




# Super Resolution Phase Retrieval for Sparse Signals

Gilles Baechler , *Student Member, IEEE*, Miranda Kreković , *Student Member, IEEE*, Juri Ranieri, Amina Chebira, Yue M. Lu , *Senior Member, IEEE*, and Martin Vetterli , *Fellow, IEEE*

**Abstract**—In a variety of fields, in particular those involving imaging and optics, we often measure signals whose phase is missing or has been irremediably distorted. Phase retrieval attempts to recover the phase information of a signal from the magnitude of its Fourier transform to enable the reconstruction of the original signal. Solving the phase retrieval problem is equivalent to recovering a signal from its auto-correlation function. In this paper, we assume the original signal to be sparse; this is a natural assumption in many applications, such as X-ray crystallography, speckle imaging and blind channel estimation. We propose an algorithm that resolves the phase retrieval problem in three stages, first, we leverage the finite rate of innovation sampling theory to super-resolve the auto-correlation function from a limited number of samples, second, we design a greedy algorithm that identifies the locations of a sparse solution given the super-resolved auto-correlation function, finally, we recover the amplitudes of the atoms given their locations and the measured auto-correlation function. Unlike traditional approaches that recover a discrete approximation of the underlying signal, our algorithm estimates the signal on a continuous domain, which makes it the first of its kind. Along with the algorithm, we derive its performance bound with a theoretical analysis and propose a set of enhancements to improve its computational complexity and noise resilience. Finally, we demonstrate the benefits of the proposed method via a comparison against Charge Flipping, a notable algorithm in crystallography.

**Index Terms**—Phase retrieval, turnpike problem, sparse signals, crystallography, finite rate of innovation, super resolution.

## I. INTRODUCTION

IMAGINE that instead of hearing a song you can only see the absolute value of its Fourier transform (FT) on a graphic equalizer. Can you recover the song from just this visual information? The general answer is “No” as there exist infinitely many signals that fit the curve displayed by the equalizer. However, if we have additional information (or priors) about the song, we may be able to recover it successfully. The reconstruction

process is the subject of this paper and is generally known as phase retrieval (PR).

Beside this day-to-day example, PR is of great interest for many real-world scenarios, where it is easier to measure the FT of a signal instead of the signal itself. During the measurement process, it may happen that the phase of the FT is lost or distorted. Phase loss occurs in many scientific disciplines, particularly those involving optics and communications; a few examples follow.

- X-ray crystallography: we measure the diffraction pattern of a crystallized molecule—that is the magnitude of its FT—and we would like to recover the structure of the molecule itself [1].
- Speckle imaging in astronomy: we measure many images of an astronomic subject and the phase of the images is compromised by the atmospheric distortion. We would like to recover the subject without the resolution downgrade imposed by the atmosphere [2].
- Blind channel estimation of multi-path communication channels: we measure samples of the channel output without knowing the input. We would like to estimate the impulse response of the channel to optimize its capacity [3].

### A. Previous Work

The field of phase retrieval was born along with X-ray crystallography, when the first structures were solved with trial-and-error methods leveraging crystal symmetries. These initial attempts prepared the ground for more systematic approaches, a first example of which was proposed by Patterson in 1935 [4]. This method is based on locating the peaks of the Patterson function—the auto-correlation function of the electron density—to determine pairwise differences between the locations of the atoms constituting a molecule.

In the 1950s, a rich family of approaches exploiting the unique relationships between intensities and phases of measured diffraction patterns was developed, e.g. Cochran [5], Sayre [6], Karle [7]. These methods operate in the Fourier space and are known as *direct methods* because they seek to solve the phase problem directly based on the observed intensities.

We would also like to emphasize the relevance of *dual-space* algorithms, where both spatial and Fourier domains play a fundamental role in reconstructing the signal. While the origin of these methods dates back to 1972 with the work of Gerchberg and Saxton [8], a lot of interest was recently sparked by the introduction of *Charge Flipping* [9], [10].

This short literature review of phase retrieval algorithms in X-ray crystallography is focused on *ab initio* methods, that

Manuscript received July 15, 2018; revised February 11, 2019 and May 13, 2019; accepted July 5, 2019. Date of publication July 26, 2019; date of current version August 15, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Masahiro Yukawa. This work was supported by the Swiss National Science Foundation grant number 20FP-1 151073, Inverse Problems regularized by Sparsity. (Gilles Baechler, Miranda Kreković, and Juri Ranieri are co-first authors.). (Corresponding author: Gilles Baechler.)

G. Baechler and J. Ranieri are with Google, Zurich 8002, Switzerland (e-mail: gilles.baechler@gmail.com; juri.ranieri@gmail.com).

M. Kreković and M. Vetterli are with the EPFL, Lausanne 1015, Switzerland (e-mail: miranda.krekovic@epfl.ch; martin.vetterli@epfl.ch).

A. Chebira was with CSEM, Neuchâtel 2002, Switzerland. She is now with ELCA, Lausanne 1007, Switzerland (e-mail: amina.chebira@gmail.com).

Y. M. Lu is with the Harvard University, Cambridge, MA 02138 USA (e-mail: yuelu@seas.harvard.edu).

Digital Object Identifier 10.1109/TSP.2019.2931169

attempt to solve the phase problem with zero or very little prior information about the structure we are trying to infer. Hence, *ab initio* methods are considered very challenging, given the minimal amount of information they have access to. Successful methods hinge on the design of an abstract data structure that reduces the degrees of freedom of the desired signal and simplifies its reconstruction. For example, direct methods exploit statistical relationships between the phases to reduce the number of unknowns, while Charge Flipping considers a discretization of the electron density.

In this paper, we focus on the PR problem on sparse signals. The sparsity assumption is legitimate and encountered in many applications; for example atoms in crystallography form a sparse structure. We consider the most compact structure one can imagine for a sparse signal: a set of  $K$  atoms defined by their locations  $\mathbf{x}_k$  and their amplitudes  $c_k$ ,

$$f(\mathbf{x}) = \sum_{k=1}^K c_k \phi(\mathbf{x} - \mathbf{x}_k) = f^s(\mathbf{x}) * \phi(\mathbf{x}), \quad (1)$$

where  $f^s(\mathbf{x}) = \sum_{k=1}^K c_k \delta(\mathbf{x} - \mathbf{x}_k)$  represents the structure,  $\mathbf{x}$  is a spatial variable defined over  $\mathbb{R}^D$  (with  $D$  being the dimensionality of the signal),  $\phi(\mathbf{x})$  is the scattering function induced by one atom and  $*$  is the convolution operator. While we assume that both the parameters  $c_k$  and the range of  $\phi(\mathbf{x})$ <sup>1</sup> belong to  $\mathbb{R}$  to avoid a heavier notation, our results can be easily extended to  $\mathbb{C}$ .

Even if the advantage of the compact model defined in (1) looks appealing, the associated algorithmic challenges are often overwhelming. Computer scientists attempted to design a scalable (i.e. with a computational complexity that is polynomial in the number of atoms  $K$ ) and stable to noise algorithm that could solve all possible instances of this problem without much success; to date, it is not even clear that such an algorithm would exist [11]. In other words, we encounter a nontrivial trade-off between the compactness of such data structures (i.e. the number of unknown variables) and the ease of solving the PR problem using them. For example, Charge Flipping easily solves many PR problems in X-ray crystallography, but it is based on a discrete spatial structure, which is definitely less compact than (1).

Recently, several sparse PR algorithms have been proposed assuming a discrete spatial domain, often borrowing inspiration from the area of compressed sensing. Two notable examples are *GrEedy Sparse PhAse Retrieval* (GESPAR) [12], based on the *2-opt* algorithm [13], and *Two-stage Sparse Phase Retrieval* (TSPR) [14], where the support is recovered by solving the discrete *turnpike problem* [15], [16]. As mentioned, both algorithms differ from our approach in that their models are discrete and the locations of the spikes are bound to a discrete grid. Even though it was not designed with continuous setups in mind, TSPR can theoretically recover locations on a continuous domain. However, while it handles noise on the measured coefficients, it does not tolerate noise in the support, which makes it impractical

for continuous setups. Other approaches with discrete support include [17], where the measurement matrix has random entries, PhaseLift [18], where the sampling matrix entries are independently sampled on the unit sphere, or [19], which is a variation of GESPAR based on the short-time Fourier transform.

The major benefit of having a continuous parametric model is that it enables estimation of the locations and amplitudes avoiding any discretization. In such a case, the achievable resolution is theoretically infinite and only limited by the noise corrupting the measurements. This is what we call *super resolution* phase retrieval. Perhaps surprisingly, the continuous sparse phase retrieval problem has received little attention in the literature. During the completion of this manuscript, we became aware of the work of Beinert *et al.* [20], [21]. They propose a super-resolution approach based on the finite rate of innovation (FRI) framework, which is also one of the building blocks of the proposed algorithm in this paper. Nonetheless, our work represents a significant improvement over the state of the art as it is robust to noise in the measurements.

## B. Main Contributions and Outline

As discussed in the previous section, the literature on sparse phase retrieval is fairly rich. Most of the prior art assumes a discrete spatial domain and proposes solutions inspired by various non-convex optimization techniques like compressed sensing and dual-space methods. In this paper, we consider the more realistic and challenging scenario of a sparse signal defined on a continuous domain (1). Current literature offers only few results in this settings and they are not usable in practice mostly due to their instability to noise. Our first main contribution is a three-stage algorithm that precisely determines a continuous sparse signal from the absolute value of its FT, even in presence of significant noise. We show that the success rate of the proposed algorithm exhibits a sharp phase transition, which is a function of the complexity of the signal, i.e. the number of sparse elements, and the noise affecting the inputs. In other words, given a randomly generated input, the algorithm has either a very high or very low probability of success. To precisely determine the boundary of such a phase transition, we propose an approximated theoretical analysis of the algorithm performance as a function of the aforementioned parameters.

In Section II, we formalize the problem and describe the typical PR measurement pipeline. In Section III, we give a high-level overview of our modular approach, discuss the main challenges and introduce a few relevant properties. Algorithms to solve these different modules are proposed in Section IV.

We then describe the details of the proposed method to recover the support, which constitutes the critical element of the pipeline: its complexity analysis can be found in Section V, together with a method to reduce its computational cost, while Section VI identifies an approximated theoretical bound (confirmed by numerical simulations) to successfully recover the signal support in a noisy regime. Then in Section VII, we propose a few improvements and variations of the algorithm to make it more robust to noise. In Section VIII, we discuss the influence of the support configuration on the resulting reconstruction.

<sup>1</sup>Note that we also require some additional assumptions regarding the kernel  $\phi(\mathbf{x})$ ; however, these assumption are introduced later in the paper, where the context clarifies them.

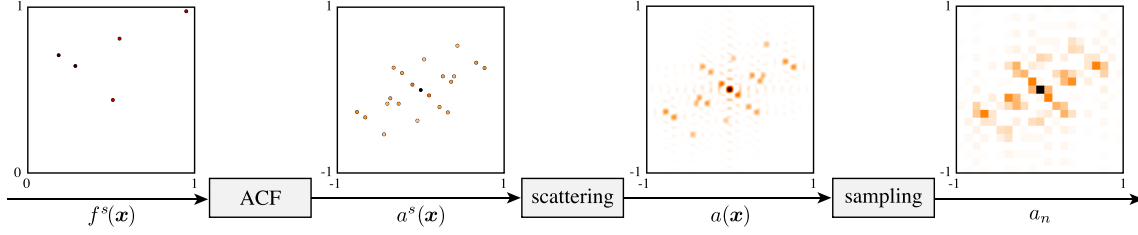


Fig. 1. Typical PR measurement pipeline: the signal of interest  $f^s(\mathbf{x})$  generates the auto-correlation function  $a^s(\mathbf{x})$ , which is first filtered by the scattering function  $\psi(\mathbf{x})$  (here an ideal lowpass filter) to yield  $a(\mathbf{x})$  and then sampled, resulting in  $a_n$ . Note that the spatial samples  $a_n$  can be obtained via the inverse discrete FT of the Fourier samples  $A_m$ , when the periodicity in the two domains holds. Darker colors represent higher intensities.

Finally, Section IX compares our PR pipeline with the state-of-the-art.

Throughout this paper, we use bold lower case letters for vectors and bold upper case letters for matrices. Upper case calligraphic letters denote sets, e.g.  $\mathcal{X}$ . Furthermore,  $\hat{\mathcal{X}}$  represents a set with noisy elements and  $\tilde{\mathcal{X}}$  an estimated set. Subscripts are reserved for indexing elements in lists and vectors. In the primal domain, continuous functions are written in lower case letters and indexed with  $\mathbf{x}$ , e.g.  $f(\mathbf{x})$  and discrete functions are indexed with  $\mathbf{n}$ , e.g.  $f_n$ . In the Fourier domain, we use capital letters, that is  $F(\boldsymbol{\omega})$  and  $F_m$ , for continuous and discrete functions, respectively.

## II. PROBLEM STATEMENT

We consider the FT of the signal defined in (1),

$$F(\boldsymbol{\omega}) = \sum_{k=1}^K c_k \exp\{-j\boldsymbol{\omega}^\top \mathbf{x}_k\} \Phi(\boldsymbol{\omega}), \quad (2)$$

where  $\boldsymbol{\omega}$  is the frequency variable and  $\Phi(\boldsymbol{\omega})$  is the FT of the known kernel  $\phi(\mathbf{x})$ .

In practice, it is impossible to measure the whole FT (2), hence we sample it. Furthermore, due to limitations of the measurement setup, we are usually only able to measure the absolute values of such samples, that we denote  $|F_m|$ , where  $F_m = F(\mathbf{m}\Omega)$ ,  $\mathbf{m} = \mathbb{Z}^D$  and  $\Omega$  is the sampling frequency. As previously mentioned, the PR problem has infinite solutions without a-priori knowledge of the signal  $f(\mathbf{x})$ , since we can assign any phase to the measurements and obtain a plausible reconstruction. The role of structures, such as (1), is to constrain the PR problem to a correct and possibly unique solution. Under the sparsity assumption, retrieving the phase is equivalent to retrieving the locations and amplitudes of  $f(\mathbf{x})$ .

The auto-correlation function (ACF)  $a(\mathbf{x})$  of  $f(\mathbf{x})$  is given by the inverse FT of  $|F(\boldsymbol{\omega})|^2$ :

$$a(\mathbf{x}) = f(\mathbf{x}) * f(-\mathbf{x}) = \mathcal{F}^{-1} [|F(\boldsymbol{\omega})|^2], \quad (3)$$

where  $\mathcal{F}^{-1}$  is the inverse FT operator [22]. Interestingly, the ACF structure is completely inherited from the signal (1):

$$\begin{aligned} a(\mathbf{x}) &= \sum_{k=1}^K \sum_{\ell=1}^K c_k c_\ell \psi(\mathbf{x} - (\mathbf{x}_k - \mathbf{x}_\ell)) \\ &= \left[ \sum_{k=1}^K \sum_{\ell=1}^K c_k c_\ell \delta(\mathbf{x} - (\mathbf{x}_k - \mathbf{x}_\ell)) \right] * \psi(\mathbf{x}) \end{aligned}$$

$$= a^s(\mathbf{x}) * \psi(\mathbf{x}), \quad (4)$$

where the kernel  $\psi(\mathbf{x})$  is the ACF of  $\phi(\mathbf{x})$  and  $a^s(\mathbf{x})$  is the ACF of the sparse structure of the train of Diracs  $f^s(\mathbf{x})$ . Equivalently, in the Fourier domain, we have

$$A(\boldsymbol{\omega}) = \sum_{k=1}^K \sum_{\ell=1}^K c_k c_\ell \exp\{-j\boldsymbol{\omega}^\top (\mathbf{x}_k - \mathbf{x}_\ell)\} |\Phi(\boldsymbol{\omega})|^2. \quad (5)$$

The PR acquisition pipeline can be summarized as the filtering of the ACF  $a^s(\mathbf{x})$  followed by sampling, where the filtering represents the scattering operation, as illustrated in Fig. 1. We now have all the ingredients to state the core problem of this paper.

*Problem 1:* Given Fourier samples  $A_m = A(\mathbf{m}\Omega)$  of the sparse ACF defined in (4), recover the support  $\mathcal{X} = \{\mathbf{x}_k\}_{k=1}^K$  and amplitudes  $\{c_k\}_{k=1}^K$  determining the signal  $f(\mathbf{x})$ .

Note that the information we are interested in is hidden behind two walls: the convolution with the kernel  $\psi(\mathbf{x})$  that spatially blurs the sparse structure of the ACF and the phase loss of the original sparse signal,  $f^s(\mathbf{x})$ , that usually characterizes any PR problem.

## III. A THREE-STAGE APPROACH

We propose to solve Problem 1 in three distinct stages: i) reconstruct the continuous ACF  $a(\mathbf{x})$  from a set of its discrete Fourier coefficients, ii) estimate the support  $\mathcal{X}$  of  $f(\mathbf{x})$  given  $a(\mathbf{x})$ , and iii) estimate its amplitudes  $\{c_k\}_{k=1}^K$ .

The first step is a classical sampling problem where we would like to fully characterize a continuous sparse signal from a set of discrete measurements.

*Problem 1.A (Sparse ACF super resolution):* Given samples  $A_m$  of the sparse ACF as defined in (4), recover its continuous version  $a(\mathbf{x})$ .

The most well-known sampling result is due to Nyquist-Shannon-Kotelnikov and guarantees perfect recovery for signals that lie in the subspace of bandlimited functions, provided that the sampling rate is high enough.

However, we suggest to leverage the sparsity of the underlying signal and recover the ACF  $a(\mathbf{x})$  via its parameters  $c_k c_\ell$  and  $\mathbf{x}_k - \mathbf{x}_\ell$ . Such parameters are then used for the successive steps of the PR process. Sparsity has two antagonistic effects on PR: it makes the problem combinatorial and hence hard to solve, but at the same time enables a divide-and-conquer approach, in which we first recover the support  $\{\mathbf{x}_k\}_{k=1}^K$  and then the amplitudes

$\{c_k\}_{k=1}^K$  of  $f(x)$ . We argue that the support contains more information than the amplitudes, hence we choose to estimate it first.

As an example, if all the atoms have the same amplitude, then only the support is useful to recover the original signal. On the other hand, if all the atoms have the same location, the problem is trivially solvable.

*Problem 1.B (Support recovery):* Assume we are given the complete set of unlabeled differences  $\mathcal{D} = \{\mathbf{d}_{k,\ell}\}_{k,\ell} = \{\mathbf{x}_k - \mathbf{x}_\ell\}_{k,\ell}$ , recover the support  $\mathcal{X} = \{\mathbf{x}_k\}_{k=1}^K$  of the sparse signal  $f(\mathbf{x})$ .

In most real-world scenarios, the set  $\mathcal{D}$  is derived from the noisy samples  $a_m$  of the ACF  $a(\mathbf{x})$  by solving Problem 1.A and hence the unlabeled differences of Problem 1.B are corrupted by noise. More precisely, noise is first introduced when measuring the samples of the ACF  $a_m$ . Such a noise is independently distributed and added to the ACF samples, and not to the distances  $\mathcal{D}$ . Next, the algorithm that recovers the distances from the ACF deforms and amplifies the initial measurement noise. Hence, we always obtain a noisy and non-symmetrical set of distances, instead of the noiseless set  $\mathcal{D}$ . Throughout this paper, we generically refer to the sampling artifacts and their propagation due to the reconstruction algorithm as noise.

In our specific case, we propose to solve the super-resolution of the ACF with an FRI-based algorithm, whose noise analysis is not trivial. We refer the interested reader to [23]–[26] for more details about it: theoretical bounds such as the Cramér-Rao [27] bound and the Barankin bound [28] are derived in [24]–[26]. Moreover, Monte Carlo simulations are provided in [26], showing the reconstruction noise distribution along with the theoretical Cramér-Rao bound.

While the literature clearly defines the noise affecting the distances  $\mathbf{d}_{k,\ell}$  as neither Gaussian nor independently distributed, we model such a noise as i.i.d. Gaussian random variables,

$$\tilde{\mathbf{d}}_{k,\ell} = \mathbf{d}_{k,\ell} + \boldsymbol{\nu}_{k,\ell}, \quad (6)$$

where  $\boldsymbol{\nu}_{k,\ell} \sim \mathcal{N}(\mathbf{0}, \sigma \mathbf{I})$ .

We further denote the set of measured differences as  $\tilde{\mathcal{D}} = \{\tilde{\mathbf{d}}_{k,\ell}\}_{k,\ell}$ . For simplicity of notation, we convert the pairs of indices  $(k, \ell)$  to  $n \in \{1, \dots, N\}$ , where  $N = K^2 - K + 1$ , and order them such that  $\|\tilde{\mathbf{d}}_1\| \leq \|\tilde{\mathbf{d}}_2\| \leq \dots \leq \|\tilde{\mathbf{d}}_N\|$ . We do not assume any ordering on the elements of  $\mathcal{X}$ . While this noise model is definitely a simplification and approximation of the known theoretical models, it enables us to analyze the performance of the proposed algorithm in Section VI and such analysis surprisingly matches what we observe in our experiments.

In what follows, we state a few interesting observations related to Problem 1.B. First, when we measure a set of differences, some information is inevitably lost and hence we should not expect a unique solution but a set of equivalent solutions.

*Observation 1:* A set of points can be reconstructed from their pairwise differences, even when labeled, only up to shifts and reflections.

To show that, we first translate and reflect the set of points  $\mathcal{X}$  as  $\mathcal{X}' = -\mathcal{X} + \bar{\mathbf{x}}$ , where we overload the arithmetic operators on sets to transform each point as  $\mathbf{x}'_k = -\mathbf{x}_k + \bar{\mathbf{x}}$ . Then, the

set of differences of the transformed points is equivalent to the original one,

$$\mathbf{d}'_{k,\ell} = \mathbf{x}'_k - \mathbf{x}'_\ell = -\mathbf{x}_k + \bar{\mathbf{x}} + \mathbf{x}_\ell - \bar{\mathbf{x}} = \mathbf{x}_\ell - \mathbf{x}_k = -\mathbf{d}_{k,\ell},$$

where the natural symmetry of  $\mathcal{D}$  compensates for the negative sign.

Second, while excluding shifts and reflections does not lead to a unique solution in general, we can still prove uniqueness under certain assumptions.

*Observation 2:* Assume that the points  $\mathbf{x}_k$  are drawn independently at random from a sufficiently smooth distribution, then the solution is unique [29].

Third, we briefly discuss the occurrence of *collisions* in the ACF. We say that there is a collision in the ACF when two different pairs of distinct points from  $\mathcal{X}$  map to the same difference in  $\mathcal{D}$ . Since we consider a continuous domain for the support, it natively prevents the appearance of collisions.<sup>2</sup>

*Observation 3:* If the locations of the points are independently drawn uniformly from a finite interval, then collisions in the ACF occur with probability zero.

Last, we note that the set of differences  $\mathcal{D}$  contains many valid solutions. In particular, we can construct two solutions from every element of  $\mathcal{X}$ ; this is a direct consequence of Observation 1.

*Observation 4:* The set of differences  $\mathcal{D}$  is a superset of  $2K$  valid solutions  $\hat{\mathcal{X}}$  to Problem 1.B and such solutions always contain the point zero, that is  $\mathbf{0} \in \hat{\mathcal{X}}$ .

To verify this, we pick an element of the support, e.g.  $\mathbf{x}_\ell$ , and build the following tentative solution,

$$\hat{\mathcal{X}} = \{\mathbf{x}_k - \mathbf{x}_\ell \mid k = 1, \dots, K\}. \quad (7)$$

Then, we notice that i)  $\hat{\mathcal{X}}$  is a valid solution with the shift fixed as  $-\mathbf{x}_\ell$ , ii)  $\hat{\mathcal{X}} \subset \mathcal{D}$  and iii) we have a solution for every element of  $\mathcal{X}$ . Moreover, due to the symmetry of the ACF, the set

$$\{\mathbf{x}_\ell - \mathbf{x}_k \mid k = 1, \dots, K\} \subset \mathcal{D} \quad (8)$$

is also a valid solution, so we reach the aforementioned  $2K$  solutions. Such an observation can be generalized to the noisy scenario: the  $2K$  subsets can still be identified and each subset represents a solution to the support recovery problem. However, these solutions are similarly corrupted by the input noise. This property is essential to the algorithm for support recovery proposed in the next section.

Once the support  $\hat{\mathcal{X}}$  of the solution has been retrieved, it remains to find the amplitudes  $\{c_k\}_{k=1}^K$  of the signal  $f(\mathbf{x})$ .

*Problem 1.C (Amplitude recovery):* Given an ACF  $a(\mathbf{x})$  as defined in (4) together with the estimated support  $\hat{\mathcal{X}}$  of  $f(\mathbf{x})$ , find the amplitudes  $\{c_k\}_{k=1}^K$ .

#### IV. ALGORITHMS

In this section, we lay down our solutions to Problems 1.A, 1.B and 1.C, effectively providing an end-to-end framework to solve the sparse PR problem.

<sup>2</sup>The support recovery algorithm proposed in this paper can in fact handle collisions and could be used on a discretized space as well. However, assuming no collisions simplifies the recovery of the amplitudes and enables a few improvements to make the algorithm more resilient to noise.

### A. ACF Super Resolution

When we look at (4), we notice that  $a(\mathbf{x})$  is completely defined by the locations  $\mathbf{x}_k - \mathbf{x}_\ell$  and the amplitudes  $c_k c_\ell$ . Hence, we can recast Problem 1.A as a parameter estimation problem given the measured samples  $A_m$  of the FT of the ACF. An effective existing approach is known as *finite rate of innovation* (FRI) sampling [30], [31]. FRI-based methods are inspired by spectral analysis techniques to estimate the locations  $\mathbf{x}_k - \mathbf{x}_\ell$ ; in what follows, we review their fundamentals for the sake of completeness. In this subsection, we restrict ourselves to the 1-dimensional case for clarity, even though our implementation is generalized to higher dimensions. The higher dimensional case was first discussed by Maravic [23], who proposed a first algorithm requiring  $\mathcal{O}(N^D)$  samples, where  $D$  is the number of dimensions. More recently, Pan *et al.* [32] came up with a multidimensional reconstruction algorithm using only  $\mathcal{O}(N)$  samples.

The essential ingredient in FRI is to represent the signal of interest as a weighted sum of complex exponentials in the following form:

$$b_m = \sum_{n=1}^N \alpha_n u_n^m. \quad (9)$$

This formulation has several similarities with (5); to see this, we define  $t_n = x_k - x_\ell$ , substitute  $\alpha_n = c_k c_\ell$  and  $u_n = \exp(-j\Omega t_n)$  and rewrite the sampled ACF  $A_m$  as follows,

$$A_m = \sum_{n=1}^N \alpha_n u_n^m |\Phi(m\Omega)|^2. \quad (10)$$

We remark that  $|\Phi(m\Omega)|^2$  does not allow us to express (10) as a sum of complex exponentials yet. However, if we assume that the kernel function  $\phi(x)$  is an ideal low-pass filter,<sup>3</sup> i.e. a sinc function, its FT becomes a box function. Thus, we can ignore such a kernel for some neighborhood of  $m$  around zero, since  $\Phi(m\Omega) = 1$  for  $|m\Omega|$  smaller than the bandwidth of the signal.

The locations  $\{d_n\}_{n=1}^N$  are fully determined by the exponentials  $\{u_n\}_{n=1}^N$ , that is  $d_n = \frac{\angle u_n}{\Omega}$ , with  $\angle u_n$  being the phase of  $u_n$ . Recovering  $u_n$  from (9) is a classical spectral estimation technique and a possible solution is provided by *Prony's method* [36], [37]. The idea is to identify a filter  $H_m$  to *annihilate*  $A_m$ , which is mathematically defined as

$$(A * H)_m = 0. \quad (11)$$

Then, the filter  $H$  can be estimated by rewriting and solving (11) as a Toeplitz system. As shown in [30], if  $A_m$  has the form of (9), then the  $z$ -transform of  $H_m$  is

$$H(z) = \sum_{n=0}^N A_n z^{-n} = \prod_{n=1}^N (1 - u_n z^{-1}), \quad (12)$$

where  $u_n$  are nothing else but the roots of  $H(z)$ . Our situation differs from usual FRI applications in the sense that the locations

<sup>3</sup>The FRI theory has also been generalized to a wide range of kernels such as combinations of B-splines and E-splines [31], [33], [34] or even arbitrary sampling kernels [35], where a linear operation enables to obtain the desired form (9) from (10).

of the ACF describe a symmetric structure. As a consequence, all roots  $u_n$  come in conjugate pairs (except for the one corresponding to the zero location).<sup>4</sup>

Once the locations are known, the amplitudes  $\alpha_n$  are found by injecting  $u_n$  in (10) and solving a linear system of equations. In this case, exploiting the symmetry of the ACF helps and our experiments have shown that it significantly improves the reconstruction: grouping the symmetric pairs together results in  $(K^2 - K)/2 + 1$  separate values to estimate.

### B. Support Recovery

For the recovery of the support, we propose a novel greedy algorithm that is initialized with a partial solution  $\hat{\mathcal{X}}_2$ , which contains two locations. At a given iteration  $k$ , we generate a partial solution  $\hat{\mathcal{X}}_{k+1}$  composed of  $k + 1$  locations, hence the algorithm has a total of  $K - 2$  iterations indexed from 2 to  $K - 1$ .

1) *Initialization*: From Observation 4, we know that the solution set  $\hat{\mathcal{X}}$  is contained in  $\tilde{\mathcal{D}}$  and  $\mathbf{0} \in \mathcal{X}$ ; this gives us the first point of the solution, that is  $\hat{\mathbf{x}}_1 = \mathbf{0}$ . Next, we identify the element  $\tilde{\mathbf{d}}_N$  in  $\tilde{\mathcal{D}}$  with the largest norm, so that we maximize the noise resilience of our algorithm. Indeed, assuming that the locations are corrupted by identically distributed noise, picking the largest norm ensures the maximal SNR of our initial solution.

Note that the value  $\tilde{\mathbf{d}}_N$  is the noisy difference between two unknown locations of  $f(\mathbf{x})$ ; without loss of generality, we call them  $\mathbf{x}_1$  and  $\mathbf{x}_2$ . The elements  $\hat{\mathbf{x}}_1 = \mathbf{0}$  and  $\hat{\mathbf{x}}_2 = \tilde{\mathbf{d}}_N$  are nothing but  $\mathbf{x}_1$  and  $\mathbf{x}_2 + \boldsymbol{\nu}_{2,1}$  translated by  $-\mathbf{x}_1$ . Therefore, we are always guaranteed that the initialized solution  $\hat{\mathcal{X}}_2 = \{\mathbf{0}, \tilde{\mathbf{d}}_N\}$  is a (noisy) subset of the set of locations  $\mathcal{X} - \mathbf{x}_1$ .

Referring again to Observation 4, we know that the set of differences  $\tilde{\mathcal{D}}$  contains the rest of the points  $\{\mathbf{x}_k - \mathbf{x}_1 + \boldsymbol{\nu}_{k,1}\}_{k=3}^K$ , that should belong to the final solution  $\hat{\mathcal{X}} = \hat{\mathcal{X}}_K$ . Furthermore, since we do not want to duplicate points in  $\hat{\mathcal{X}}_k$ , we initialize a set of possible elements of the solution  $\mathcal{P}_2 = \tilde{\mathcal{D}} \setminus \{\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_N\}$ . Due to noise, the vector  $\mathbf{0}$  is not in  $\tilde{\mathcal{D}}$ , so we remove the closest element  $\tilde{\mathbf{d}}_1$ .

2) *Main Algorithm*: In the noiseless case, we would like to identify the element  $\hat{\mathbf{x}}_{k+1} \in \mathcal{P}_k$  to be added to the current solution set  $\hat{\mathcal{X}}_k$ , that is  $\hat{\mathcal{X}}_{k+1} = \hat{\mathcal{X}}_k \cup \hat{\mathbf{x}}_{k+1}$ . This new element of the solution should be chosen such that the pairwise differences between the elements of  $\hat{\mathcal{X}}_{k+1}$  form a subset of  $\mathcal{D}$ .

This intuition can be generalized to the noisy case: we would like to identify the element  $\hat{\mathbf{x}}_{k+1} \in \mathcal{P}_k$  such that the set of pairwise differences of the points in  $\hat{\mathcal{X}}_{k+1} = \hat{\mathcal{X}}_k \cup \hat{\mathbf{x}}_{k+1}$  is the *closest to be a subset* of the measured  $\tilde{\mathcal{D}}$ . To that end, we propose to generalize the concept of subset to a noisy environment by searching for the differences in  $\tilde{\mathcal{D}}$  that are closest in  $\ell^2$ -norm to the pairwise differences of the elements in  $\hat{\mathcal{X}}_{k+1}$ .

At each step  $k$ , we identify the element in  $\mathcal{P}_k$  that, when added to the partial solution  $\hat{\mathcal{X}}_k$ , minimizes the error with respect to the measured set of differences  $\tilde{\mathcal{D}}$ . More precisely, at every iteration

<sup>4</sup>Note we tried to exploit this fact by enforcing the symmetry of the roots given as conjugate pairs. However, our experiments did not show a significant impact.

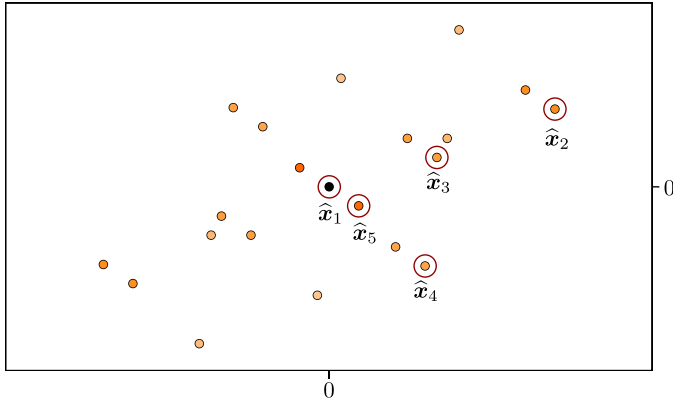


Fig. 2. 2D Instance of Algorithm 1 on the ACF  $a^s(\mathbf{x})$  from Fig. 1. We start by setting  $\hat{\mathbf{x}}_1 = \mathbf{0}$  and identifying  $\hat{\mathbf{x}}_2$ , the point with the largest norm. Points  $\hat{\mathbf{x}}_3$  to  $\hat{\mathbf{x}}_5$  are then selected in a greedy way according to (13). The solution coincides with the initial signal  $f^s(\mathbf{x})$  displayed in Fig. 1.

---

**Algorithm 1:** Support Recovery.
 

---

**Input:** A set of  $N = K^2 - K + 1$  differences  $\tilde{\mathcal{D}} = \{\tilde{\mathbf{d}}_n\}_{n=1}^N$  ordered by their norms

**Output:** A set of  $K$  points  $\hat{\mathcal{X}}$  such that their pairwise differences generate  $\tilde{\mathcal{D}}$

$$\hat{\mathcal{X}}_2 = \{\mathbf{0}, \tilde{\mathbf{d}}_N\}$$

$$\mathcal{P}_2 = \tilde{\mathcal{D}} \setminus \{\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_N\}$$

**for**  $k = 2, \dots, K - 1$  **do**

$$\hat{\mathbf{x}}_{k+1} = \underset{\mathbf{p} \in \mathcal{P}_k}{\operatorname{argmin}} \sum_{\hat{\mathbf{x}} \in \hat{\mathcal{X}}_k} \min_{\tilde{\mathbf{d}} \in \tilde{\mathcal{D}}} \|\mathbf{p} - \hat{\mathbf{x}} - \tilde{\mathbf{d}}\|^2$$

$$\hat{\mathcal{X}}_{k+1} = \hat{\mathcal{X}}_k \cup \hat{\mathbf{x}}_{k+1}$$

$$\mathcal{P}_{k+1} = \mathcal{P}_k \setminus \hat{\mathbf{x}}_{k+1}$$

**end for**

**return**  $\hat{\mathcal{X}}_K$

---

$k$  we solve the following optimization problem,

$$\hat{\mathbf{x}}_{k+1} = \underset{\mathbf{p} \in \mathcal{P}_k}{\operatorname{argmin}} \sum_{\hat{\mathbf{x}} \in \hat{\mathcal{X}}_k} \min_{\tilde{\mathbf{d}} \in \tilde{\mathcal{D}}} \|\mathbf{p} - \hat{\mathbf{x}} - \tilde{\mathbf{d}}\|^2. \quad (13)$$

This procedure is summarized in Algorithm 1 and its application on the ACF  $a^s(\mathbf{x})$  from Fig. 1 is illustrated in Fig. 2.

### C. Amplitude Recovery

If we assume that collisions can occur, recovering the amplitudes with a given ACF and support is equivalent to solving a system of quadratic equations. However, if there are no collisions, we suggest a simple but efficient algebraic solution to Problem 1.C, inspired from [29]. While the method of [29] relies on a matrix inversion step to solve the problem, we propose here to work in the logarithmic domain. Numerical simulations have shown that it is both faster and more robust to noise.

Let  $\mathbf{c} = [c_1, c_2, \dots, c_K]^\top$  be a vector made of the amplitudes to be recovered. If we define a matrix  $\mathbf{C} = \mathbf{c}\mathbf{c}^\top$ , all the elements outside of the diagonal of such a matrix are the amplitudes of the measured ACF, that is  $C_{i,j} = c_i c_j$ . Notice that we cannot

observe the diagonal entries  $C_{i,i} = c_{i,i}^2$ , as we just have access to their sum  $a_0^s = \sum_i c_{i,i}^2$ , which is the value of the ACF at 0. This is unfortunate since they are precisely the values we are interested in, up to a squaring operator.

We recast Problem 1.C as a matrix completion problem, where we would like to estimate the diagonal entries  $C_{i,i}$  under the constraint of  $\mathbf{C}$  being a rank-one matrix. The first step of our proposed method is to introduce a matrix  $\mathbf{L}$  such that

$$L_{i,j} = \begin{cases} \log(C_{i,j}) = \ell_i + \ell_j & \text{for } i \neq j \\ 0 & \text{otherwise,} \end{cases} \quad (14)$$

where  $\ell_i = \log(c_i)$ . The sum of the  $i$ th row of  $\mathbf{L}$  is given by

$$\sum_{j=1}^K L_{i,j} = (K-1)\ell_i + \sum_{j=1}^K \ell_j - \ell_i = (K-2)\ell_i + \sum_{j=1}^K \ell_j, \quad (15)$$

where the term  $\sum_j \ell_j$  does not vary between rows. Hence, its value can be obtained from summing all the entries in  $\mathbf{L}$ ,

$$\begin{aligned} s &= \sum_{i=1}^K \sum_{j=1}^K L_{i,j} = (K-2) \sum_{i=1}^K \ell_i + K \sum_{j=1}^K \ell_j \\ &= 2(K-1) \sum_{j=1}^K \ell_j. \end{aligned} \quad (16)$$

Then, we recover the vector  $\boldsymbol{\ell} = [\ell_1, \ell_2, \dots, \ell_K]^\top$  for  $K > 2$  as

$$\boldsymbol{\ell} = \frac{1}{K-2} \left( \mathbf{L}^\top \mathbf{1} - \frac{s}{2(K-1)} \mathbf{1} \right), \quad (17)$$

where  $\mathbf{1}$  is the all-ones vector.<sup>5</sup> Finally, it suffices to compute  $c_i = \exp(\ell_i)$  to retrieve the amplitudes.

Note that this solution assumes that  $\mathbf{C}$  is symmetric; this might not be the case in a noisy setup, but we enforce it by replacing  $\mathbf{C}$  with  $\frac{1}{2}(\mathbf{C} + \mathbf{C}^\top)$ . In case of collisions, the problem does not have an algebraic solution and a possible convex relaxation is provided in [14]. In practice, this is often not a concern due to Observation 3.

Putting all pieces together, these three stages combines to enable the recovery of a continuous signal from its noisy sampled ACF; Figure 3 illustrates a few examples of recoveries of trains of Diracs based on the combination of these three steps.

In what follows, we study and propose improvements to the performance of our PR algorithm, focusing our attention on the support recovery step, i.e. Algorithm 1. In fact, the first step—the super-resolution with FRI—is well represented in literature, where theoretical analyses, extensive simulations in noisy scenarios and efficient denoising schemes have been proposed [23], [24], [38]. On the other hand, the amplitude recovery, while being novel, only consists of simple algebraic manipulations that are not computationally costly.

<sup>5</sup>When  $K = 2$ , the entries  $\ell_1, \ell_2$  can be recovered by solving a system of two equations.

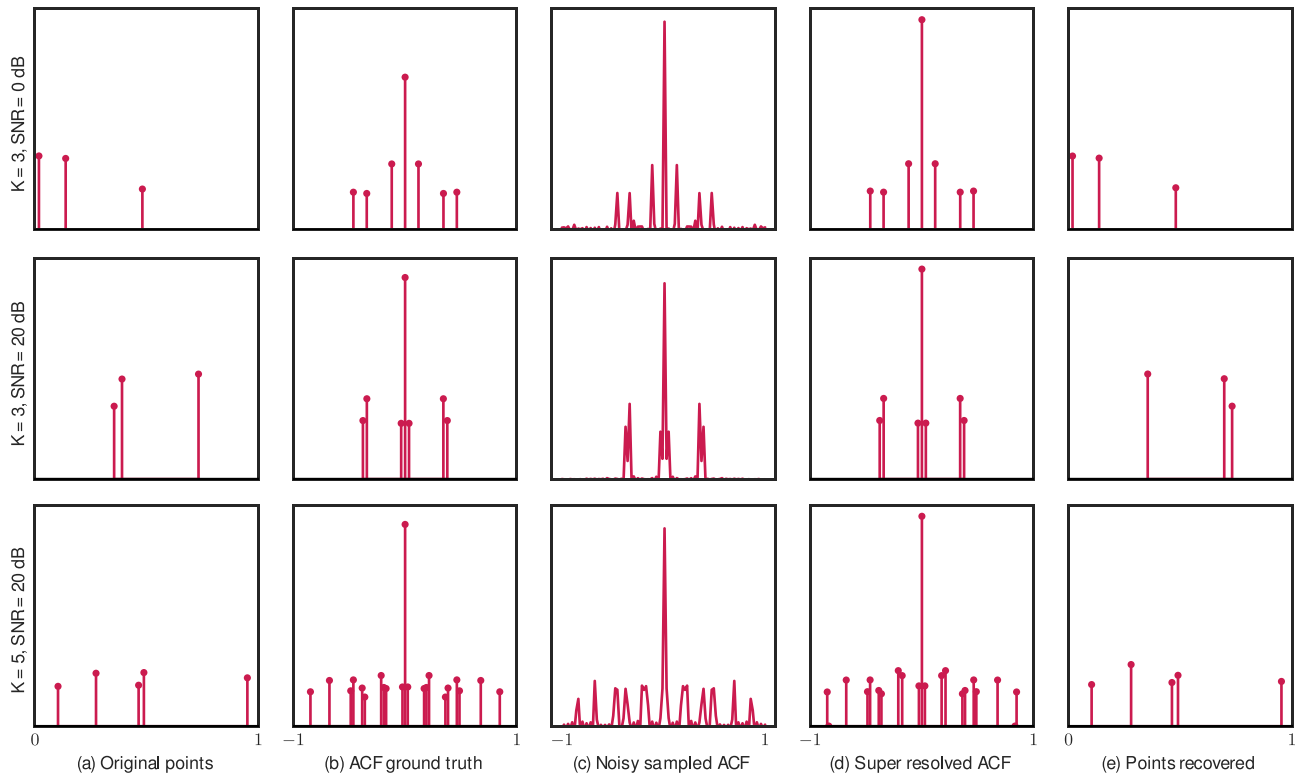


Fig. 3. Examples of our algorithm in 1 dimension for different values of  $K$  and different noise regimes: (a) original points; (b) corresponding continuous ACF; (c) discrete noisy ACF with 100 samples (sinc sampling kernel used); (d) output of the FRI-based super-resolution algorithm; (e) result of the support recovery algorithm.

## V. COMPLEXITY ANALYSIS

Algorithm 1 has  $K$  rounds. In each of these rounds, we go through all points in the existing solution set  $\hat{\mathcal{X}}_k$ , and for each point we compute the difference with all the values in  $\tilde{\mathcal{D}}$ . Since there are  $\mathcal{O}(K)$  points in  $\hat{\mathcal{X}}_k$  and  $\mathcal{O}(K^2)$  elements in  $\tilde{\mathcal{D}}$ , this is done in  $\mathcal{O}(K^3)$  operations. Furthermore, for each of these computed differences, we need to find the closest element in  $\tilde{\mathcal{D}}$ , which requires additional  $\mathcal{O}(K^2)$  comparisons. In total, the complexity of our algorithm is  $\mathcal{O}(K^6)$ . Even though this is high and limits the field of application to reasonable sizes, it compares favorably to an exhaustive search strategy, which grows exponentially with  $K$ .

It is possible to trade time complexity for storage complexity. Indeed, we observe that we compute at each round the following values

$$\tilde{\mathbf{d}}_{i,j} = \underset{\tilde{\mathbf{d}} \in \tilde{\mathcal{D}}}{\operatorname{argmin}} \|\mathbf{p}_j - \hat{\mathbf{x}}_i - \tilde{\mathbf{d}}\|^2, \quad (18)$$

for every point  $\hat{\mathbf{x}}_i \in \hat{\mathcal{X}}_k$  and candidate  $\mathbf{p}_j \in \mathcal{P}_k$ . However, since we are just moving one element from  $\mathcal{P}_k$  to  $\hat{\mathcal{X}}_{k+1}$  at each iteration, we propose to cache the values (18) in a lookup table to reduce the total computational cost. By doing so, we only need to update each  $\tilde{\mathbf{d}}_{i,j}$  when the corresponding candidate  $\mathbf{p}_j$  is removed from  $\mathcal{P}_k$  to be added to  $\hat{\mathcal{X}}_{k+1}$ .

The theoretical complexity when caching  $\tilde{\mathbf{d}}_{i,j}$  is not trivial to analyze, but in practice we notice a significant improvement, as illustrated in Fig. 4.

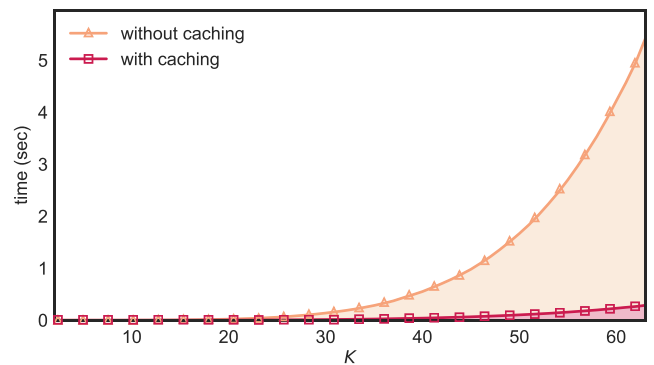


Fig. 4. Comparison of the average run time of the original algorithm and its cached version. The times reported are the average of 100 runs of the algorithm. The dashed lines represent curves of the form  $CK^\alpha$  that are fitted to the data. For the method without caching, we have  $C = 4.25 \cdot 10^{-6}$  and  $\alpha = 5.06$ , while for the method with caching we have  $C = 3.88 \cdot 10^{-6}$  and  $\alpha = 4.37$ . Remark how the caching is reducing the polynomial degree of the computational cost by approximately one.

## VI. PERFORMANCE ANALYSIS

In what follows, we study the expected performance of Algorithm 1 in the presence of noise.

More precisely, we model the probability that Algorithm 1 finds the correct solution as a function of the noise variance  $\sigma^2$  and the number of elements  $K$  to characterize its performance. We consider a one-dimensional problem, that is  $D = 1$ , to lighten notation and simplify the discussion. However, all the

results can be easily generalized to the multidimensional setup introduced in Problem 1.B. Note that our analysis requires several approximations for the sake of tractability. Nonetheless, we consider this analysis to be of interest. First, our approximated analysis matches incredibly closely the empirical performance of the algorithm, see Section VI-A. Second, other researchers may find our results interesting as a starting point for a more rigorous approach.

We start our analysis by observing that the probability of success can be factored along the  $K - 2$  iterations

$$P(\sigma, K) = \prod_{k=2}^{K-1} P_k(\sigma, K), \quad (19)$$

where  $P(\sigma, K)$  is the probability of success of the support recovery algorithm and  $P_k(\sigma, K)$  is the conditional probability of success at iteration  $k$ , given that the algorithm was correct until iteration  $k - 1$ .

We focus our attention on what happens at iteration  $k$ , i.e. we study the probability  $P_k(\sigma, K)$ . First, we split the set of possible elements of the solutions  $\mathcal{P}_k$  as the union of two disjoint sets:  $\mathcal{C}_k$  containing the elements that when picked by the algorithm generate a correct partial solution at iteration  $k$ , and  $\mathcal{W}$  containing the elements that when picked corrupt the partial solution. Second, we generalize the cost function used in the main optimization problem (13) to a generic set of 1D elements  $\mathcal{A}$  as,

$$g(\mathcal{A}, \hat{\mathcal{X}}_k) = \min_{p \in \mathcal{A}} \sum_{\hat{x} \in \hat{\mathcal{X}}_k} \min_{\tilde{d} \in \tilde{\mathcal{D}}} (p - \hat{x} - \tilde{d})^2. \quad (20)$$

Below, we use  $g(\mathcal{A}, \hat{\mathcal{X}}_k)$  with both sets and single elements as arguments: in other words, the expression  $g(a, \hat{\mathcal{X}}_k)$  is interpreted as  $g(\{a\}, \hat{\mathcal{X}}_k)$ .

Then, we compute the probability that the support recovery algorithm picks an element from  $\mathcal{C}_k$  instead of an element from  $\mathcal{W}$ , when searching for the solution of (13). This happens if the cost of  $\mathcal{C}_k$  is smaller than the one of  $\mathcal{W}$  measured via (20),

$$\begin{aligned} P_k(\sigma, K) &= \mathbb{P}(g(\mathcal{C}_k, \hat{\mathcal{X}}_k) < g(\mathcal{W}, \hat{\mathcal{X}}_k)) \\ &= \mathbb{P}(\exists c \in \mathcal{C}_k | g(c, \hat{\mathcal{X}}_k) < g(\mathcal{W}, \hat{\mathcal{X}}_k)) \\ &= 1 - \mathbb{P}(\nexists c \in \mathcal{C}_k | g(c, \hat{\mathcal{X}}_k) < g(\mathcal{W}, \hat{\mathcal{X}}_k)) \\ &= 1 - \mathbb{P}(\forall c \in \mathcal{C}_k | g(c, \hat{\mathcal{X}}_k) \geq g(\mathcal{W}, \hat{\mathcal{X}}_k)). \end{aligned} \quad (21)$$

We assume that the events  $g(c, \hat{\mathcal{X}}_k) \geq g(\mathcal{W}, \hat{\mathcal{X}}_k)$  are independent for all  $c \in \mathcal{C}_k$  and obtain

$$P_k(\sigma, K) = 1 - \prod_{c \in \mathcal{C}_k} \mathbb{P}\left(\frac{g(c, \hat{\mathcal{X}}_k)}{g(\mathcal{W}, \hat{\mathcal{X}}_k)} \geq 1\right). \quad (22)$$

This is a crude simplification, but it enables us to compute an approximation of  $P_k(\sigma, K)$  that will not impair the quality of the end result, as we will demonstrate later. With a similar development as (21), we can write

$$\mathbb{P}\left(\frac{g(c, \hat{\mathcal{X}}_k)}{g(\mathcal{W}, \hat{\mathcal{X}}_k)} \geq 1\right) = 1 - \mathbb{P}\left(\forall w \in \mathcal{W} \mid \frac{g(c, \hat{\mathcal{X}}_k)}{g(w, \hat{\mathcal{X}}_k)} < 1\right).$$

Again, we approximate  $P_k(\sigma, K)$  assuming the independence of the events  $g(w, \hat{\mathcal{X}}_k)$  as

$$P_k(\sigma, K) = 1 - \prod_{c \in \mathcal{C}_k} \left(1 - \prod_{w \in \mathcal{W}} \mathbb{P}\left(\frac{g(c, \hat{\mathcal{X}}_k)}{g(w, \hat{\mathcal{X}}_k)} < 1\right)\right). \quad (23)$$

Then, we focus our attention on the term  $\mathbb{P}\left(\frac{g(c, \hat{\mathcal{X}}_k)}{g(w, \hat{\mathcal{X}}_k)} < 1\right)$ . First, we compute the cost of adding an element  $c$  from  $\mathcal{C}_k$  to  $\hat{\mathcal{X}}_{k+1}$ ,

$$\begin{aligned} g(c, \hat{\mathcal{X}}_k) &= \sum_{\hat{x} \in \hat{\mathcal{X}}_k} \min_{\tilde{d} \in \tilde{\mathcal{D}}} (c - \hat{x} - \tilde{d})^2 \\ &= \sum_{\ell=1}^k \min_{\tilde{d} \in \tilde{\mathcal{D}}} (c - (x_\ell - x_1 + \nu_{\ell,1}) - \tilde{d})^2, \end{aligned} \quad (24)$$

where each  $\hat{x} \in \hat{\mathcal{X}}_k$  is a shifted noisy version of an element of  $\mathcal{X}$ . Following a similar reasoning, the newly added element  $c$  can be expressed as  $c = x_{k+1} - x_1 + \nu_{k+1,1}$ . By substituting this expression into (24), we further obtain

$$\begin{aligned} g(c, \hat{\mathcal{X}}_k) &= \sum_{\ell=1}^k \min_{\tilde{d} \in \tilde{\mathcal{D}}} (x_{k+1} + \nu_{k+1,1} - x_\ell - \nu_{\ell,1} - \tilde{d})^2 \\ &\stackrel{(a)}{\approx} \sum_{\ell=1}^k (\nu_{k+1,1} - \nu_{\ell,1} - \nu_{k+1,\ell})^2 \\ &= 3\sigma^2 Q_k^{(1)}, \end{aligned} \quad (25)$$

where  $Q_k^{(1)} \sim \chi_k^2$ , and in (a) we approximate  $g(c, \hat{\mathcal{X}}_k)$  by selecting the difference  $\tilde{d} = x_{k+1} - x_\ell + \nu_{k+1,\ell}$ . We select this specific  $\tilde{d}$  as it is likely to be picked, provided that the noise variance  $\sigma^2$  is small compared to the values  $x_i$ . This also significantly simplifies (25) by dropping the random variables  $x_1, x_{k+1}$ , and  $x_\ell$ .

Second, we analyze the cost of making an error  $g(w, \hat{\mathcal{X}}_k)$  at iteration  $k$ —that is selecting any element  $w \in \mathcal{W}$  given  $\hat{\mathcal{X}}_k$ :

$$g(w, \hat{\mathcal{X}}_k) = \sum_{\hat{x} \in \hat{\mathcal{X}}_k} \min_{\tilde{d} \in \tilde{\mathcal{D}}} (w - \hat{x} - \tilde{d})^2. \quad (26)$$

We express the minimum in (26) as an exhaustive check of all the possible selections of  $k$  differences from  $\tilde{\mathcal{D}}$ . To do so, we define  $\mathcal{M}_k$  as the set containing all the  $k$ -permutations of  $\tilde{\mathcal{D}}$ , and rewrite the probability of selecting a correct location  $c$  instead of a wrong one  $w$  for any given  $c$  and  $w$  from (23) as follows,

$$\mathbb{P}\left(\frac{g(c, \hat{\mathcal{X}}_k)}{g(w, \hat{\mathcal{X}}_k)} < 1\right) = \mathbb{P}\left(\frac{g(c, \hat{\mathcal{X}}_k)}{e(w, \boldsymbol{\pi}, \hat{\mathcal{X}}_k)} < 1, \forall \boldsymbol{\pi} \in \mathcal{M}_k\right).$$

Here, we introduced  $e(w, \boldsymbol{\pi}, \hat{\mathcal{X}}_k)$  as the cost for a given permutation  $\boldsymbol{\pi}$ ,

$$e(w, \boldsymbol{\pi}, \hat{\mathcal{X}}_k) = \sum_{i=1}^k (w - \hat{x}_i - \pi_i)^2, \quad (27)$$

where the elements in  $\boldsymbol{\pi}$  and  $\hat{\mathcal{X}}_k$  are indexed with  $i$ .



Once more, we assume that all these selections are independent to obtain

$$P\left(\frac{g(c, \hat{\mathcal{X}}_k)}{g(w, \hat{\mathcal{X}}_k)} < 1\right) = \prod_{\pi \in \mathcal{M}_k} P\left(\frac{g(c, \hat{\mathcal{X}}_k)}{e(w, \pi, \hat{\mathcal{X}}_k)} < 1\right). \quad (28)$$

Finally, we discuss the probabilistic aspects of (27). The terms  $\omega$ ,  $\hat{x}_i$  and  $\pi_i$  are each made of the difference between two points plus a noise value. Indeed, they have the form

$$p = x_i - x_j + \nu_{i,j},$$

for some specific indices  $i$  and  $j$ . Assuming that the points in  $\mathcal{X}$  are uniformly distributed between  $-0.5$  and  $0.5$ , and the noise elements  $\nu_{i,j}$  are independent and identically distributed as Gaussian random variables with zero mean and variance  $\sigma^2$  (as previously discussed in Section III), we can approximate (27) as

$$\begin{aligned} e(w, \pi, \hat{\mathcal{X}}_k) &\stackrel{(a)}{\approx} \sum_{\ell=1}^k \left( \sum_{i=1}^6 Y_i + \sum_{j=1}^3 Z_j \right)^2 \\ &\stackrel{(b)}{\approx} \sum_{\ell=1}^k \left( W + \sum_{j=1}^3 Z_j \right)^2 \\ &= \left( 3\sigma^2 + \frac{1}{2} \right) Q_k^{(2)}, \end{aligned} \quad (29)$$

where  $Q_k^{(2)} \sim \chi_{k}^2$ ,  $Y_i \sim U[-0.5, 0.5]$ ,  $Z_j \sim \mathcal{N}(0, \sigma^2)$  and  $W \sim \mathcal{N}(0, \frac{1}{2})$ . In (a), we approximate the sum by assuming independence between all the random variables and in (b) we approximate the sum of six random variables uniformly distributed on  $[-0.5, 0.5]$  with a normal random variable with variance  $\sigma^2 = \frac{1}{2}$ .

We now have all the ingredients to compute the probability of success at iteration  $k$  (23), as

$$\begin{aligned} P_k(\sigma, K) &= 1 - \prod_{c \in \mathcal{C}_k} \left( 1 - \prod_{w \in \mathcal{W}} P\left(\frac{g(c, \hat{\mathcal{X}}_k)}{g(w, \hat{\mathcal{X}}_k)} < 1\right) \right) \\ &\approx 1 - \prod_{c \in \mathcal{C}_k} \left( 1 - \prod_{w \in \mathcal{W}} \prod_{S \in \mathcal{M}_k} P\left(\frac{Q_k^{(1)}}{Q_k^{(2)}} < \frac{3\sigma^2 + \frac{1}{2}}{3\sigma^2}\right) \right) \\ &= 1 - \left( 1 - P\left(\frac{Q_k^{(1)}}{Q_k^{(2)}} < \frac{3\sigma^2 + \frac{1}{2}}{3\sigma^2}\right)^{|\mathcal{M}_k||\mathcal{W}|} \right)^{|\mathcal{C}_k|} \\ &= 1 - \left( 1 - F\left(\frac{3\sigma^2 + \frac{1}{2}}{3\sigma^2}, k, k\right)^{|\mathcal{M}_k||\mathcal{W}|} \right)^{|\mathcal{C}_k|}, \end{aligned} \quad (30)$$

where  $F(x, k_1, k_2)$  is the cumulative distribution function of an F-distribution with parameters  $k_1$  and  $k_2$ ; it can be calculated using regularized incomplete beta functions. Last, we determine the size of the sets as

$$\begin{aligned} |\mathcal{C}_k| &= K - k, \\ |\mathcal{W}| &= N - K = K^2 - 2K + 1, \\ |\mathcal{M}_k| &= N^k. \end{aligned} \quad (31)$$

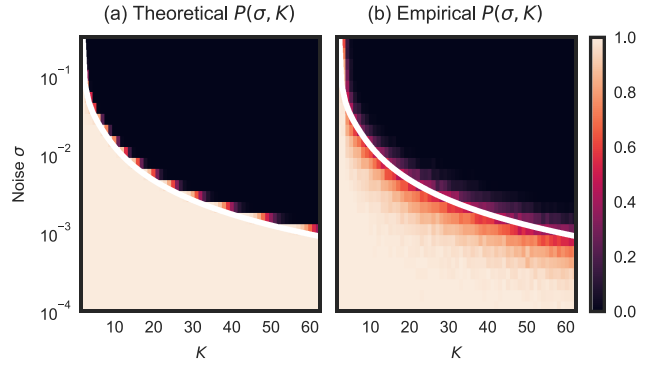


Fig. 5. Comparison of the (a) theoretical and (b) empirical probability of success for Algorithm 1 in 2 dimensions with respect to the size of the problem  $K$  and the noise  $\sigma$  affecting the set of differences. In both plots, the white line represents  $P(\sigma, K) = 0.5$ .

Note that as the number of points  $K$  increases, these exponents grow faster and push any probability that is not 1 to 0; hence, we expect a steep phase transition.

Along the path of our analysis, we made a few rough assumptions that we cannot theoretically justify regarding the independence of events, e.g. in (22), (23) and (28). While we would like to be more rigorous, we provide below numerical evidence that such assumptions hold in practice as the algorithm's performance exhibits a phase transition matching closely the derived theoretical bound (30). Swiss A

#### A. Numerical Simulations

We define the *index-based* error as a binary metric that is equal to 0 if the solution set  $\hat{\mathcal{X}}$  is of the form (7), and 1 otherwise. This error can be used to empirically measure the probability of success of Algorithm 1: we approximate it by running several experiments with different levels of noise  $\sigma$  and numbers of points  $K$ . In Fig. 5, we report the results of such an experiment and compare it with our theoretical result obtained in (30). We confirm that  $P(\sigma, K)$  exhibits a sharp phase transition—we can identify pairs  $(K, \sigma)$  for which the algorithm always succeeds and pairs for which it always fails. However, the empirical phase transition is less sharp than the theoretical one and this is probably due to our approximations regarding the independence of events. Nonetheless, the two phase transitions are closely aligned for each value of  $K$ .

In the following, we develop some intuition that may explain why these approximations appear to be so tight. We claim that, even though not *all* events are pairwise independent, *most of them* are. As an example, when we look at

$$g(p, \hat{\mathcal{X}}_k) = \sum_{\hat{x} \in \hat{\mathcal{X}}_k} \min_{\tilde{d} \in \tilde{\mathcal{D}}} (p - \hat{x} - \tilde{d})^2, \quad (32)$$

for two different values  $p_1$  and  $p_2$  of  $p$ , the respective cost functions  $g(p_1, \hat{\mathcal{X}}_k)$  and  $g(p_2, \hat{\mathcal{X}}_k)$  probably share a few common differences  $\tilde{d}$ . However, we observe that at round  $k$ , only  $k$  out of  $K^2 - 2K + 1$  differences are selected, one for every  $\hat{x} \in \hat{\mathcal{X}}_k$ . Then, assuming that most pairs  $(g(p_1, \hat{\mathcal{X}}_k), g(p_2, \hat{\mathcal{X}}_k))$

are independent is practically equivalent to assume that we select the differences  $\tilde{d}$  uniformly at random within the minimization. Moreover, we believe that the few dependent events ignored by such assumptions are one of the likely causes of the different steepness exhibited by the theoretical and observed phase transition.

## VII. IMPROVING NOISE RESILIENCE

We now discuss strategies and variations of our support recovery algorithm aiming at improving the quality of the solution in noisy settings. We chose not to include them in the analysis as they make it more intricate.

### A. Deleting Solutions From the Set of Differences

When a new point  $\hat{x}_{k+1}$  is added to  $\hat{\mathcal{X}}_k$ , Algorithm 1 ignores some useful information. Assuming that there are no collisions and no noise, we know that the values  $\hat{\mathcal{X}}_k - \hat{x}_{k+1}$  and  $\hat{x}_{k+1} - \hat{\mathcal{X}}_k$  in  $\mathcal{D}$  cannot belong to the solution  $\hat{\mathcal{X}}$  as they are part of  $\mathcal{W}$ . Thus, as soon as  $\hat{x}_{k+1}$  is added to the solution set, we can remove all values of the form  $\hat{\mathcal{X}}_k - \hat{x}_{k+1}$  and  $\hat{x}_{k+1} - \hat{\mathcal{X}}_k$  from  $\mathcal{D}$ .

The same reasoning applies to the noisy case, but we pick the closest values in  $\tilde{\mathcal{D}}$  as we do not have exact differences. More formally, when we add a new point  $\hat{x}_{k+1}$  to the solution  $\hat{\mathcal{X}}_k$ , we dispose of the following  $2k$  elements of  $\tilde{\mathcal{D}}$ ,

$$\tilde{d}^* = \operatorname{argmin}_{\tilde{d} \in \tilde{\mathcal{D}}} \|\pm \hat{x} \mp \hat{x}_{k+1} - \tilde{d}\|^2, \quad \forall \hat{x} \in \hat{\mathcal{X}}_k.$$

This approach results in two opposing effects. On one hand, we introduce the risk of erroneously discarding a point  $\tilde{d}^*$  that belongs to the solution. On the other hand, we are pruning many elements out of  $\tilde{\mathcal{D}}$  and naturally reduce the risk of picking an erroneous candidate later on in the recovery process. As we will show in Section VII-D, the benefits out-weight the risks.

### B. Symmetric Cost Function

Next, we replace the cost function (13) with a symmetric one to leverage the natural symmetry of the ACF.

In Algorithm 1, we search for the vectors in  $\tilde{\mathcal{D}}$  closest to the computed differences  $\mathbf{p} - \hat{\mathcal{X}}_k$  for each candidate  $\mathbf{p}$ . We strengthen its noise resilience by jointly searching for the vectors closest to  $\mp \hat{\mathcal{X}}_k \pm \mathbf{p}$  and choosing the candidate  $\mathbf{p}$  that minimizes the sum of both errors. Specifically, we rewrite the cost function (13) as

$$\hat{x}_{k+1} = \operatorname{argmin}_{\mathbf{p} \in \mathcal{P}_k} \sum_{\hat{x} \in \hat{\mathcal{X}}_k} \min_{\tilde{d}, \tilde{d}' \in \tilde{\mathcal{D}}} \left\| \mathbf{p} - \hat{x} - \tilde{d} \right\|^2 + \left\| \hat{x} - \mathbf{p} - \tilde{d}' \right\|^2. \quad (33)$$

We stress that this improvement is compatible with the idea of caching introduced in Section V. We can in fact cache the following pairs

$$(\tilde{d}, \tilde{d}')_{i,j} = \operatorname{argmin}_{\tilde{d}, \tilde{d}' \in \tilde{\mathcal{D}}} \|\mathbf{p}_j - \hat{x}_i - \tilde{d}\|^2 + \|\hat{x}_i - \mathbf{p}_j - \tilde{d}'\|^2, \quad (34)$$

for each  $\hat{x}_i \in \hat{\mathcal{X}}_k$  and  $\mathbf{p}_j \in \mathcal{P}_k$  and recompute them when  $\mathbf{p}_j$  gets added to the solution  $\hat{\mathcal{X}}_{k+1}$ .

### C. Denoising of Partial Solutions

At each iteration  $k$  of Algorithm 1, we have a partial solution  $\hat{\mathcal{X}}_{k+1}$  and, from (13), we identify for each pair  $\hat{x}_i, \hat{x}_j \in \hat{\mathcal{X}}_{k+1}$  a difference  $\hat{d}_{i,j}$  that is the closest to  $\hat{x}_i - \hat{x}_j$ . In other words, we are simultaneously labeling the differences  $\hat{d}_{i,j}$  using our current partial solution; such a labeling is completed as  $k$  reaches the final iteration  $K - 1$ .

This partial labeling can be exploited to *denoise* the set  $\hat{\mathcal{X}}_{k+1}$  as it provides unused additional constraints and mitigates the error propagation between the iterations. More precisely, we propose to find a set of points  $\{\hat{x}_i\}_{i=1}^{k+1}$  that minimizes the following cost function

$$J(\{\hat{x}_i\}_{i=1}^{k+1}) = \sum_{i,j} \|\hat{d}_{i,j} - (\hat{x}_i - \hat{x}_j)\|^2. \quad (35)$$

The solution to (35) is derived in closed-form by setting its first derivative to 0. As it is based on a least-square-error criterion, it is then optimal when the differences are corrupted by additive Gaussian noise.

This leads to a simple and effective strategy: refine the estimate of the solution set at each step by taking the average of the differences related to each point  $\hat{x}_i \in \hat{\mathcal{X}}_{k+1}$  as

$$\hat{x}_i = \frac{1}{k+1} \sum_{j=1}^{k+1} \hat{d}_{i,j}, \quad (36)$$

where we recompute all  $\hat{x}_i$  as they are used in the  $k+1$  iteration. To see why this works, we separate the sum as

$$\frac{1}{k+1} \sum_{j=1}^{k+1} \hat{d}_{i,j} = \mathbf{x}_i - \frac{1}{k+1} \sum_{j=1}^{k+1} \mathbf{x}_j + \frac{1}{k+1} \sum_{j=1}^{k+1} \nu_{i,j}.$$

We observe that  $-\frac{1}{k+1} \sum_{j=1}^{k+1} \mathbf{x}_j$  is the same translation for all points  $\hat{x}_i$ . The consequence of this approach is that the total noise is reduced as we average its different realizations over  $k+1$  values. Note that since Algorithm 1 assumes that  $\hat{x}_1 = \mathbf{0}$  in  $\hat{\mathcal{X}}_k$ , we also translate back all the points by  $-\hat{x}_1$  after each update.

Unfortunately, the idea of caching the differences introduced in Section V is not compatible with the denoising of the partial solutions. As at each step we modify the partial solution set  $\hat{\mathcal{X}}_k$ , the differences between  $\hat{\mathcal{X}}_k$  and  $\tilde{\mathcal{D}}$  change accordingly, which makes it impossible to cache them. Hence, there exists a hard trade-off between quality and complexity, and we should pick the right strategy depending on the requirements of each specific practical scenario.

### D. Comparison of Improvement Strategies

Last, we evaluate the significance of our proposed improvements on Algorithm 1. We quantify the results using the index-based error introduced in Section VI, as well as the  $\ell^2$  error,

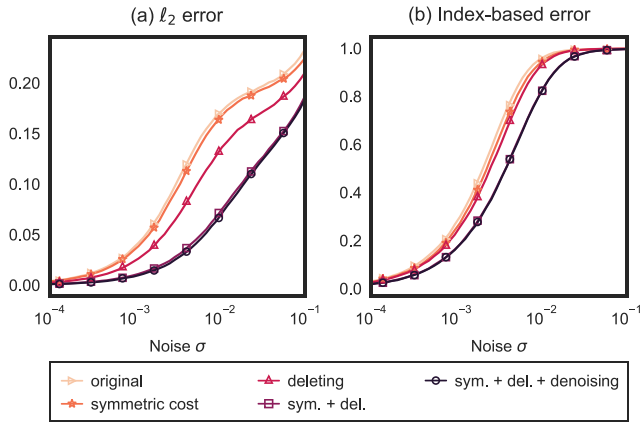


Fig. 6. Average error for the different combinations of improvements of the algorithm. We create  $\mathcal{X}$  from  $K = 6$  1D points chosen uniformly at random from the interval  $[0, 1]$ , create  $\mathcal{D}$  accordingly and add Gaussian noise  $\mathcal{N}(0, \sigma^2)$  to its elements. The  $\ell^2$  and the index-based errors are computed for different levels of noise  $\sigma$  and different improvements of the original algorithm.

TABLE I  
THE AVERAGE DECREASE IN  $\ell^2$  ERROR FOR DIFFERENT IMPROVEMENT STRATEGIES OF THE ORIGINAL ALGORITHM 1

	sym.	del.	sym. + del.	sym. + del. + denoising
$K = 6$	5%	27%	59%	62%
$K = 10$	1%	28%	48%	49%

which we define as the  $\ell^2$ -norm of the difference between the underlying points  $\mathcal{X}$  and their estimation  $\hat{\mathcal{X}}$ .<sup>6</sup>

The comparison of the different improvement strategies is illustrated in Fig. 6. In this experiment, we draw  $K = 6$  one-dimensional points uniformly at random from the interval  $[0, 1]$  and add Gaussian noise  $\mathcal{N}(0, \sigma^2)$  on their pairwise differences. We run Algorithm 1 and the proposed improvements for different noise levels  $\sigma$ . It is clear that all the proposed strategies enhance the original algorithm, with respect to both the index-based error and the  $\ell^2$  error.

Moreover, we also observe that different strategies combine constructively: for example, the symmetric cost function decreases the  $\ell^2$  error by 5% on average, while deleting solutions from the set of differences improves the results by 27% on average. When combined together, the average error decreases by 59%. Including the denoising further enhances the algorithm, as the average error decreases by 62%. Similarly, for the index-based error there is an evident shift between the phase transitions of the original algorithm with and without improvements.

In addition to the described experiment, we illustrate the different improvement strategies for  $K = 10$  1D points in Fig. 10 in Appendix. We quantify the performance of each of these enhancing methods for  $K = 6$  and  $K = 10$  in Table I. We can conclude that different strategies combine constructively and improve the performance of the original algorithm in both cases.

<sup>6</sup>This requires to first align the two sets of points  $\mathcal{X}$  and  $\hat{\mathcal{X}}$  by minimizing the  $\ell^2$ -norm between their elements, subject to any shift and/or reflection.

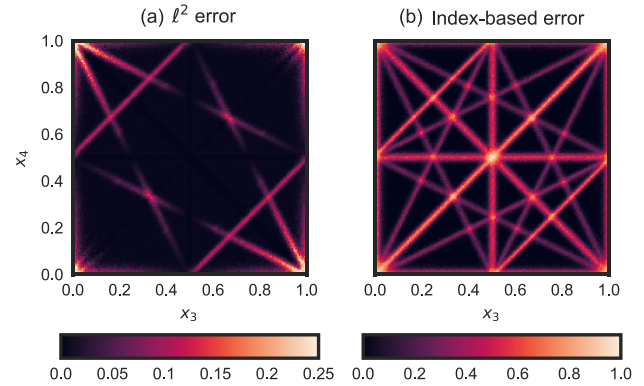


Fig. 7. Influence of the points' locations on the estimation errors. We solve a 1D instance of the problem with  $K = 4$ ,  $x_1 = 0$ , and  $x_2 = 1$ . The locations  $x_3$  and  $x_4$  vary along the  $x$ - and  $y$ -axis.

## VIII. INFLUENCE OF THE POINTS LOCATIONS

The algorithm performance around the phase transition in Fig. 5 also seems to indicate that some configurations of points are easier to recover than others. In this section, we run a small experiment to visualize the challenges posed by certain configurations.

We consider a low-complexity setup ( $K = 4$ ,  $D = 1$ ), fix the support boundaries, that is  $x_1 = 0$  and  $x_2 = 1$ , and study the reconstruction error for various pairs  $(x_3, x_4) \in [0, 1]^2$ . We generate several instances of this problem and perturb the differences in  $\mathcal{D}$  with additive Gaussian noise with zero mean and  $\sigma = 0.01$ . We measure the performance of Algorithm 1 (with all the improvements introduced in Section VII) using both the index-based and the  $\ell^2$  error. The average errors are then shown in Fig. 7, where we observe that there exist some combinations of points that lead to a significantly higher error.

We now develop intuition about a few interesting cases that emerged from the previous experiment. For the sake of simplicity, we consider a noiseless setting where collisions in the ACF or non-uniqueness of the solution are the only causes of challenging configurations.

- 1) *Collision between a difference and a point*: When a difference and a point collide, it can happen that the difference is mistaken for the point. This does not influence the  $\ell^2$  error, but causes an index-based error. An example of such a case is when  $x_3 = x_4$  (the main diagonal in Fig. 7): both the difference  $x_4 - x_3$  and  $x_1$  have value 0. As a consequence, the sets  $\mathcal{X}' = \{x_1, x_2, x_3, x_4\}$  and  $\mathcal{X}'' = \{x_4 - x_3, x_2, x_3, x_4\}$  are both equal to  $\mathcal{X} = \{0, 1, x_3, x_3\}$ , but the latter is not of the form (7).
- 2) *Constant difference 0.5*: When  $x_4 = x_3 \pm 0.5$ , we can actually find more than one set of 4 points that map to a subset of the given differences. In the case  $x_4 = x_3 + 0.5$ , the differences are  $\mathcal{D} = \pm\{0, 1, x_3, x_3 + 0.5, 1 - x_3, 0.5 - x_3, 0.5\}$ ; thus,  $\mathcal{D}$  contains all pairwise difference from both  $\mathcal{X}' = \{0, 1, x_3, x_3 + 0.5\}$  and  $\mathcal{X}'' = \{0, 1, 0.5, x_3\}$ . However,  $\mathcal{X}''$  does not lead to a zero  $\ell^2$  error.
- 3) *Collision of differences when adding a new point to the solution set*: This is for example the case of  $x_4 =$

$1 - 2x_3$  with  $\mathcal{D} = \pm\{0, 1, x_3, 1 - 2x_3, 1 - x_3, 2x_3, 1 - 3x_3\}$ . The differences 0 and 1 are always selected in the first and the second step. In the third step, we could potentially add  $2x_3$  to  $\mathcal{X}_2 = \{0, 1\}$  and reduce the set of differences to  $\mathcal{D} = \pm\{x_3, 1 - x_3, 1 - 3x_3\}$ . Next, we select  $x_3$  as a new point. We can verify that the differences of  $x_3$  and the values in  $\mathcal{X}_3 = \{0, 1, 2x_3\}$  exist in  $\mathcal{D}$ . However, in this verification we use the value  $x_3$  in  $\mathcal{D}$  twice: once as the difference between  $x_3$  and 0, and once as the difference between  $x_3$  and  $2x_3$ . The set of pairwise differences of  $\mathcal{X}_4 = \{0, 1, 2x_3, x_3\}$  is indeed contained in the original  $\mathcal{D}$ , but neither its  $\ell^2$  error nor its index-based error is zero. Notice that if we swap the third and the fourth step, this confusion would be avoided as  $x_3$  would be removed from the set of differences in the third step.

These three cases explain all the segments visible in Fig. 7. Such an analysis also applies to noisy regimes; the main difference is that we move from very localized configurations to blurrier areas where the solution is ambiguous. In fact, we introduced some noise into the experiment in Fig. 7 to enable the visualization of the *lines* identifying challenging patterns—a noiseless setting would have just led to infinitesimally thin lines. Such patterns become blurrier and wider as noise increases. These areas where reconstruction is harder also explain the not-so-sharp phase transition in Fig. 5: when drawing supports of  $K$  elements at random, the probability of hitting a challenging pattern significantly grows with the noise. To the limit, these blurred lines cover the entire domain and the probability of success is null.

## IX. COMPARISON WITH CHARGE FLIPPING

In this section, we evaluate the performance of the proposed PR algorithm in comparison with other state-of-the-art methods. Recall that our algorithm is, to the best of our knowledge, the first to operate in a noisy continuous-support setup, whereas other algorithms either assume sparse discrete signals or do not take noise into consideration. Indeed, the vast majority of PR methods are simply not designed to work with continuous supports; examples are PhaseLift [39], which recasts the PR problem as a semi-definite program, and GESPAR [12], which linearizes the PR problem with the damped Gauss-Newton method. In general, these approaches assume that the signals of interest are sparse vectors. As seen in Fig. 1, when the locations are not aligned with the sampling grid, the discretized signal contains very few—if any—nonzero entries as the scattering function spreads the sharp continuous locations.

A few algorithms are designed [20] or can be adapted [14] to work with continuous supports, but they fall short when the measured support  $\tilde{\mathcal{D}}$  is noisy. For example, TSPR [14] relies on the triangle equality between locations to recover the support; as soon as the locations are corrupted with noise, these equalities do not hold anymore.

The difference in setup and assumptions between our proposed algorithm and the state of the art makes a direct comparison basically impossible. In fact, it is not possible to adapt all the

mentioned algorithms to our generalized setup while maintaining their performance to a point that makes the comparison interesting and relevant.

One of the few algorithms that exhibited reasonable performance in this setup is the *Charge Flipping* algorithm [9], [10]: even though it was originally implemented to operate in a discrete domain, our experiments have shown that it is resilient to some noise on the ACF support.

### A. Charge Flipping

Charge Flipping is one of the reference algorithms in crystallography. It belongs to the class of dual-space algorithms as it alternatively acts on the spatial and Fourier domains, designated *real* and *reciprocal* space in crystallography. After randomly assigning a phase to the observed magnitudes of the discrete Fourier transform (DFT) coefficients, it iteratively performs the following two operations:

- 1) In the real space, it flips the sign of the values that are below some fixed threshold  $\delta$ .
- 2) In the reciprocal space, it enforces that the magnitude of the signal corresponds to the measured magnitude.

Charge Flipping directly takes as input the DFT coefficients of the ACF, while our support recovery algorithm operates on a continuous version of the ACF. This is a significant advantage of our algorithm over Charge Flipping as we do not assume that the support of the points is aligned with a grid. To have an adequate comparison between the two, we need to consider the entire pipeline, combining the three algorithms exposed in Section IV; this is illustrated in Fig. 8.

### B. Experimental Setup and Results

We generate DFT coefficients corresponding to a sparse signal as described in (1), discard their phase information, and corrupt them with zero-mean Gaussian noise. Notice that in Sections VI and VII, we assume noise on the support of the points; here, since we are evaluating the entire estimation pipeline, we are dealing with noise that is applied to the DFT coefficients instead. Obviously, these noisy DFT coefficients also lead to a noisy support of the super-resolved ACF, but it is neither Gaussian nor independent anymore. In fact, as FRI algorithms rely on nonlinear methods, the noise of its output is difficult to characterize.

The discretization of the Fourier domain is equivalent to a periodization of the spatial domain. As a consequence, the squared magnitude of the DFT coefficients corresponds to a *circular* ACF. While it is certainly possible to adapt Algorithm 1 to handle circular ACFs by testing all the possible  $2^D$  quadrants for every observed  $\tilde{\mathbf{a}} \in \tilde{\mathcal{D}}$ , we chose to zero-pad the support of  $f(\mathbf{x})$  until its ACF is not circular anymore.

Regarding Charge Flipping, we notice that its performance highly depends on the initial solution as well as the choice of  $\delta$ . To avoid giving an unfair advantage to our algorithm, we run Charge Flipping 10 times for each experiment and pick the best solution; practical experiments show that the performance gain is marginal when going above such a number of repetitions. Furthermore, best practice [10] suggests to pick  $\delta = b\theta$ , where  $b$

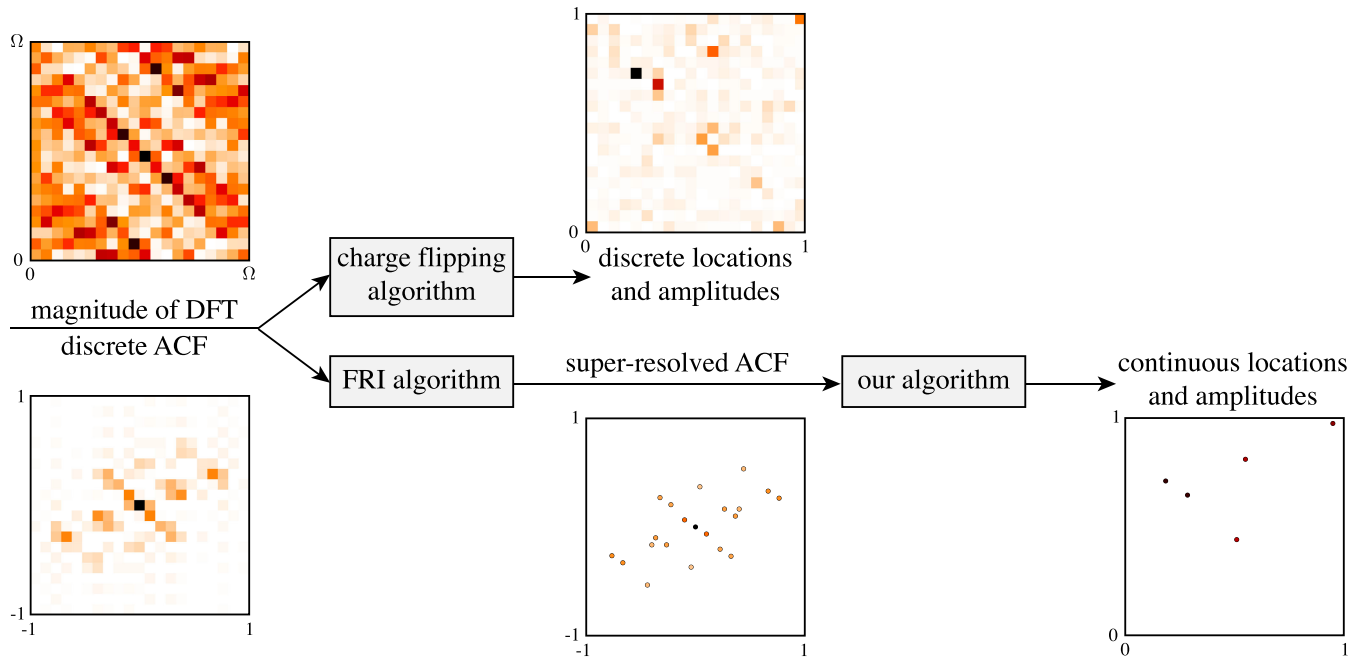


Fig. 8. PR pipeline for Charge Flipping and our algorithm. First the signal  $a(x)$  is sampled and we observe the magnitude of its DFT,  $A_m$ , which also corresponds to a discrete version of its ACF. These DFT coefficients are directly used by Charge Flipping to recover a discretized support of  $f(x)$ . Our approach proceeds in two stages: first, using FRI we compute a super-resolved version of the ACF, and then by applying the proposed algorithm, we recover the continuous version of  $f(x)$ .

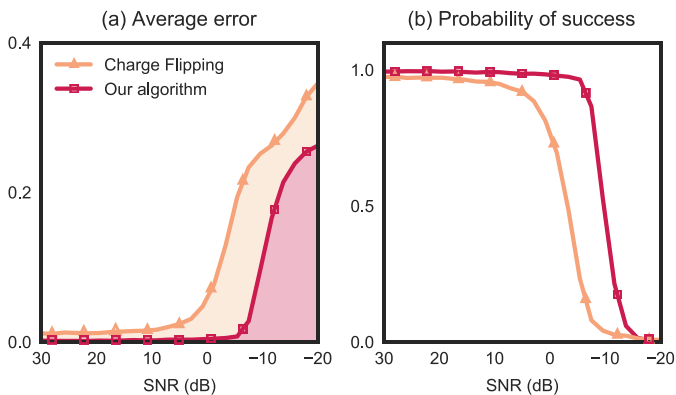


Fig. 9. Comparison of our algorithm with Charge Flipping. The performance is evaluated for  $K = 5$  1D points chosen uniformly at random from  $[0, 1]$ . The number of DFT coefficients is 200. Figure (a) shows the  $\ell^2$  reconstruction error on the locations for different values of the input SNR. Figure (b) reports the percentage of success: we consider that the algorithms fail when the resulting  $\ell^2$  error is larger than some threshold 0.04.

is a constant around 1-1.2 and  $\theta$  is the standard deviation of the measured signal. Our experiments showed that progressively decreasing the value of  $\delta$  also improves the performance of Charge Flipping. This mimics the behavior of the simulated annealing algorithm, where the temperature is steadily decreased until convergence.

Then, given noisy DFT coefficients as input, we compare the  $\ell^2$  error on the support of the points for both algorithms, as well as a probability of successfully recovering the support. To define the latter, we say that an algorithm fails when the

$\ell^2$  error is higher than a specific threshold. Fig. 9 shows that our FRI super-resolution algorithm surpasses Charge Flipping in terms of both metrics. It is not surprising that our algorithm exhibits a superior performance in a low noise regime—it even achieves exact reconstruction in the absence of noise—since it is not bound to a grid. On the other hand, Charge Flipping always suffers from approximation errors due to the implicit discretization: in the noiseless case and for a grid of size 200, we calculate that the expected  $\ell^2$  error on the support of  $K = 5$  points is about 0.0056, which is in adequacy with the baseline observed in Fig. 9a. Simulations also indicate that our algorithm outperforms Charge Flipping in high noise environments. Indeed, the reconstruction error is consistently lower and its phase transition compares favorably as well.

## X. CONCLUSION

We presented a novel approach to solve the phase retrieval problem for sparse signals. While conventional algorithms operate in discretized space and recover the support of the points on a grid, the power of FRI sampling combined with the sparsity assumption on the signal model enables to recover the support of the points in continuous space. We provided a mathematical expression that approximates the probability of success of our support recovery algorithm and confirmed our result via numerical experiments. We observed that while our algorithm runs in polynomial time with respect to the sparsity number of the signal, it remains relatively costly. To alleviate the computational costs without impacting the quality of the reconstruction, we

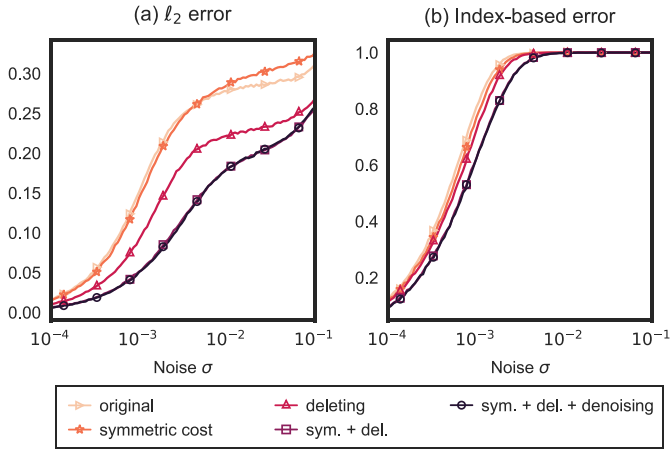


Fig. 10. Average error for the different combinations of improvements of the support recovery algorithm. We draw  $K = 10$  1D points chosen uniformly at random from the interval  $[0, 1]$  and add Gaussian noise  $\mathcal{N}(0, \sigma^2)$  to their pairwise differences. The  $\ell^2$  and the index-based errors are computed for different levels of noise  $\sigma$  and different improvements on the original algorithm.

TABLE II  
THE DECREASE IN  $\ell^2$  ERROR OF OUR ALGORITHM WITH RESPECT TO CHARGE FLIPPING AVERAGED OVER DIFFERENT INTERVALS OF THE INPUT SNR

	Input SNR, in dB				
	[30, 20)	[20, 10)	[10, 0)	[0, -10)	[-10, -20]
$K = 5$	87%	85%	88%	87%	25%
$K = 7$	25%	22%	33%	63%	41%

proposed a caching layer to avoid repeating calculations. Furthermore, we introduced several improvements that contribute to enhance the quality of estimation in the presence of noise. Finally, we showed that our super resolution PR algorithm outperforms Charge Flipping, one of the state-of-the-art algorithms, both in terms of average reconstruction error and success rate.

#### APPENDIX

To provide additional confidence in our results, we repeat the experiment from Section VII-D for a larger number of points. We draw  $K = 10$  one-dimensional points uniformly at random from the interval  $[0, 1]$  and add Gaussian noise  $\mathcal{N}(0, \sigma^2)$  on their pairwise differences. We run Algorithm 1 and the proposed improvements from Section VII for different noise levels  $\sigma$ . The comparison of the different improvement strategies is illustrated in Fig. 10 and numerically confirmed in Table I. As in Section VII-D, we conclude that the combinations of the proposed strategies significantly enhance the original algorithm, with respect to both the index-based error and the  $\ell^2$  error.

Furthermore, we present additional numerical results that confirm the superior performance of our proposed PR algorithm over the Charge Flipping algorithm. We reproduce the experimental setup from Section IX-B for  $N = 7$  points and 100 DFT coefficients, and compare the  $\ell^2$  error on the support of the points for both algorithms, as well as the probability of successfully recovering the support: Fig. 11 and Table II show that our FRI

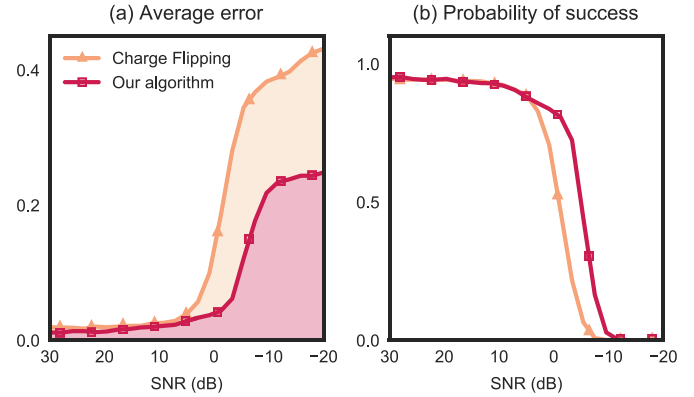


Fig. 11. Comparison of our algorithm with Charge Flipping. There are  $K = 7$  1D points chosen uniformly at random from  $[0, 1]$ . The number of DFT coefficients is 100. (a) The  $\ell^2$  reconstruction error on the locations for different values of the input SNR. (b) The percentage of success ( $\ell^2$  error being smaller than some threshold 0.04).

super-resolution algorithm surpasses Charge Flipping in terms of both metrics.

#### ACKNOWLEDGMENT

The authors would like to thank A. Scholefield for his comments and advice on the manuscript. The work was divided as follows: J. Ranieri, A. Chebira, Y. M. Lu, and M. Vetterli designed research, J. Ranieri devised the support recovery algorithm and its performance bound, G. Baechler and M. Kreković proposed algorithmic improvements and carried out experiments, and G. Baechler, M. Kreković, and J. Ranieri wrote the manuscript.

#### REFERENCES

- [1] R. P. Millane, "Phase retrieval in crystallography and optics," *J. Opt. Soc. Amer.*, vol. 7, no. 3, pp. 394–411, Mar. 1990.
- [2] K. T. Knox, "Image retrieval from astronomical speckle patterns," *J. Opt. Soc. Amer.*, vol. 66, no. 11, pp. 1236–1239, 1976.
- [3] Y. Barbotin and M. Vetterli, "Fast and robust parametric estimation of jointly sparse channels," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 2, no. 3, pp. 402–412, Sep. 2012.
- [4] A. L. Patterson, "A direct method for the determination of the components of interatomic distances in crystals," *Zeitschrift für Kristallographie*, vol. 90, pp. 517–542, 1935.
- [5] W. Cochran, "Relations between the phases of structure factors," *Acta Crystallographica*, vol. 8, no. 8, pp. 473–478, Aug. 1955.
- [6] D. Sayre, "The squaring method: A new method for phase determination," *Acta Crystallographica*, vol. 5, no. 1, pp. 60–65, Jan. 1952.
- [7] J. Karle and H. Hauptman, "The phases and magnitudes of the structure factors," *Acta Crystallographica*, vol. 3, pp. 181–187, Jul. 1950.
- [8] R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of the phase from image and diffraction plane pictures," *Optik*, vol. 35, 1972, Art. no. 237.
- [9] G. Oszlányi and A. Sütő, "Ab initio structure solution by charge flipping," *Acta Crystallographica Sect. A: Found. Crystallogr.*, vol. 60, no. 2, pp. 134–141, Mar. 2004.
- [10] G. Oszlányi and A. Sütő, "The charge flipping algorithm," *Acta Crystallographica Sect. A: Found. Crystallogr.*, vol. 64, no. 1, pp. 123–134, 2008.
- [11] S. S. Skiena and G. Sundaram, "A partial digest approach to restriction site mapping," *Bull. Math. Biol.*, vol. 56, no. 2, pp. 275–294, 1994.
- [12] Y. Shechtman, A. Beck, and Y. C. Eldar, "GESPARG: Efficient phase retrieval of sparse signals," *IEEE Trans. Signal Process.*, vol. 62, no. 4, pp. 928–938, Feb. 2014.
- [13] G. A. Croes, "A method for solving traveling-salesman problems," *Oper. Res.*, vol. 6, no. 6, pp. 791–812, 1958.

- [14] K. Jaganathan, S. Oymak, and B. Hassibi, "Sparse phase retrieval: Uniqueness guarantees and recovery algorithms," *IEEE Trans. Signal Process.*, vol. 65, no. 9, pp. 2402–2410, May 2017.
- [15] T. Dakić, *On the Turnpike Problem*. Burnaby, BC, Canada: Simon Fraser Univ., 2000.
- [16] S. S. Skiena, W. D. Smith, and P. Lemke, "Reconstructing sets from interpoint distances," in *Proc. 6th Annu. Symp. Comput. Geom.*, 1990, pp. 332–339.
- [17] X. Li and V. Voroninski, "Sparse signal recovery from quadratic measurements via convex programming," *SIAM J. Math. Anal.*, vol. 45, no. 5, pp. 3019–3033, 2013.
- [18] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Commun. Pure Appl. Math.*, vol. 66, no. 8, pp. 1241–1274, 2013.
- [19] Y. C. Eldar, P. Sidorenko, D. G. Mixon, S. Barel, and O. Cohen, "Sparse phase retrieval from short-time Fourier measurements," *IEEE Signal Process. Lett.*, vol. 22, no. 5, pp. 638–642, May 2015.
- [20] R. Beinert and G. Plonka, "Sparse phase retrieval of one-dimensional signals by Prony's method," *Frontiers Appl. Math. Statist.*, vol. 3, 2017, Art. no. 5.
- [21] R. Beinert and G. Plonka, "Ambiguities in one-dimensional phase retrieval of structured functions," *Proc. Appl. Math. Mech.*, vol. 15, no. 1, pp. 653–654, 2015.
- [22] M. Vetterli, J. Kovačević, and V. K. Goyal, *Foundations of Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2014.
- [23] I. Maravić and M. Vetterli, "Exact sampling results for some classes of parametric non-bandlimited 2-D signals," *IEEE Trans. Signal Process.*, vol. 52, no. 1, pp. 175–189, Jan. 2004.
- [24] P. L. Dragotti and F. Homann, "Sampling signals with finite rate of innovation in the presence of noise," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 2941–2944.
- [25] Y. Barbotin, "Parametric estimation of sparse channels theory and applications," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2014.
- [26] T. Blu, P. Dragotti, M. Vetterli, P. Marziliano, and L. Coulot, "Sparse sampling of signal innovations," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 31–40, Mar. 2008.
- [27] H. Cramér, *Mathematical Methods of Statistics* (Series Almqvist & Wiksells Akademiska Handböcker). Princeton, NJ, USA: Princeton Univ. Press, 1946.
- [28] E. Barankin, "Locally best unbiased estimates," *Ann. Math. Statist.*, vol. 20, pp. 477–501, 1949.
- [29] J. Ranieri, A. Chebira, Y. M. Lu, and M. Vetterli, "Phase retrieval for sparse signals: Uniqueness conditions," 2013, arXiv:1308.3058.
- [30] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1417–1428, Jun. 2002.
- [31] P. L. Dragotti, M. Vetterli, and T. Blu, "Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1741–1757, May 2007.
- [32] H. Pan, T. Blu, and M. Vetterli, "Efficient multidimensional diracs estimation with linear sample complexity," *IEEE Trans. Signal Process.*, vol. 66, no. 17, pp. 4642–4656, Sep. 2018.
- [33] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16, no. 6, pp. 22–38, Nov. 1999.
- [34] M. Unser and T. Blu, "Cardinal exponential splines: Part I—Theory and filtering algorithms," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1425–1438, Apr. 2005.
- [35] J. A. Urigen, T. Blu, and P. L. Dragotti, "FRI sampling with arbitrary kernels," *IEEE Trans. Signal Process.*, vol. 61, no. 21, pp. 5310–5323, Nov. 2013.
- [36] G. de Prony, "Essai expérimental et analytique sur les lois de la dilatabilité des fluides élastiques et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alcool à différentes températures," *J. de l'Ecole Polytechnique*, vol. 1, pp. 24–76, 1795.
- [37] P. Stoica and R. Moses, *Spectral Analysis of Signals*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2005.
- [38] H. Pan, T. Blu, and M. Vetterli, "Towards generalized FRI sampling with an application to source resolution in radioastronomy," *IEEE Trans. Signal Process.*, vol. 65, no. 4, pp. 821–835, Feb. 2017.
- [39] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM Rev.*, vol. 57, no. 2, pp. 225–251, 2015.



**Gilles Baechler** (S'13) was born in Fribourg, Switzerland, in 1986. He received the B.Sc. and M.Sc. degrees from EPFL, Lausanne, Switzerland, in 2009 and 2012, respectively, and the Ph.D. degree in computer and communication sciences, under the supervision of Prof. Martin Vetterli and Dr. Adam Scholefield, from the Audiovisual Communications Laboratory (LCAV), EPFL, Lausanne, Switzerland, in September 2018. During his bachelor's, he did an exchange with the Electrical and Computer Engineering Department, Carnegie Mellon University, Pittsburgh, PA, USA. During his master's, he did a six-month internship with the Cryptography and Security group, Bell Labs, Murray Hill, NJ, USA. He did his master's thesis with Qualcomm, San Diego, CA, USA, in the Office of the Chief Scientist, where he worked on finite rate of innovation (FRI) sampling. In 2013, he worked for a year at Morphean, a small startup active in the videosurveillance world. During the Fall 2017, he did an internship as a Software Engineer with Google Zurich. He is currently working with Google AI, Zürich, Switzerland, where he develops machine learning solutions for mobile platforms. His research interests include sparse sampling, image processing, and machine learning.



**Miranda Kreković** (S'15) received the B.Sc. degree in computing, information processing and multimedia systems from the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, in 2012 and the master's (*summa cum laude*) degree in information and communication technology from the University of Zagreb, Zagreb, Croatia, in 2014. She is currently working toward the Ph.D. degree with Audiovisual Communications Laboratory (LCAV), EPFL, Lausanne, Switzerland, under the supervision of Prof. Martin Vetterli and Prof. Ivan Dokmanić. From 2010 to 2014, she was a receiver of the Dean's Award "Josip Lončar" for outstanding performance awarded to the top 1% of all students. Her present research mainly addresses simultaneous localization and mapping (SLAM) for an echolocating robot in indoor environments. She studies the question of uniqueness of the room geometry reconstruction from acoustic impulse responses and works on computational tools and algorithms for acoustic SLAM. In fall 2016, she did a research internship with Google Mountain View with Sound Understanding team, where she worked on the blind source separation for simultaneous speech. In summer 2018, she did an internship with Google Zürich with the VideoAds Quality team. Apart from the Ph.D. work, her big passion is to promote computer science and computational thinking to children and teenagers. As a co-founder of GirlsCoding.org, she organizes hands-on workshops to motivate girls at the early age to embrace technologies and a possibility of having a career in STEM fields, gives talks to parents and children on related topics, and creates a content for programming courses in private schools. In 2018, she was a finalist of Women Technmakers Scholars Program.

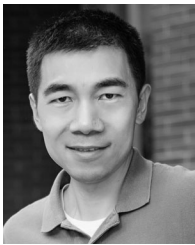


**Juri Ranieri** received the M.S. and B.S. degrees in electronic engineering from the Università di Bologna, Bologna, Italy, in 2009 and 2007, respectively, and the Ph.D. degree, under the supervision of Prof. Martin Vetterli and Dr. Amina Chebira, from the Audiovisual Communications Laboratory (LCAV), EPFL, Lausanne, Switzerland, in 2014. From July to December 2009, he was a visiting student with the Audiovisual Communications Laboratory (LCAV), École Polytechnique Fédérale de Lausanne (EPFL). From January to August 2010, he was with IBM Zurich to investigate the lithographic process as a signal processing problem. From September 2010 to September 2014, he was with the doctoral school, EPFL. From April to July 2013, he was an intern with Lyric Labs of Analog Devices, Cambridge, MA, USA. He is currently working with Google Zurich, Zurich, Switzerland, on abuse detection and security. His main research interests are inverse problems, sensor placement, and sparse phase retrieval.



**Amina Chebira** received the master's degree in communication systems from EPFL, Lausanne, Switzerland, the master's degree in mathematics from Paris 7 University, Paris, France, and the Ph.D. degree in biomedical engineering from Carnegie Mellon University, Pittsburgh, PA, USA. She is currently a Senior Manager for Data Sciences with ELCA Informatique SA. Previously, she was with the Swiss Center for Electronics and Microtechnology as a Project Manager in the Embedded Vision Systems Group, and was also the head of Vision Systems Research for CSEM.

Prior to that, she was with the Audiovisual Communications Laboratory, EPFL as a Postdoctoral Researcher. Her interests lie in information extraction from data, signal and image processing, and multiresolution analysis.



**Yue M. Lu** (S'04–M'07–SM'12) received the undergraduate degree from Shanghai Jiao Tong University, Shanghai, China, the M.Sc. degree in mathematics and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, both in 2007. Following his postdoctoral work with the Audiovisual Communications Laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, he joined Harvard University, where he is Gordon McKay Professor of Electrical Engineering and of Applied Mathematics

at the John A. Paulson School of Engineering and Applied Sciences. His research interests include theoretical and algorithmic aspects of high-dimensional statistical signal and information processing. He received the Most Innovative Paper Award of IEEE International Conference on Image Processing (ICIP) in 2006, the Best Student Paper Award of IEEE ICIP in 2007, and the Best Student Presentation Award at the 31st SIAM SEAS Conference in 2007. Student papers supervised and coauthored by him won the Best Student Paper Award of IEEE International Conference on Acoustics, Speech and Signal Processing in 2011, the Best Student Paper Award of IEEE Global Conference on Signal and Information Processing (GlobalSIP) in 2014, and the Best Student Paper (First Prize) of the IEEE CAMSAP Workshop in 2017. He was the recipient of the 2015 ECE Illinois Young Alumni Achievement Award. He is currently a member of the Signal Processing Theory and Methods Technical Committee, the Machine Learning for Signal Processing Technical Committee, and the Big Data Special Interest Group (SIG) of the IEEE Signal Processing Society. He was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He was a member of the IEEE Image, Video, and Multidimensional Signal Processing Technical Committee and an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING.



**Martin Vetterli** (S'86–M'86–SM'90–F'95) received the Dipl. El.-Ing. degree from ETH Zurich (ETHZ), Zurich, Switzerland, in 1981, the M.S. degree from Stanford University, Stanford, CA, USA, in 1982, and the Doctoratès Sciences degree from EPF Lausanne (EPFL), Lausanne, Switzerland, in 1986. He was a Research Assistant at Stanford and EPFL, and has worked for Siemens and AT&T Bell Laboratories. In 1986, he joined Columbia University in New York, where he was an Associate Professor of Electrical Engineering and co-director of the Image and Advanced Television Laboratory. In 1993, he joined the University of California at Berkeley, where he was a Professor with the Department of Electrical Engineering and Computer Sciences until 1997, and has held an Adjunct Professor position until June 2010. Since 1995, he has been a Professor of Communication Systems with EPF Lausanne, Switzerland, where he chaired the Communications Systems Division (1996/97), and heads the Audiovisual Communications Laboratory. From 2001 to 2004, he directed the National Competence Center in Research on mobile information and communication systems. He also was a Vice-President with EPFL from October 2004 to February 2011 in charge, among others, of international affairs and computing services. He has held visiting positions at ETHZ (1990) and Stanford (1998). From March 2011 to 2012, he was the Dean of the School of Computer and Communication Sciences of EPFL. From 2013 to 2016, he led the Swiss National Science Foundation and since 2017 he is President of EPFL. His research interests include sampling, wavelets, multirate signal processing, computational complexity, signal processing for communications, digital image/video processing, joint source/channel coding, signal processing for sensor networks and inverse problems like acoustic tomography. He is a fellow of ACM, a fellow of EURASIP, and a member of SIAM and NAE. He is on the editorial boards for *Applied and Computational Harmonic Analysis*, the *Journal of Fourier Analysis and Application*, and the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING and has been elected Foreign Member of the NAE in 2015. He received the Best Paper Award of EURASIP in 1984, the Research Prize of the Brown Boverly Corporation (Switzerland) in 1986, the IEEE Signal Processing Society's Senior Paper Awards in 1991, in 1996 and in 2006 (for papers with D. LeGall, K. Ramchandran, and Marziliano and Blu, respectively). He won the Swiss National Latsis Prize in 1996, the SPIE Presidential award in 1999, the IEEE Signal Processing Technical Achievement Award in 2001, the IEEE Signal Processing Society Award in 2010 for fundamental contributions to signal processing theory, technology and education, and the IEEE Jack S. Kilby Signal Processing Medal in 2017. He is an ISI highly cited researcher in engineering. He was a member of the Swiss Council on Science and Technology from 2000 to 2003. He was a plenary speaker at various conferences (e.g., IEEE ICIP, ICASSP, ISIT) and is the coauthor of three books with J. Kovacevic, *Wavelets and Subband Coding*, 1995, with P. Prandoni *Signal Processing for Communications*, 2008 and with J. Kovacevic and V.K. Goyal, *Foundations of Signal Processing*, 2015. M. Vetterli has authored or coauthored more than 190 journal papers on a variety of topics in signal/image processing and communications and holds more than 50 patents and patent applications.