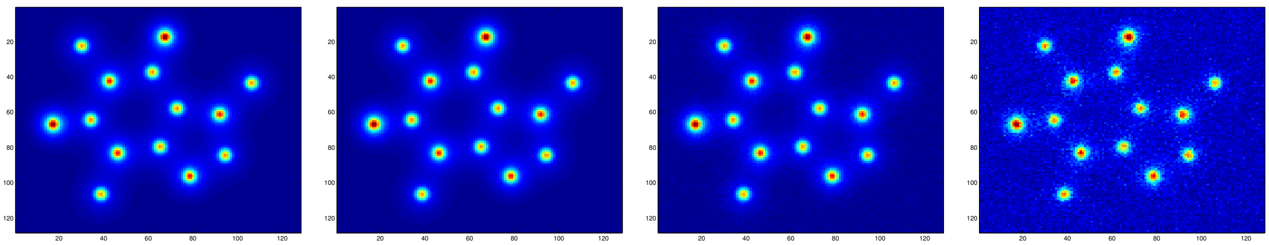
(b) CPU time

Fig. 7. MSE and CPU time versus number of variables N in SαS noise.

as a function of SNR, where resolution is achieved if the relative MSE $\|\hat{\mathbf{X}}_i - \mathbf{X}\|_F^2 / \|\mathbf{X}\|_F^2 < 10^{-5}$. It is shown in Fig. 9 that The BI-AltGD achieves the best performance and is followed by AltGD, while both of them outperform the TWF method in the low SNR regime.

V. Conclusion

In this paper, we considered phase retrieval in the presence of grossly corrupted data – i.e., outliers. We formulated this problem as an ℓ_p fitting problem, where $0 < \ell_p < 2$, and provided an algorithmic framework that is based on two-block inexact alternating optimization. Two algorithms, namely, AltIRLS and AltGD, were proposed under this framework. Although the algorithms cannot be analyzed using standard convergence results for alternating optimization due to a nonconvex consraint, we managed to show that the algorithms converge to a KKT point. The tools used for convergence analysis can also be used for analyzing convergence of other types of algorithms that involve non-convex constraints and inexact alternating optimization. Pertinent CRBs were derived for the noisy



(a) Original

(b) BI-AltGD

(c) AltGD

(d) TWF

Fig. 8. Retrieving molecule image in SαS noise.

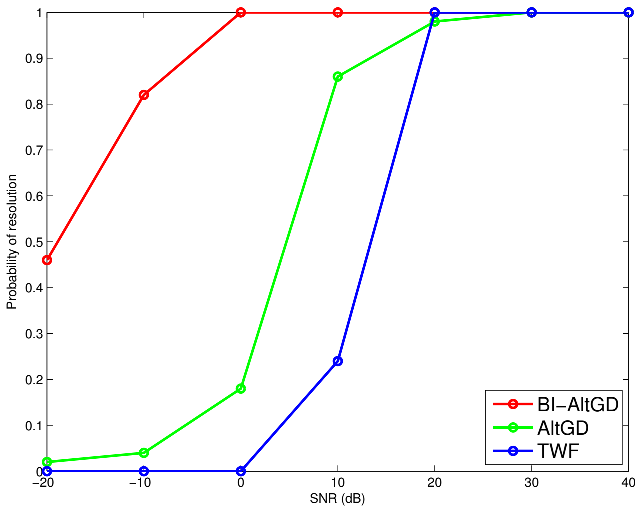


Fig. 9. Probability of resolution versus SNR in SαS noise for the caffeine molecule data.

measurement models considered. Simulations showed that the proposed algorithms are promising in dealing with outliers in the context of phase retrieval.

Appendix A
Proof of Proposition 2.1

Let us denote

$$f(\mathbf{u},\mathbf{x}) = \sum_{m=1}^{M} \left(|y_m u_m - \mathbf{a}_m^H \mathbf{x}|^2 + \epsilon\right)^{p/2}.$$

We also define

$$g(\mathbf{u}^{(r)},\mathbf{x}) = \sum_{m=1}^{M} \left(w_m^{(r)} \left(y_m u_m^{(r)} - \mathbf{a}_m^H \mathbf{x} \right)^2 - \phi_p(w_m^{(r)}) \right).$$

According to Lemma 2.1, it is easily seen that

$$f(\mathbf{u}^{(r)},\mathbf{x}) \leq g(\mathbf{u}^{(r)},\mathbf{x}), \quad \forall \mathbf{x}, \tag{36a}$$

$$f(\mathbf{u}^{(r)},\mathbf{x}^{(r)}) = g(\mathbf{u}^{(r)},\mathbf{x}^{(r)}), \tag{36b}$$

$$\nabla_{\mathbf{x}} f(\mathbf{u}^{(r)},\mathbf{x}^{(r)}) = \nabla_{\mathbf{x}} g(\mathbf{u}^{(r)},\mathbf{x}^{(r)}). \tag{36c}$$

Our update strategy can be therefore summarized as

$$\mathbf{x}^{(r+1)} = \min_{\mathbf{x}} \; g(\mathbf{u}^{(r+1)},\mathbf{x}) \tag{37a}$$

$$\mathbf{u}^{(r+1)} = \min_{|\mathbf{u}|=\mathbf{1}} \ f(\mathbf{u}, \mathbf{x}^{(r+1)}) \tag{37b}$$

Now, we have

$$\begin{aligned} f(\mathbf{u}^{(r)}, \mathbf{x}^{(r)}) &= g\left(\mathbf{u}^{(r)}, \mathbf{x}^{(r)}\right) && \text{(38a)} \\ &\geq g\left(\mathbf{u}^{(r)}, \mathbf{x}^{(r+1)}\right) && \text{(38b)} \\ &\geq f\left(\mathbf{u}^{(r)}, \mathbf{x}^{(r+1)}\right) && \text{(38c)} \\ &\geq f\left(\mathbf{u}^{(r+1)}, \mathbf{x}^{(r+1)}\right) && \text{(38d)} \end{aligned}$$

where (38a) follows (36b), (38b) is obtained because of (37a), (38c) holds since we have the property in (36a), and (38b) is obtained by the fact that the subproblem w.r.t. $\mathbf{u}$ is optimally solved, i.e., (37b).

Assume that $\{r_j\}$ denotes the index set of a convergent subsequence, and that $\{\mathbf{x}^{(r_j)}, \mathbf{u}^{(r_j)}\}$ converges to $(\mathbf{x}^\star, \mathbf{u}^\star)$ Then, we have

$$\begin{aligned} g(\mathbf{x}, \mathbf{u}^{(r_j)}) &\geq g\left(\mathbf{x}^{(r_j+1)}, \mathbf{u}^{(r_j)}\right) && \text{(39a)} \\ &\geq f\left(\mathbf{x}^{(r_j+1)}, \mathbf{u}^{(r_j)}\right) && \text{(39b)} \\ &\geq f\left(\mathbf{x}^{(r_j+1)}, \mathbf{u}^{(r_j+1)}\right) && \text{(39c)} \\ &\geq f\left(\mathbf{x}^{(r_{j+1})}, \mathbf{u}^{(r_{j+1})}\right) && \text{(39d)} \\ &= g\left(\mathbf{x}^{(r_{j+1})}, \mathbf{u}^{(r_{j+1})}\right), && \text{(39e)} \end{aligned}$$

where (39d) is obtained by the fact that $r_{j+1} \geq r_j + 1$ since r_j indexes a subsequence. Taking $j \to \infty$, we see that

$$g(\mathbf{x}, \mathbf{u}^\star) \geq g(\mathbf{x}^\star, \mathbf{u}^\star). \tag{40}$$

The inequality in (40) means that $\mathbf{x}^\star$ is blockwise minimizer of $g(\mathbf{x}, \mathbf{u}^\star)$. Therefore, it satisfies the partial KKT condition w.r.t. $\mathbf{x}$, i.e.,

$$\nabla_{\mathbf{x}} g(\mathbf{x}^\star, \mathbf{u}^\star) = \mathbf{0}. \tag{41}$$

By (36c), we immediately have

$$\nabla_{\mathbf{x}} f(\mathbf{x}^\star, \mathbf{u}^\star) = \mathbf{0}. \tag{42}$$

Similarly, by the update rule in (37b), we have

$$f(\mathbf{x}^{(r_j)}, \mathbf{u}) \geq f(\mathbf{x}^{(r_j)}, \mathbf{u}^{(r_j)}) \tag{43}$$

and thus

$$f(\mathbf{x}^\star, \mathbf{u}) \geq f(\mathbf{x}^\star, \mathbf{u}^\star). \tag{44}$$

Therefore, $\mathbf{u}$ also satisfies the partial KKT condition

$$\nabla_{\mathbf{u}} \ f(\mathbf{x}^\star, \mathbf{u}^\star) + \sum_{m=1}^{M} \lambda_m^\star (|u_m^\star| - 1) = 0, \tag{45}$$

where λ_m for $m = 1, \ldots, M$ are dual variables. Combining (42) and (45), the proof is complete.

Appendix B
CRB for Laplacian Noise

The likelihood function for Laplacian noise is given by [43]-[44]

$$p(\mathbf{y};\mathbf{x}) = \prod_{i=1}^{M} \frac{1}{\sigma_n^2} \exp\left\{ -\frac{2}{\sigma_n} \left| y_i - |\mathbf{a}_i^H \mathbf{x}| \right| \right\} \tag{46}$$

with its log-likelihood function being

$$\ln p(\mathbf{y};\mathbf{x}) = -M \ln(\sigma_n^2) - \frac{2}{\sigma_n} \sum_{i=1}^{M} \left| y_i - |\mathbf{a}_i^H \mathbf{x}| \right|. \tag{47}$$

The vector of unknown parameters for complex-valued $\mathbf{x}$ is

$$\boldsymbol{\beta} = [\text{Re}\{x_1\} \ \cdots \ \text{Re}\{x_N\}, \ \text{Im}\{x_1\} \ \cdots \ \text{Im}\{x_N\}]^T. \tag{48}$$

It is worth noting that σ_n^2 is actually an unknown parameter which should be considered as a part of $\boldsymbol{\beta}$. However, since σ_n^2 is uncorrelated with the real and imaginary parts of x_i, their mutual Fisher information is zero. It will not impact the final CRB formula for $\mathbf{x}$. For this reason, we do not include σ_n^2 in $\boldsymbol{\beta}$. Thus, the FIM can be partitioned into four parts, i.e.,

$$\mathbf{F}_{L,c} = \begin{bmatrix} \mathbf{F}_{L,rr} & \mathbf{F}_{L,ri} \\ \mathbf{F}_{L,ir} & \mathbf{F}_{L,ii} \end{bmatrix} \tag{49}$$

where

$$[\mathbf{F}_{L,c}]_{m,n} = E\left[\frac{\partial \ln p(\mathbf{y};\mathbf{x})}{\partial \boldsymbol{\beta}_m} \frac{\partial \ln p(\mathbf{y};\mathbf{x})}{\partial \boldsymbol{\beta}_n} \right]. \tag{50}$$

The partial derivative of $\ln p(\mathbf{y};\mathbf{x})$ with respective to $\boldsymbol{\beta}_m$ is

$$\begin{aligned} \frac{\partial \ln p(\mathbf{y};\mathbf{x})}{\partial \boldsymbol{\beta}_m} &= -\frac{2}{\sigma_n} \sum_{i=1}^{M} \frac{\partial \left| y_i - |\mathbf{a}_i^H \mathbf{x}| \right|}{\partial \boldsymbol{\beta}_m} \\ &= \frac{2}{\sigma_n} \sum_{i=1}^{M} \frac{y_i - |\mathbf{a}_i^H \mathbf{x}|}{\left| y_i - |\mathbf{a}_i^H \mathbf{x}| \right|} \frac{\partial |\mathbf{a}_i^H \mathbf{x}|}{\partial \boldsymbol{\beta}_m} \\ &= \frac{2}{\sigma_n} \sum_{i=1}^{M} \text{sgn}(y_i - |\mathbf{a}_i^H \mathbf{x}|) \frac{\partial |\mathbf{a}_i^H \mathbf{x}|}{\partial \boldsymbol{\beta}_m} \end{aligned} \tag{51}$$

where

$$\text{sgn}(a) = \begin{cases} 1, & a > 0 \\ -1, & a < 0 \end{cases} \tag{52}$$

and

$$\frac{\partial |\mathbf{a}_i^H \mathbf{x}|}{\partial \boldsymbol{\beta}_m} = \begin{cases} \dfrac{[\text{Re}\left\{ \mathbf{a}_i \mathbf{a}_i^H \mathbf{x} \right\}]_m}{|\mathbf{a}_i^H \mathbf{x}|}, & \text{for } \boldsymbol{\beta}_m = \text{Re}\{x_m\} \\ \dfrac{[\text{Im}\left\{ \mathbf{a}_i \mathbf{a}_i^H \mathbf{x} \right\}]_m}{|\mathbf{a}_i^H \mathbf{x}|}, & \text{for } \boldsymbol{\beta}_m = \text{Im}\{x_m\}. \end{cases} \tag{53}$$

Substituting (51) into (50), we have

$$[\mathbf{F}_{L,c}]_{m,n} = \frac{4}{\sigma_n^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \frac{\partial |\mathbf{a}_i^H \mathbf{x}|}{\partial \boldsymbol{\beta}_m} \frac{\partial |\mathbf{a}_i^H \mathbf{x}|}{\partial \boldsymbol{\beta}_n}$$

$$\times E\left[\text{sgn}(y_i - |\mathbf{a}_i^H \mathbf{x}|) \cdot \text{sgn}(y_j - |\mathbf{a}_j^H \mathbf{x}|)\right]. \tag{54}$$

Next we compute the value of $E[\text{sgn}(y_i - |\mathbf{a}_i^H \mathbf{x}|)\text{sgn}(y_j - |\mathbf{a}_j^H \mathbf{x}|)]$. For notational simplicity, let $s_i = \text{sgn}(y_i - |\mathbf{a}_i^H \mathbf{x}|)$. It is obvious that when $i = j$, we have

$$E[s_i s_j] = 1. \tag{55}$$

For $i \neq j$, we first write

$$\begin{aligned} E[s_i s_j] &= \Pr(s_i = s_j) \times (+1) + \Pr(s_i \neq s_j) \times (-1) \\ &= 2\Pr(s_i = s_j) - 1 \end{aligned} \tag{56}$$

where $\Pr$ standards for the probability. Then the value of $\Pr(s_i = s_j)$ is computed as

$$\begin{aligned} \Pr(s_i = s_j) &= \Pr(s_i = 1 | s_j = 1)\Pr(s_j = 1) \\ &\quad + \Pr(s_i = -1 | s_j = -1)\Pr(s_j = -1) \\ &= \Pr(s_i = 1)\Pr(s_j = 1) \\ &\quad + \Pr(s_i = -1)\Pr(s_j = -1) \\ &= 0.5 \end{aligned} \tag{57}$$

where $\Pr(s_i = 1) = \Pr(s_i = -1) = 0.5$ and the second equation follows by independence of s_i and s_j when $i \neq j$. Substituting (57) into (56) and using (55) yields

$$E[s_i s_j] = \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases} \tag{58}$$

Substituting (58) into (54), we obtain

$$[\mathbf{F}_{L,c}]_{m,n} = \frac{4}{\sigma_n^2} \sum_{i=1}^{M} \frac{\partial |\mathbf{a}_i^H \mathbf{x}|}{\partial \boldsymbol{\beta}_m} \frac{\partial |\mathbf{a}_i^H \mathbf{x}|}{\partial \boldsymbol{\beta}_n}. \tag{59}$$

Now using (53) and (59), the four sub-FIMs can be easily derived as

$$\begin{aligned} \mathbf{F}_{L,rr} &= \frac{4}{\sigma_n^2} \text{Re}\{\mathbf{A}^H \text{diag}(\mathbf{A}\mathbf{x})\} \cdot \text{diag}(|\mathbf{A}\mathbf{x}|^{-2}) \\ &\quad \times \text{Re}\{\mathbf{A}^H \text{diag}(\mathbf{A}\mathbf{x})\}^T \end{aligned} \tag{60}$$

$$\begin{aligned} \mathbf{F}_{L,ii} &= \frac{4}{\sigma_n^2} \text{Im}\{\mathbf{A}^H \text{diag}(\mathbf{A}\mathbf{x})\} \cdot \text{diag}(|\mathbf{A}\mathbf{x}|^{-2}) \\ &\quad \times \text{Im}\{\mathbf{A}^H \text{diag}(\mathbf{A}\mathbf{x})\}^T \end{aligned} \tag{61}$$

$$\begin{aligned} \mathbf{F}_{L,ri} &= \frac{4}{\sigma_n^2} \text{Re}\{\mathbf{A}^H \text{diag}(\mathbf{A}\mathbf{x})\} \cdot \text{diag}(|\mathbf{A}\mathbf{x}|^{-2}) \\ &\quad \times \text{Im}\{\mathbf{A}^H \text{diag}(\mathbf{A}\mathbf{x})\}^T \end{aligned} \tag{62}$$

$$\mathbf{F}_{L,ir} = \mathbf{F}_{L,ri}^T. \tag{63}$$

Combining (60)-(63), we obtain

$$\mathbf{F}_{L,c} = \frac{4}{\sigma_n^2} \mathbf{G}_{L,c} \, \text{diag}(|\mathbf{A}\mathbf{x}|^{-2}) \, \mathbf{G}_{L,c}^T \tag{64}$$

where

$$\mathbf{G}_{L,c} = \begin{bmatrix} \text{Re}\{\mathbf{A}^H \text{diag}(\mathbf{A}\mathbf{x})\} \\ \text{Im}\{\mathbf{A}^H \text{diag}(\mathbf{A}\mathbf{x})\} \end{bmatrix}. \tag{65}$$

This completes the proof of Theorem 1.

APPENDIX C
PROOF OF RANK PROPERTY OF $\mathbf{F}_{L,c}$ AND $\mathbf{F}_{L,r}$

We note here that $\mathbf{F}_{L,c}$ is derived under the assumption of nonzero $\mathbf{x}$. If $\mathbf{A}$ has full column rank, we have $\text{diag}(|\mathbf{A}\mathbf{x}|^{-2})$ is full rank. As a result, computing the rank of $\mathbf{F}_{L,c}$ is equivalent to computing the rank of $\mathbf{G}_{L,c}$. To this end, define a nonzero vector $\mathbf{v} = [\ \mathbf{v}_1^T\ \mathbf{v}_2^T\ ]^T \in \mathbf{R}^{2N}$, which leads to

$$\mathbf{G}_{L,c}^T \mathbf{v} = \text{Re}\{\mathbf{A}^H \text{diag}(\mathbf{A}\mathbf{x})\}^T \mathbf{v}_1 + \text{Im}\{\mathbf{A}^H \text{diag}(\mathbf{A}\mathbf{x})\}^T \mathbf{v}_2. \tag{66}$$

Now let $\mathbf{u} = \mathbf{v}_1 + j\mathbf{v}_2$, then

$$\begin{aligned} \mathbf{G}_{L,c}^T \mathbf{v} &= \text{Re}\left\{ \left(\mathbf{A}^H \text{diag}(\mathbf{A}\mathbf{x})\right)^H \mathbf{u} \right\} \\ &= \text{Re}\left\{ (\mathbf{A}\mathbf{x})^* \odot (\mathbf{A}\mathbf{u}) \right\} \end{aligned} \tag{67}$$

which equals to zero if and only if

$$\mathbf{u} = j\mathbf{x} \tag{68}$$

i.e.,

$$\mathbf{G}_{L,c}^T \mathbf{v} = \text{Re}\left\{ j\, |\mathbf{A}\mathbf{x}|^2 \right\} = \mathbf{0}. \tag{69}$$

This means that there is only one direction $\mathbf{v} = [\ -\text{Im}\{\mathbf{x}\}^T\ \text{Re}\{\mathbf{x}\}^T\ ]^T$, which is non-zero, lies in the null space of $\mathbf{G}_{L,c}$, thus also in the null space of $\mathbf{F}_{L,c}$.

In the real $\mathbf{x}$ case, similar to the proof of $\mathbf{F}_{L,c}$, it suffices to show that there is no nonzero vector $\mathbf{v} \in \mathbb{R}^N$ making $\mathbf{F}_{L,r}\mathbf{v} = \mathbf{0}$. It is easy to see that

$$\mathbf{G}_{L,r}^T \mathbf{v} = \text{Re}\left\{ (\mathbf{A}\mathbf{x})^* \odot (\mathbf{A}\mathbf{v}) \right\}. \tag{70}$$

Since $\mathbf{v}$ is real-valued, it cannot be set to $\mathbf{v} = j\mathbf{x}$ to make (70) zero. Given a nontrivial $\mathbf{A}$, it is impossible to find a $\mathbf{v}$ such that $\mathbf{A}\mathbf{v} = \mathbf{0}$ almost surely. Therefore, $\mathbf{F}_{L,r}$ is full rank almost surely. This completes the proof.

APPENDIX D
PROOF OF THEOREM 3.1

Before we do the proof, we note that, the subscripts 'r' and 'c' are omitted in the following derivations, because the proof is true for both real and complex cases. Assuming that the noise is Gaussian, the log-likelihood function can be written as

$$\ln p(\mathbf{y}; \mathbf{x}) = -\frac{M}{2} \ln(2\pi\sigma_n^2) - \frac{1}{2\sigma_n^2} \sum_{i=1}^{M} (y_i - |\mathbf{a}_i^H \mathbf{x}|)^2. \tag{71}$$

Unlike the Laplacian case, it is easier to take the second-order derivative of (71) to compute the FIM. The (m, n) entry of the FIM is given by

$$[\mathbf{F}_G]_{m,n} = -\mathbb{E}\left[\frac{\partial^2 \ln p(\mathbf{y};\mathbf{x})}{\partial\boldsymbol{\beta}_m \partial\boldsymbol{\beta}_n}\right] \tag{72}$$

where

$$\frac{\partial^2 \ln p(\mathbf{y};\mathbf{x})}{\partial\boldsymbol{\beta}_m \partial\boldsymbol{\beta}_n} = \frac{1}{\sigma_n^2}\sum_{i=1}^{M}\Bigg((y_i - |\mathbf{a}_i^H\mathbf{x}|)\frac{\partial^2|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_m\partial\boldsymbol{\beta}_n} \\ -\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_m}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_n}\Bigg). \tag{73}$$

Taking the expectation of both sides of (73) yields

$$\mathbb{E}\left[\frac{\partial^2 \ln p(\mathbf{y};\mathbf{x})}{\partial\boldsymbol{\beta}_m \partial\boldsymbol{\beta}_n}\right] = -\frac{1}{\sigma_n^2}\sum_{i=1}^{M}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_m}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_n} \tag{74}$$

where $\mathbb{E}[y_i - |\mathbf{a}_i^H\mathbf{x}|] = 0$. Hence,

$$[\mathbf{F}_G]_{m,n} = \frac{1}{\sigma_n^2}\sum_{i=1}^{M}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_m}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\boldsymbol{\beta}_n}. \tag{75}$$

By comparing (75) with (59), we find that the Fisher information for Gaussian noise is $1/4$ the one for Laplacian nosie, which proves the claim in Theorem 3.1.

Appendix E
Proof of Theorem 3.4

In this case, the unknown parameter vector $\boldsymbol{\beta}$ contains the amplitude and phase of $\mathbf{x}$, that is

$$\boldsymbol{\beta} = \begin{bmatrix} |\mathbf{x}^T|, \angle(\mathbf{x}^T) \end{bmatrix}^T. \tag{76}$$

Using (59), we have

$$\mathbf{F}_L = \begin{bmatrix} \mathbf{F}_{L,|\mathbf{x}||\mathbf{x}|} & \mathbf{F}_{L,|\mathbf{x}|\angle(\mathbf{x})} \\ \mathbf{F}_{L,\angle(\mathbf{x})|\mathbf{x}|} & \mathbf{F}_{L,\angle(\mathbf{x})\angle(\mathbf{x})} \end{bmatrix} \tag{77}$$

where

$$\mathbf{F}_{L,|\mathbf{x}||\mathbf{x}|} = \frac{4}{\sigma_n^2}\sum_{i=1}^{M}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial|\mathbf{x}|}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial|\mathbf{x}^T|} \tag{78}$$

$$\mathbf{F}_{L,\angle(\mathbf{x})\angle(\mathbf{x})} = \frac{4}{\sigma_n^2}\sum_{i=1}^{M}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\angle(\mathbf{x})}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\angle(\mathbf{x}^T)} \tag{79}$$

$$\mathbf{F}_{L,|\mathbf{x}|\angle(\mathbf{x}^T)} = \frac{4}{\sigma_n^2}\sum_{i=1}^{M}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial|\mathbf{x}|}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\angle(\mathbf{x}^T)} \tag{80}$$

$$\mathbf{F}_{L,\angle(\mathbf{x})|\mathbf{x}|} = \mathbf{F}_{L,|\mathbf{x}|\angle(\mathbf{x}^T)}^T. \tag{81}$$

The computation of $\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial|\mathbf{x}|}$ is

$$\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial|\mathbf{x}|} = \frac{1}{|\mathbf{a}_i^H\mathbf{x}|}\Bigg[\mathrm{Re}\{\mathbf{a}_i^H\mathbf{x}\}\frac{\partial\mathrm{Re}\{\mathbf{a}_i^H\mathbf{x}\}}{\partial|\mathbf{x}|}$$

$$+ \operatorname{Im}\{\mathbf{a}_i^H\mathbf{x}\}\frac{\partial \operatorname{Im}\{\mathbf{a}_i^H\mathbf{x}\}}{\partial|\mathbf{x}|}\Bigg] \tag{82}$$

where

$$\frac{\partial \operatorname{Re}\{\mathbf{a}_i^H\mathbf{x}\}}{\partial|\mathbf{x}|} = \operatorname{Re}\left\{\mathbf{a}_i^* \odot \frac{\mathbf{x}}{|\mathbf{x}|}\right\} \tag{83}$$

$$\frac{\partial \operatorname{Im}\{\mathbf{a}_i^H\mathbf{x}\}}{\partial|\mathbf{x}|} = \operatorname{Im}\left\{\mathbf{a}_i^* \odot \frac{\mathbf{x}}{|\mathbf{x}|}\right\}. \tag{84}$$

$$\tag{85}$$

Substituting (83) and (84) into (82), we obtain

$$\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial|\mathbf{x}|} = \frac{1}{|\mathbf{a}_i^H\mathbf{x}|}\operatorname{Re}\left\{\mathbf{a}_i^H\mathbf{x}\left(\mathbf{a}_i \odot \frac{\mathbf{x}^*}{|\mathbf{x}|}\right)\right\}. \tag{86}$$

One the other hand, the derivative of $|\mathbf{a}_i^H\mathbf{x}|$ w.r.t. $\angle(\mathbf{x})$ is

$$\begin{aligned}\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\angle(\mathbf{x})} = \frac{1}{|\mathbf{a}_i^H\mathbf{x}|}\Bigg[&\operatorname{Re}\{\mathbf{a}_i^H\mathbf{x}\}\frac{\partial \operatorname{Re}\{\mathbf{a}_i^H\mathbf{x}\}}{\partial\angle(\mathbf{x})} \\ &+ \operatorname{Im}\{\mathbf{a}_i^H\mathbf{x}\}\frac{\partial \operatorname{Im}\{\mathbf{a}_i^H\mathbf{x}\}}{\partial\angle(\mathbf{x})}\Bigg]\end{aligned} \tag{87}$$

where

$$\frac{\partial \operatorname{Re}\{\mathbf{a}_i^H\mathbf{x}\}}{\partial\angle(\mathbf{x})} = \operatorname{Im}\left\{\mathbf{a}_i \odot \mathbf{x}^*\right\} \tag{88}$$

$$\frac{\partial \operatorname{Im}\{\mathbf{a}_i^H\mathbf{x}\}}{\partial\angle(\mathbf{x})} = \operatorname{Re}\left\{\mathbf{a}_i \odot \mathbf{x}^*\right\}. \tag{89}$$

$$\tag{90}$$

Plugging (88) and (89) into (87) yields

$$\frac{\partial|\mathbf{a}_i^H\mathbf{x}|}{\partial\angle(\mathbf{x})} = \frac{1}{|\mathbf{a}_i^H\mathbf{x}|}\operatorname{Im}\left\{\mathbf{a}_i^H\mathbf{x}\left(\mathbf{a}_i \odot \mathbf{x}^*\right)\right\}. \tag{91}$$

Using (78)-(81), (86) and (91), and defining

$$\mathbf{G}_L = \begin{bmatrix}\operatorname{diag}(|\mathbf{x}|^{-1}) & \\ & \mathbf{I}_N\end{bmatrix}\begin{bmatrix}\operatorname{Re}\left\{\operatorname{diag}\left(\mathbf{x}^*\right)\mathbf{A}^H\operatorname{diag}(\mathbf{A}\mathbf{x})\right\} \\ \operatorname{Im}\{\operatorname{diag}(\mathbf{x}^*)\mathbf{A}^H\operatorname{diag}(\mathbf{A}\mathbf{x})\}\end{bmatrix}. \tag{92}$$

we finish the proof of Theorem 3.

References

[1] C. Qian, X. Fu, N.D. Sidiropoulos, and L. Huang, “Inexact Alternating Optimization for Phase Retrieval with Outliers,” submitted to *EUSIPCO 2016*, Aug. 29 – Sep. 2, 2016, Budapest, Hungary.

[2] R. Gerchberg and W. Saxton, “A practical algorithm for the determination of phase from image and diffraction plane pictures,” *Optik*, vol. 35, pp. 237-246, 1972.

[3] J. R. Fienup, “Phase retrieval algorithms: A comparison,” *Applied Optics*, vol. 21, no. 15, pp. 2758-2769, 1982.

[4] P. Netrapalli, P. Jain and S. Sanghavi, “Phase retrieval using alternating minimization,” *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp 4814-4826. 2015.

[5] E. J. Candès, X. Li and M. Soltanolkotabi, “Phase retrieval via Wirtinger Flow: Theory and algorithms,” *IEEE Trans. Information Theory*, vol. 61, no. 4, pp. 1985-2007, 2015.

[6] C. Qian, N. D. Sidiropoulos, K. Huang, L. Huang and H. C. So, “Phase retrieval using feasible point pursuit: Algorithms and Cramér-Rao bound,” *IEEE Trans. Signal Process.*, submitted, 2015.

[7] O. Mehanna, K. Huang, B. Gopalakrishnan, A. Konar and N. S. Sidiropoulos, “Feasible point pursuit and successive approximation of non-convex QCQPs,” *IEEE Signal Process. Letters*, vol. 22, no. 7, pp. 804-808, 2015.

[8] E. J. Candès, T. Strohmer, and V. Voroninski. “PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming,” *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1241-1274, 2013.

[9] E. J. Candès, Y. C. Eldar, T. Strohmer and V. Voroninski, “Phase retrieval via matrix completion,” *SIAM Review*, vol. 57, no. 2 pp. 225-251, 2015.

[10] I. Waldspurger, A. d’Aspremont, and S. Mallat, “Phase recovery, maxcut and complex semidefinite programming,” *Mathematical Programming*, vol. 149, no. 1-2, pp. 47-81, 2015.

[11] J. Sigl, “Nonlinear residual minimization by iteratively reweighted least squares,” *arXiv preprint arXiv*, 1504.06815, 2015.

[12] T. T. Cai, X. Li and Z. Ma, “Optimal rates of convergence of noisy sparse phase retrieval via thresholded wirtinger flow,” *arXiv preprint arXiv*, 1504.03385, 2015.

[13] D. S. Weller, A. Pnueli, O. Radzyner, G. Divon, Y. C. Eldar and J. A. Fessler, “Phase retrieval of sparse signals using optimization transfer and ADMM,” *Proc. of 2014 Internat. Conf. on Image Process. (ICIP)*, pp. 1342-1346, Paris, 2014.

[14] D. S. Weller, A. Pnueli, G. Divon, O. Radzyner, Y. C. Eldar and J. A. Fessler, “Undersampled phase retrieval with outliers,” *IEEE Trans. Computational Imaging*, vol. 1, no. 4, pp. 247-258, 2015.

[15] H. M. L. Faulkner and J. M. Rodenburg, “Movable aperture lensless transmission microscopy: A novel phase retrieval algorithm,” *Physical review letters*, vol. 93, no. 2, pp. 023903, 2004

[16] Y. Chen and E. Candés, “Solving random quadratic systems of equations is nearly as easy as solving linear systems,” *Advances in Neural Information Processing System*, pp. 739-747, 2015.

[17] J. N. Cederquist and C. C. Wackerman, “Phase-retrieval error: A lower bound,” *Journal of the Optical Society of America A*, vol. 4, no. 9 pp. 1788-1792, 1987.

[18] R. Balan, “The Fisher information matrix and the CRLB in a non-AWGN model for the phase retrieval problem,” *Proc. of 2015 Internat. Conf. on Sampl. Theory and Applications (SampTA)*, pp. 178-182, Washington, DC, 2015.

[19] R. Balan, “Reconstruction of signals from magnitudes of redundant representations,” *arXiv preprint arXiv*, 1207.1134, 2012.

[20] R. Balan, “Reconstruction of signals from magnitudes of redundant representations: The complex case,” *Foundations of Computational Mathematics*, pp. 1-45, 2013.

[21] A. S. Bandeira, J. Cahill, D. G. Mixon and A. A. Nelson, “Saving phase: Injectivity and stability for phase retrieval,” *Applied and Computational Harmonic Analysis*, vol. 37, no. 1, pp. 106-125, 2014.

[22] Ya-Feng Liu, Shiqian Ma, Yu-Hong Dai, and Shuzhong Zhang, “A smoothing SQP framework for a class of composite ℓ_q minimization over polyhedron,” *Mathematical Programming*, pp. 1-35, 2015.

[23] X. Fu, W.-K. Ma, K. Huang, and N. D. Sidiropoulos, “Robust volume minimization-based structured matrix factorization via alternating optimization,” to appear in *Proceeding of IEEE Internat. Conf. Acoust., Speech and Signal Process. (ICASSP)*, Shanghai, 2016.

[24] Ya-Feng Liu, Yu-Hong Dai, and Shiqian Ma, “Joint power and admission control: Non-convex ℓ_q approximation and an effective polynomial time deflation approach,” *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3641-3656, 2015.

[25] R. Chartrand and Wotao Yin, “Iteratively reweighted algorithms for compressive sensing,” *Proceeding of IEEE Internat. Conf. Acoust., Speech and Signal Process.*, Las Vegas, NV, pp. 3869-3872, 2008.

[26] W. J. Zeng, H. C. So and L. Huang, “ℓ_p-MUSIC: Robust direction-of-arrival estimator for impulsive noise environments,” *IEEE Trans. Signal Process.*, vol. 61, no. 17, pp. 4296-4308, 2013.

[27] X. Fu, K. Huang, W.-K. Ma, N. D. Sidiropoulos, and R. Bro, “Joint tensor factorization and outlying slab suppression with applications,” *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6315-6328, 2015.

[28] P. W. Holland and R. E. Welsch, “Robust regression using iteratively reweighted least-squares,” *Communications in Statistics-theory and Methods*, vol. 6, no. 9, pp. 813-827, 1977.

[29] I. Daubechies, R. DeVore, M. Fornasier and C. S. G unt urk, “Iteratively reweighted least squares minimization for sparse recovery,” *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1-38, 2010.

[30] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM journal on imaging sciences*, vol.2, no.1, pp. 183–202, 2009.

[31] X. Fu, K. Huang, B. Yang, W.-K. Ma and N.D. Sidiropoulos, “Robust volume minimization-based matrix factorization for remote sensing and document clustering,” *IEEE Trans. Signal Process.*, submitted, 2016.

[32] Y. Xu, R. Hao, W. Yin, and Z. Su, “Parallel matrix factorization for low-rank tensor completion,” *arXiv preprint arXiv*:1312.1254, 2013.

[33] Y. Nesterov, “A method for unconstrained convex minimization problem with the rate of convergence O $(1/k2)$,” *Doklady an SSSR*, vol. 269, pp. 543?547, 1983.

[34] O. Fercoq and P. Richtàrik, “Accelerated, parallel, and proximal coordinate descent,” *SIAM Journal on Optimization*, vol. 25, no. 4, pp. 1997-2023, 2015.

[35] Y. Nesterov, “Efficiency of coordinate descent methods on huge-scale optimization problems,” *SIAM Journal on Optimization*, vol. 22, no. 2 pp. 341-362, 2012.

[36] J. Song, P. Babu, and D. Palomar, “Optimization methods for designing sequences with low autocorrelation sidelobes,” *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 3998-4009, 2015.

[37] T. Qiu, P. Babu and D. P. Palomar, “PRIME: Phase retrieval via majorization-minimization,” *arXiv preprint arXiv*, 1511.01669, 2015.

[38] K. Wei, “Phase retrieval via Kaczmarz methods,” *arXiv preprint arXiv*, 1502.01822, 2015.

[39] P. Stoica and T. L. Marzetta, “Parameter estimation problems with singular information matrices,” *IEEE Trans. Signal Process.*, vol. 49, no. 1, pp. 87-90, 2001.

[40] A. O. Hero, III, J. A. Fessler and M. Usman, “Exploring estimator bias-variance tradeoffs using the uniform CR bound,” *IEEE Trans. Signal Process.*, vol. 44, pp. 2026-2041, Aug. 1996

[41] C. R. Rao, *Linear statistical inference and its applications*, 2nd ed. New York: Wiley, 1973.

[42] K. Huang and N. D. Sidiropoulos, “Putting nonnegative matrix factorization to the test: A tutorial derivation of pertinent Cramér-Rao bounds and performance benchmarking,” *IEEE Signal Processing Magazine, Special Issue on Source Separation and Applications*, vol. 31, no. 3, pp. 76-86, 2014.

[43] S. A. Vorobyov, Y. Rong, N. D. Sidiropoulos and A. B. Gershman, “Robust iterative fitting of multilinear models,” *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2678-2689, 2005.

[44] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*. Upper Saddle River, NJ: Prentice-Hall, 1998.

[45] F. Fogel, I. Waldspurger, and A. d’Aspremont, “Phase retrieval for imaging problems,” *arXiv preprint arXiv*, 1304.7735, 2013.