Signal Processing 215 (2024) 109267

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Bivariate retrieval from intensity of cross-correlation

Dan Rosen [*], Michael B. Wakin

Colorado School of Mines, Department of Electrical Engineering, 1500 Illinois St., Golden, CO, 80401, USA

ARTICLE INFO

Keywords:
Low-rank phase retrieval
Pulse characterization
Ultra-fast optics
Frequency-resolved optical gating (FROG)
Wirtinger descent
Iterative hard tensor thresholding

ABSTRACT

Pulse characterization in ultra-fast optics presents a powerful motivation to study phase retrieval problems of high order. Frequency- and time-resolved techniques for pulse characterization both construct measurements that depend on the intensity of the cross-correlation between two unknown signals undergoing known modulations. The problem of recovering these signals has been traditionally studied and solved with alternating minimization, but recently Wirtinger gradient techniques were demonstrated to invert frequency-resolved measurements on a symmetric signal pair (Pinilla et al., 2019). In this paper, we construct a generalized Wirtinger gradient and Hessian to solve a wide breadth of problems including signal recovery from time- and frequency-resolved measurements. We further demonstrate that both measurement paradigms are special cases of low-rank phase retrieval but with a special structure that disrupts spectral initializers. To combat this problem, we present a tensor-based iterative hard thresholding initializer that, when paired with a Wirtinger gradient descent, is capable of recovering unknown signals with fewer measurements than matrix-based alternating minimization or spectral initialization methods. Finally, we employ Wirtinger gradient descent to recover signals from real-world DSCAN (Wilhelm et al., 2021) measurements and compare results with the existing state-of-the-art.

1. Introduction

Recovering information from measurements that discard phase information is a broadly studied topic that unites algorithms from the optimization community with powerful motivations from optics. Phase retrieval in optics is widely employed to recover crystal structure and unit cell projections in crystallography [1] and images in ptychographic and coded diffraction imaging [2].

Physical nuances in non-linear ultra-fast optics have created the need to recover information from more complicated and intricate interactions between light and material—and even between light and itself. Specifically, laser pulse characterization [3] defies classical formulations of phase retrieval by recovering unknown complex signals from phaseless *quadratic* measurements rather than phaseless *linear* measurements. Pulse characterization presents a unique challenge because at the incredibly brief time-duration of the pulses involved, direct in-phase and quadrature (IQ)-style heterodyne sampling is not possible. Instead, measurements must be made in an integrated fashion, where the pulse interacts with sensors that integrate intensity over time-scales much larger than the duration of a laser-pulse. This loss of both phase and time information motivates the use of a probing function to temporally encode the pulse under test. The probing function is typically another laser pulse, either one that is shorter in duration than or a copy of the pulse under test. Depending on the specific optical setup, linear optical elements are used for pulse modulation in spatial and temporal domains, while non-linear optics are used to multiply pulses by other pulses or enact a non-linear function on a single pulse. Laser pulse characterization produces a series of measurements wherein one or two unknown vectors multiply with modulation functions and each other before they are integrated in intensity. The reader is referred to [4] for a complete treatment on ultra-fast optics.

Recovering pulses from phaseless measurements of their interaction can be formulated as a higher-order phase retrieval problem [5]. The matrix formed by the outer product of two signals contains all unique products involved in cross-correlation of two signals, and we note that optical pulse characterization measurements can be viewed as phaseless quadratic measurements of this rank-one outer product matrix. This raises the possibility that algorithmic techniques for low-rank phase retrieval [6] may bring new approaches to solve the pulse characterization problem. Among these, techniques like anchored regression [7] have been demonstrated effective in low-rank phase retrieval and could present a convex optimization approach to pulse characterization.

Meanwhile, many algorithmic techniques to solve the pulse characterization problem exist. Frequency-resolved optical gating (FROG) [8] remains one of the most commonly studied measurement paradigms for pulse characterization in the optics and optimization communities

[*] Corresponding author.
E-mail address: drosen@mines.edu (D. Rosen).

https://doi.org/10.1016/j.sigpro.2023.109267
Received 21 March 2023; Received in revised form 15 August 2023; Accepted 20 September 2023
Available online 23 September 2023

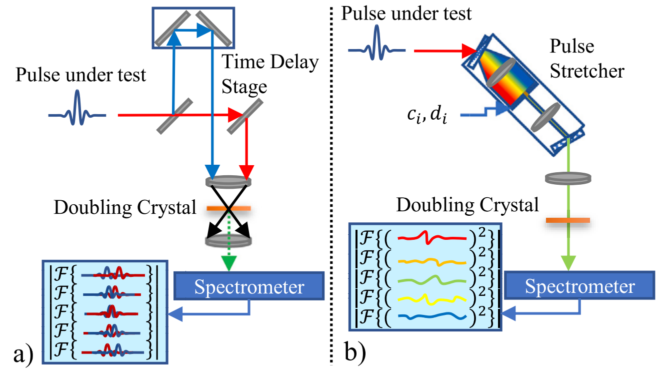

Fig. 1. Comparison of frequency- and time-resolved pulse characterization systems. (a) Simplified setup of a frequency-resolved optical gating (FROG) system. Two pulses, or a single pulse that is split, are physically time-delayed with respect to one another with the use of a physical delay path. The two pulses are brought together in a non-linear medium, where their product is produced and measured at a spectrometer. (b) Simplified setup of a time-resolved pulse characterization system. A pulse enters a pulse-shaper [15], which modulates the pulse in the wavelength domain. The pulse is then squared in the time domain and measured in the wavelength domain at the spectrometer. When properly converted from wavelength to frequency, the intensity of cross-correlation of the pulse's modulated frequency content is measured at the spectrometer.

alike. Recovery techniques from FROG measurements include various alternating minimization algorithms [8–11] and gradient descent methods [12]. Alternating minimization techniques typically rely on alternating between a forward model that imposes magnitude constraints and a gating constraint that ensures the forward model acts only on a rank-one matrix projection [9]. Gradient approaches typically seek to minimize a least-squares cost function with descent following a Wirtinger gradient [12].

Recently developed time-resolved pulse characterization systems [13,14] use less physically complicated optical setups for collecting measurements compared to frequency-resolved systems. Whereas FROG systems typically measure intensity of cross-correlation using a physical translation stage to move one pulse in time against another, time-resolved systems use non-linear effects in a transform domain to compute the intensity of cross-correlation for modulated spectra. Mathematically, the two approaches are very similar, with the difference being that the time-resolved systems aim to recover the complex spectrum of a pulse, while frequency-resolved systems aim to recover its complex time-domain profile. Frequency- and time-resolved systems are compared in Fig. 1. Generally speaking, time-resolved techniques are advantageous because of their simplified optical setups and the control they offer over spectral modulation. This is expanded on in Section 2.

The similarities between classical measurement techniques like FROG and more recent time-resolved techniques motivate a broader definition of the pulse characterization problem. In this paper, we formulate the general problem of bivariate recovery from intensity of cross-correlation (ICC), and we present its representations both as a low-rank phase retrieval problem and as a low-rank tensor recovery problem. We demonstrate how these formulations motivate new algorithms and initializers. We present preliminary work toward understanding theoretical convergence properties of the intensity cost function, as well as an examination of the computational and memory complexity of several approaches. Finally, we demonstrate the effectiveness of these algorithms on recovery from real measurements and compare results against a recent inversion algorithm from the optics community [16].

In Section 2 we present the ICC problem and its matrix and tensor formulations. In Section 3, we detail a least-squares cost function, its gradient, and its Hessian. In Section 4, we discuss a limitation of spectral approaches for solving the ICC problem and demonstrate several effective alternating minimization approaches. In Section 5.1, we provide numerical experiments that demonstrate the efficacy and complexity of gradient and tensor approaches to this problem as well as numerical investigations into cost function behavior and convergence. Finally, in Section 6, we demonstrate the performance of algorithms in this paper on data taken from a physical system.

Notation: Many equations in this paper involve the manipulation of complex-valued vectors and matrices. The superscript * represents elementwise complex conjugation of a scalar, vector, or matrix. The superscript ^{H} represents the conjugate transpose of a vector or matrix, while the superscript ^{T} represents the non-conjugate transpose of a vector or matrix. Vectors bearing ^{T} or ^{H} are interpreted as row vectors. The notation $x \sim C(\mathbf{0}, \mathbb{I}^N)$ represents a complex Gaussian random vector with zero mean and identity covariance in dimension N.

2. Background and problem definition

2.1. Intensity of cross-correlation

We consider the recovery of two vectors $x_0 \in \mathbb{C}^M$ and $y_0 \in \mathbb{C}^N$ from a series of I phaseless cross-correlation intensity measurements of the form

$$\begin{aligned} h_i[k] &= \left| c_i x_0 \bigotimes_k d_i y_0 \right|^2 + \eta_i[k] \qquad (1) \\ &= \left| \sum_{n=\max(0,-k)}^{\min(M,N-k)-1} c_i[n] d_i^*[n+k] x_0[n] y_0^*[n+k] \right|^2 + \eta_i[k], \\ & i \in \{0, 1, \dots, I-1\}, \\ & k \in \{-M+1, -M+2, \dots, N-1\}. \end{aligned}$$

Here, $\{c_i\}_{i=0}^{I-1} \subset \mathbb{C}^M$ and $\{d_i\}_{i=0}^{I-1} \subset \mathbb{C}^N$ are complex modulation vectors that are point-wise multiplied with the target vectors x_0 and y_0. A second indexing variable k represents the lag shift in the cross-correlation measurement. Presented vectors and sums are zero-based. Real-valued noise is denoted by $\eta_i[k]$. We define signal-to-noise ratio (SNR) throughout this paper as

$$\mathrm{SNR} = \frac{\sum_{i,k} |h_i[k]|_2^2}{\sum_{i,k} |\eta_i[k]|_2^2}. \qquad (2)$$

2.2. Blind FROG

Among several existing FROG measurement paradigms, one that matches the definition of ICC presented in (1) is blind FROG [5]. In our notation, the measurements of pulses $x_0 \in \mathbb{C}^M$ and $y_0 \in \mathbb{C}^N$ in blind FROG can be expressed as

$$h_i[k] = \left| \sum_{n=\max(0,-k)}^{\min(M,N-k)-1} e^{-2\pi jin/M} x_0[n] y_0^*[n+k] \right|^2 + \eta_i[k]. \qquad (3)$$

We see that (3) is an instance of (1) with $c_i[n] = e^{-2\pi jin/M}$ and $d_i[n] = 1$.

An aspect of FROG that was explored in [5] was that the lag k may be larger in time than the spacing between samples of x_0 and y_0, which lowers the non-ambiguous bandwidth in the recovery of x_0 and y_0. This treatment is largely left out of this paper, where we focus primarily on time-resolved approaches. In time-resolved approaches, the lag variable is controlled by sample density in a spectrometer rather than the coarser physical spacing of a delay line.

2.3. Time-resolved techniques

Measurement systems such as SPARC [14] and DSCAN [16] belong to the family of time-resolved pulse characterization techniques. In stark contrast with FROG, where the delay in ICC measurements is produced by physically translating the path of one pulse with respect to the other, time-resolved techniques compute ICC measurements by modulating and measuring in the spectral domain. This difference is illustrated in Fig. 1. A key advantage of time-resolved systems is the ability to control the modulation applied in the spectral domain by the use of a mask. Masks are complex-valued and can modulate both the amplitude and the phase of the pulse at various wavelengths. Upon returning to the time domain at the exit of the compressor, the signal is passed through a doubling crystal. The crystal acts to square (without conjugation) the pulse, rather than multiply it by another pulse. This results in the measurement (at the spectrometer) of the intensity of the cross-convolution of the modulated frequency spectrum of the pulse with itself, rather than a measurement of the intensity of cross-correlation.

In summary, time-resolved techniques take the form of the ICC measurements presented in (1) where x_0 represents the spectrum of the pulse to be measured, $y_0 = x_0^*[::-1]$, where $[::-1]$ represents time-reversal, and for all i, $d_i = c_i^*[::-1]$.

DSCAN [16] provides a minimalist system for time-resolved pulse characterization. DSCAN opts to move the position of the second diffraction grating in the Martinez pulse compressor to impart a phase profile on the spectrum of the pulse. Subsequent modulations are achieved at unique locations of the grating. With DSCAN, the modulation $c_i[n]$ is given by

$$c_i[n] = \exp\left(j\frac{nL[i]}{c}\sqrt{1-\left(\frac{2\pi c}{nd}-\sin(\theta[i])\right)^2}\right). \tag{4}$$

Here, $L[i]$ denotes the position (positive or negative) of the second grating of the pulse compressor away from the nominal four focal length separation, $\theta[i]$ denotes the incident angle of the compressor's first grating [16] typically held at the Littrow angle of the center-wavelength, and d represents the density of the diffraction grating used. We will return to this measurement paradigm in Section 6 when we present experiments using real data.

The generality of (1) and the adaptability of the algorithms presented here to various frequency- and time-resolved pulse measurement systems suggest that we are not limited to using existing pulse correlation and convolution systems. The requirements on the forward model (i.e., properties of $\{c_i\}$, $\{d_i\}$, and measurement count (I)) to guarantee recovery are topics of future research.

2.4. Extension from existing phase retrieval problems

This problem extends naturally from that of linear phase retrieval where a complex vector $x \in \mathbb{C}^N$ is recovered from phaseless linear measurements $y = |Ax|^2$, $A \in \mathbb{C}^{M\times N}$. This problem has been studied for the better part of the last half-century and has yielded convex and non-convex techniques in a multitude of varieties [17–25].

The approaches that are studied in this paper draw direct analog to three phase-retrieval algorithms applied to the higher-order phase retrieval problem raised by pulse characterization. The first of these algorithms is the Gerchberg–Saxtonalgorithm [18] which iteratively constrains magnitude of Fourier coefficients against a transform domain constraint. Applied to pulse characterization, the most common alternating minimization method is the PCGPA algorithm [10] that we generalize to two unknown pulses in this paper. The second is Wirtinger Flow and its variants [19,26], which is typically an initialization procedure followed by a gradient descent cost minimization. This paper examines several initializations followed by a gradient descent over intensity cost. The final extension is from PhaseLift [21], a lifted technique that minimizes the trace of a larger "lifted" linear matrix recovery problem rather than solve a non-linear vector recovery problem. In this paper we expand this concept to a linear tensor rank minimization problem lifted from a higher-order low-rank phase retrieval problem.

Current state-of-the-art in phase retrieval is dominated largely by alternating minimization approaches like Gerchberg–Saxtonand gradient descent techniques. Recent years have seen extensive study into modifications and improvements of both forms of these algorithms, with particular attention from the optimization community directed toward improved gradient techniques. In particular, advances in initializers, transitioning to amplitude based cost functions, and the use of reweighting and truncation per iteration techniques have demonstrated significant practical use [22,23,25]. A brief examination of an amplitude based approach is included in our numerical results.

Expanded capabilities in phase retrieval led to the more recent measurement paradigm of low-rank phase retrieval [6,7,27]. Low-rank phase retrieval examines the recovery of a matrix X from a series of phaseless matrix inner products of the form $y_i = |\langle A_i, X\rangle|^2$. This problem is typically motivated by simultaneously solving for a series of ordinary vector-based phase retrieval problems of the form $y_j = A_j x_j$ with the caveat that the x_j vary only in some low-rank fashion when concatenated into a matrix $X = [x_1, x_2, \ldots]$ [6]. This formulation allows generalizes measurement matrices A that can extend over multiple x_j, and this joint information can effectively recover the matrix X and thereby $[x_1, x_2, \ldots]$. Low-rank phase retrieval is also motivated in blind deconvolution problems [7] where pairs of signals are recovered from the Fourier magnitude of their convolution. The technique we share in this paper can be viewed as a low-rank phase retrieval problem with distinct challenges which will be reviewed in Section 4.1, and is distinct from these previous low-rank phase retrieval problems largely due to the structure of the measurement matrices A_i used in ICC.

2.5. Problem formulations

At this point, our analysis will be made considerably easier and the motivation for our algorithms will be made more clear by rewriting the problem in (1) in several forms. These formulations will share some common notations:

- $l[k] = \max(0, -k)$ and $u[k] = \min(M, N-k) - 1$ are the lower and upper limits, respectively, for the index of summation n in the cross-correlation at k lag appearing in (1).
- For a matrix C, $\mathrm{diag}(C,\ k)$ is a matrix of all zeros, except the kth diagonal which is filled with the kth diagonal of C.
- $\Lambda_i[k] = \mathrm{diag}(c_i d_i^H,\ k)$ represents an $M \times N$ matrix whose values are all zero except for the kth diagonal, which is filled with the lagged product of $c_i[n]d_i^*[n+k]$.
- $\mathcal{L}_i^k = \Lambda_i[k]^* \otimes \Lambda_i[k]$ is the fourth order tensor whose entries are zero except where filled as $\mathcal{L}_i^k[q,r,s,t] = \mathrm{diag}(c_i d_i^H,\ k)[q,r]^*\,\mathrm{diag}(c_i d_i^H,\ k)[s,t]$.
- $\mathcal{X}_0 = x_0 \otimes y_0^* \otimes x_0^* \otimes y_0$ is the fourth order outer product tensor with non-zero entries indexed as $\mathcal{X}[q,r,s,t] = x_0[q]y_0^*[r]x_0^*[s]y_0[t]$.

2.5.1. Quartic bivariate vector formulation

The first formulation we construct is straightforward from (1). The ICC measurements (1) can be expressed as a quartic function of the vectors x_0 and y_0:

$$\begin{aligned} h_i[k] &= \left|\sum_{n=l[k]}^{u[k]} c_i[n]d_i^*[n+k]x_0[n]y_0^*[n+k]\right|^2 + \eta_i[k] \\ &= |x_0^T\,\mathrm{diag}(c_i d_i^H,\ k)y_0^*|^2 + \eta_i[k] \\ &= |x_0^T \Lambda_i[k] y_0^*|^2 + \eta_i[k] \\ &= x_0^T \Lambda_i[k] y_0^* x_0^H \Lambda_i^*[k] y_0 + \eta_i[k]. \end{aligned} \tag{5}$$

This formulation will be useful for the construction of a Wirtinger gradient in Section 3.

2.5.2. Quadratic low-rank matrix formulation

Our second formulation highlights the fact that the ICC measurements (1) can also be expressed as phaseless quadratic measurements of the rank-one matrix $x_0 y_0^H$ formed by the outer product of the two unknown vectors x_0 and y_0:

$$\begin{aligned} h_i[k] &= |x_0^T \Lambda_i[k] y_0^*|^2 + \eta_i[k] \\ &= |\langle \Lambda_i^*[k],\ x_0 y_0^H \rangle|^2 + \eta_i[k]. \end{aligned} \tag{6}$$

Here, $\langle A,\ B\rangle = \mathrm{trace}(B^H A)$ represents the Frobenius inner product between matrices A and B.

This formulation reveals that ICC falls into the category of low-rank phase retrieval problems [6]. Interestingly, in ICC the low-rank (in fact, rank-one) structure of $x_0 y_0^H$ arises due to the lagged quadratic relationship between measurements and variables; in other settings [6], low-rank structure connects phaseless measurements spanning a time period over which a one-dimensional objective signal varies. An important distinction in ICC is that matrices $\Lambda_i^*[k] = \mathrm{diag}(c_i d_i^H,\ k)^*$ against which $x_0 y_0^H$ is measured in (6) are entry-wise sparse and disjoint across different k. This structure differs from the global, often low-rank measurement matrices that typically appear in the low-rank phase retrieval literature [6,7]. This also has substantial implications for the efficacy of spectral initializers as we discuss in Section 4.2.

2.5.3. Linear low-rank tensor formulation

Finally, it is important to note that the ICC measurements (1) can also be expressed as linear measurements of a rank-one fourth-order tensor formed by x_0, y_0, and their complex conjugates:

$$\begin{aligned} h_i[k] &= |\langle \Lambda_i^*[k],\ x_0 y_0^H \rangle|^2 + \eta_i[k] \\ &= \langle \Lambda_i^*[k],\ x_0 y_0^H \rangle \langle \Lambda_i[k],\ x_0^* y_0^T \rangle + \eta_i[k] \\ &= \langle \Lambda_i^*[k] \otimes \Lambda_i[k],\ x_0 \otimes y_0^* \otimes x_0^* \otimes y_0 \rangle + \eta_i[k] \\ &= \langle \mathcal{L}_i^k,\ \mathcal{X}_0 \rangle + \eta_i[k]. \end{aligned} \tag{7}$$

This linearity will be the foundation on which we base our iterative hard tensor thresholding approach for inverting measurements as we discuss in Section 4.4.

3. Cost function, wirtinger gradient, and wirtinger hessian

3.1. Wirtinger calculus

Gradient-based optimization techniques for phase retrieval typically differentiate a cost function built on the intensity or amplitude error between measurements generated from truth and the current approximation. Starting from (5), we construct a least-squares cost function for pulse characterization from ICC measurements that extends cost functions used in Wirtinger Flow for linear phase retrieval [26], and we provide the gradient and Hessian of that cost function. Because the vectors x_0 and y_0 are in general complex-valued, we use the Wirtinger gradient and Hessian [19,26,28]. We also characterize the ambiguities of this cost function as they align with current understandings of ICC ambiguities in the literature [12].

Throughout this section, some notation will be condensed. Let $x \in \mathbb{C}^M$ and $y \in \mathbb{C}^N$ denote the optimization variables with which we aim to recover x_0 and y_0 (up to some ambiguities to be discussed). The modulated cross-correlation at lag k between $c_i x$ and $d_i y$ is represented by $g_i[k] = x^T \Lambda_i[k] y^*$. The intensity of modulated cross-correlation for this choice of x and y is then given by $H_i[k] = |g_i[k]|^2$. Finally, the error between the actual measurements $h_i[k]$ and those predicted by x and y is given by $e_i[k] = h_i[k] - H_i[k]$.

The least-squares intensity error cost function for ICC is given by

$$f(x,y) = \frac{1}{2} \sum_{i=0}^{I-1} \sum_{k=-M+1}^{N-1} \left(h_i[k] - x^T \Lambda_i[k] y^* x^H \Lambda_i^*[k] y \right)^2. \tag{8}$$

This cost function adds the squared differences for all lags without any special weighting, and it operates on the error in intensity as opposed to magnitude (the square root of intensity, amplitude is used synonymously). Derivation of the Wirtinger gradient and Hessian is simplified with the use of a concatenated variable $z = \left[x^T, y^T\right]^T$ and corresponding cost function of this single variable. This representation and the derivation of its Wirtinger gradient and Hessian are shown in Appendix A. Their adaptation to the ICC problem is presented here.

For the cost function (8), the generic Wirtinger gradient derived in (26) can be evaluated with the substitutions made in (19) and (20) to arrive at the following expression for the ICC Wirtinger gradient:

$$\nabla f(x,y) = -\sum_{i=0}^{I-1} \sum_{k=-M+1}^{N-1} e_i[k] \begin{pmatrix} g_i[k]\Lambda_i^*[k]y \\ g_i^*[k]\Lambda_i^T[k]x \\ g_i^*[k]\Lambda_i[k]y^* \\ g_i[k]\Lambda_i^H[k]x^* \end{pmatrix}. \tag{9}$$

Note that at the ground truth (when $x = x_0$ and $y = y_0$) in a noiseless problem, $e_i[k] = 0$ for all i, k, and so the gradient $\nabla f(x,y)$ will be zero as expected. Also note that the gradient will also be zero if x and y lie in the null-space of $\sum_{i=0}^{I-1} \sum_{k=-M+1}^{N-1} e_i[k] g_i^*[k] \Lambda_i^T[k]$ and $\sum_{i=0}^{I-1} \sum_{k=-M+1}^{N-1} e_i[k] g_i[k] \Lambda_i^*[k]$, respectively. A more detailed discussion of convergence results to date will be discussed in Section 5.3.1.

Similarly, the generic Wirtinger Hessian derived in (36) can be evaluated with the substitutions made in (19) and (20) to arrive at the following expression for the ICC Wirtinger Hessian:

$$\nabla^2 f(x,y) = \sum_{i=0}^{I-1} \sum_{k=-M+1}^{N-1} v_1 v_1^H - e_i[k] \left(v_2 v_2^H + v_3 v_3^H + V \right), \tag{10}$$

where

$$v_1, v_2, v_3 = \begin{pmatrix} g_i[k]\Lambda_i^*[k]y \\ g_i^*[k]\Lambda_i^T[k]x \\ g_i^*[k]\Lambda_i[k]y^* \\ g_i[k]\Lambda_i^H[k]x^* \end{pmatrix}, \begin{pmatrix} \Lambda_i^*[k]y \\ \mathbf{0} \\ \mathbf{0} \\ \Lambda_i^H[k]x^* \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ \Lambda_i^T[k]x \\ \Lambda_i[k]y^* \\ \mathbf{0} \end{pmatrix},$$

and

$$V = \begin{pmatrix} \mathbf{0} & g_i[k]\Lambda_i^*[k] & \mathbf{0} & \mathbf{0} \\ g_i^*[k]\Lambda_i^T[k] & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & g_i^*[k]\Lambda_i[k] \\ \mathbf{0} & \mathbf{0} & g_i[k]\Lambda_i^H[k] & \mathbf{0} \end{pmatrix}.$$

Recall that at the ground truth (when $x = x_0$ and $y = y_0$) in a noiseless problem, $e_i[k] = 0$ for all i, k. In this case, the ICC Wirtinger Hessian in (10) reduces to a sum of outer products of the form $v_1 v_1^H$ and is therefore positive semi-definite. While this bodes well for local convergence properties, analysis away from ground truth is more complicated and is discussed further in Section 5.3.1.

3.2. Ambiguities

The cost function presented in (8) is invariant to certain transformations of the optimization variables. In particular, for any $x \in \mathbb{C}^M$ and $y \in \mathbb{C}^N$, $f(x,y) = f(x_a, y_a)$, where

$$\begin{aligned} x_a[n] &= a x[n] e^{j(\phi_x + \psi n)} \\ y_a[n] &= a^{-1} y[n] e^{j(\phi_y + \psi n)} \end{aligned} \tag{11}$$

for any nonzero $a \in \mathbb{R}$ and for any $\phi_x, \phi_y, \psi \in \mathbb{R}$. This fact is proved in Appendix B.

We therefore see that certain ambiguities plague the recovery of any ground truth vectors x_0 and y_0 from ICC measurements: (*i*) a global amplitude scaling factor resulting from the correlation function, (*ii*) a global phase offset resulting from phaseless measurements, and (*iii*) a frequency offset resulting from phaseless correlations. These are very similar in form to the ambiguities that are discussed in FROG recovery [5,12]. In the noiseless case, we note that the Wirtinger gradient will again be zero and the Wirtinger Hessian again positive semi-definite at any (x_a, y_a) constructed by applying the transformation (11) to the ground truth (x_0, y_0).

3.3. Wirtinger descent algorithm

To minimize the cost function (8), we use the off-the-shelf L-BFGS-B algorithm from Scipy's optimization library [29]. This function operates on real variables. We adapt the Wirtinger gradient and Hessian to functions of real variables in Appendix C. In remaining sections, L-BFGS-B applied to the cost function in Eq. (8)will be referred to as Wirtinger descent.

4. Initializer algorithms inspired by low-rank matrix and tensor formulations

In this section, we present several additional algorithms for estimating the vectors x_0 and y_0 from ICC measurements of the form (1). To complement the Wirtinger descent algorithm, which derives from the quartic bivariate vector formulation of ICC outlined in Section 2.5.1, the algorithms in this section are inspired by the quadratic low-rank matrix formulation outlined in Section 2.5.2 and the linear low-rank tensor formulation outlined in Section 2.5.3. Each of these algorithms has the capacity to solve the ICC problem, although they differ in computational complexity, memory usage, convergence rate, and sample complexity. Consequently, we have identified that the algorithms introduced in this section are well suited to the role of initializer for Wirtinger descent. Numerical comparisons of these initializers are provided in Section 5.1.

Inspired by the quadratic low-rank matrix formulation of ICC, we present a modified spectral initializer that is specially adapted to the structure of ICC measurements. Additionally, we discuss the use of alternating matrix projection algorithms for solving the ICC problem. These algorithms are capable of converging very quickly, but are typically inferior to Wirtinger descent as they require more measurements to converge.

A common problem exists between traditional spectral initializers and alternating minimization approaches for ICC in that disjoint measurements prevent information from separate lags of $h_i[k]$ from being combined effectively. To remedy this, we also lift the low-rank matrix problem into a low-rank tensor recovery problem, which we solve using an iterative hard tensor thresholding algorithm. This algorithm is capable of operating in a low measurement regime, but it is computationally intensive. For this reason, it is better suited as a starting point for Wirtinger descent.

4.1. Differences from classical low-rank phase retrieval

The formulation of ICC in (6) shows that recovery from ICC can be viewed as a low-rank phase retrieval problem. However, several major differences exist between ICC and problems studied in the low-rank phase retrieval literature [6,7,27].

The first difference is that the objective matrix $x_0 y_0^H$ in (6) is not only low-rank, but known a priori to be rank-one. While this is a powerful prior, a complication that exists in ICC is that the measurement matrix $\Lambda_i^*[k]$ has a rank that is typically $u[k] - l[k]$ (assuming the entries of c_i, d_i are non-zero). In contrast, most low-rank phase retrieval problems involve interactions with low-rank measurement matrices [7,27].

An additional complication is the disjoint nature of ICC measurements as a function of k. For $k_1, k_2 \in \{-M+1, \ -M+2, \ \dots, \ N-1\}$ with $k_2 \neq k_1$, we have $\langle \Lambda_i^*[k_1], \ \Lambda_i^*[k_2] \rangle = 0$ for all i. We also note that for a given k, $\Lambda_i^*[k]$ is only capable of operating on the kth diagonal of $x_0 y_0^H$ and no "global" (i.e., multi-diagonal) measurements of $x_0 y_0^H$ exist. This leaves the rank-one structure of $x_0 y_0^H$ as the only assumption that can combine information across diagonals.

Another way to view this stark contrast with existing low-rank phase retrieval works is to think of the ICC recovery problem as a series of $M + N - 1$ disjoint linear phase retrieval problems ($y = |Ax|^2$) that each contribute a single diagonal of the matrix $x_0 y_0^H$ as shown in Fig. 2. Measurements $h_i[k]$ for a fixed k are created using only the kth diagonal of the outer product matrix $x_0 y_0^H$. One could then attempt to solve for each of the diagonals separately using any preferred phase retrieval algorithm, and collect their results at the end to solve for the outer product matrix $x_0 y_0^H$. However, because phase retrieval algorithms recover vectors only up to an ambiguous global phase offset, all diagonals recovered separately will have a unique global phase-offset. This means that if one were to form the $M + N - 1$ phase retrieval problems and solve for each diagonal of $x_0 y_0^H$, each diagonal would be multiplied by an arbitrary unknown phase, and the resulting matrix would in general have rank greater than one.

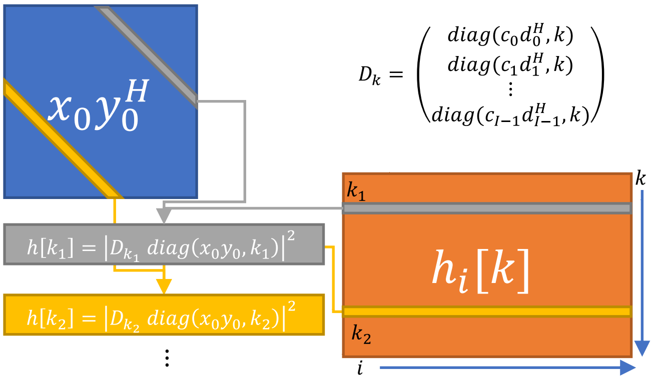
$x_0 y_0^H$
$D_k = \begin{pmatrix} diag(c_0 d_0^H, k) \\ diag(c_1 d_1^H, k) \\ \vdots \\ diag(c_{I-1} d_{I-1}^H, k) \end{pmatrix}$
$h[k_1] = |D_{k_1} \, diag(x_0 y_0, k_1)|^2$
$h[k_2] = |D_{k_2} \, diag(x_0 y_0, k_2)|^2$
$\vdots$
k_1
$h_i[k]$
k_2
k
i

Fig. 2. ICC measurements from (1) represent k diagonal-disjoint phase retrieval problems of a rank-one matrix. Every measurement $h_i[k]$ represents the intensity of the inner product between a diagonal of our objective and a diagonal of a rank-one measurement matrix: $h_i[k] = \left| \langle \mathrm{diag}(c_i d_i^H, \ k)^*, \ \mathrm{diag}(x y^H, \ k) \rangle \right|^2$.

This lack of global measurements also prevents spectral decomposition from properly combining information in spectral initializers like those employed in [7,30]. For the ICC problem, the spectral initializer is mostly easily constructed from the tensor formation shown in (7), where the ICC measurements are the linear projection of the fourth order tensor $\mathcal{X} = x_0 \otimes y_0^* \otimes x_0^* \otimes y_0$ onto successive fourth-order measurement tensors $\mathcal{L}_i^k = \Lambda_i^*[k] \otimes \Lambda_i[k]$. Letting $\mathrm{vec}\ (\mathcal{X})$ denote a vectorization of $\mathcal{X}$ and $\mathcal{Y}$ denote a vector containing the ICC measurements $\{h_i[k]\}_{i,k}$, we may write $\mathcal{Y} = \mathcal{A}\,\mathrm{vec}\ (\mathcal{X})$, where $\mathcal{A}$ is a matrix whose rows are formed by the appropriate vectorizations of $\{\mathcal{L}_i^k\}_{i,k}$. A typical spectral initializer at this stage would use the adjoint of $\mathcal{A}$ to construct $\mathcal{Z} = \mathcal{A}^H \mathcal{Y}$, reshape $\mathcal{Z}$ to form a matrix corresponding to the unfolding of $\mathcal{X} \rightarrow \mathrm{vec}\ (x_0 y_0^H)(\mathrm{vec}\ (x_0 y_0^H))^H$, and then take the leading eigenvector of this matrix as an initial guess of $\mathrm{vec}\ (x_0 y_0^H)$. We will refer to the matrix reshaping of $\mathcal{Z}$ as the spectral matrix. In typical low-rank phase retrieval settings [7], measurement matrices are global and the spectral matrix has a leading eigenvector that contains global information from the objective matrix. But here, $\langle \mathcal{L}_i^{k_1}, \mathcal{L}_j^{k_2} \rangle = 0$ when $k_1 \neq k_2$, and the resulting spectral matrix will produce a block diagonal matrix as shown in Fig. 3. In particular, the spectral matrix produced for ICC measurements will have a block for every diagonal of $x_0 y_0^H$, and taking the leading eigenvector will merely return a spectral estimate for the largest energy diagonal of $x_0 y_0^H$. Spectral decomposition could be used with the largest $M + N - 1$ eigenvectors of the spectral matrix to recover estimates of all of the diagonals of $x_0 y_0^H$, but these would again be ambiguous up to phase rotations and the resulting estimate of $x_0 y_0^H$ would in general have rank greater than one. A technique to align these diagonals is presented in Section 4.2.

Because $\mathrm{vec}\ (x_0 y_0^H)(\mathrm{vec}\ (x_0 y_0^H))^H$ is inherently a rank-one matrix, one may consider low-rank matrix completion techniques to augment the block diagonal estimates such as those in the right panel of Fig. 3. However, we consider an even more direct approach for exploiting low-rank structure. The values of $\mathcal{Z}$ can be reshaped into a tensor approximation of $\mathcal{X}$, and iterative hard thresholding on this tensor forms the core principle of the initializer we present in Section 4.4.

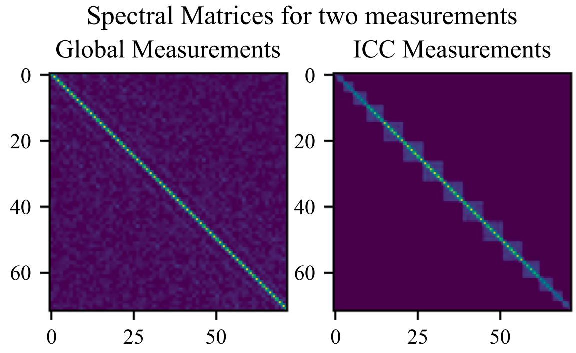

Fig. 3. Using $x_0 \sim \mathcal{C}(\mathbf{0}, \mathbb{I}^6)$ and $y_0 \sim \mathcal{C}(\mathbf{0}, \mathbb{I}^{12})$, we construct a spectral matrix in the fashion of [7] from 1000 measurements of the form $h_i = \left|\langle A_i,\ x_0 y_0^H\rangle\right|^2$. For the purposes of illustration, the outer product matrix $x_0 y_0^H$ has been vectorized along its diagonals. Left: measurements h_i correspond to global rank-one measurements, where each A_i is the outer product of two vectors. Such measurements align with recent research in low-rank phase retrieval [7,27]. The leading eigenvector for the spectral matrix (which is not visually apparent because the matrix also contains a multiple of the identity) can be reshaped into a 6×12 matrix to provide a spectral initializer for $x_0 y_0^H$ in low-rank phase retrieval. Right: measurements h_i depend on diagonals of $x_0 y_0^H$, i.e., each A_i is a high-rank, diagonal matrix as one encounters in ICC. Each eigenvector of this block-diagonal spectral matrix will only provide information about a single diagonal of $x_0 y_0^H$.

4.2. Spectral initializer for disjoint low-rank phase retrieval

We present here a low-complexity approach to synchronize the phases across noisy estimates of the diagonals of the rank-one matrix $x_0 y_0^H$. The aim of this technique is to resolve the phase mismatch between diagonals recovered with spectral techniques to produce a rank-one matrix whose leading eigenvector serves as a good estimate of vec $(x_0 y_0^H)$.

Our approach is based on an examination of local sub-matrix conditions in a properly phase synchronized matrix. To illustrate local regularity condition we employ, for two vectors $x \in \mathbb{C}^M$ and $y \in \mathbb{C}^N$, consider an outer product matrix of the form xy^H that has been multiplied by an arbitrary phase per diagonal ϕ_k. In any 2×2 sub-matrix with nonzero entries, notice that the product of two diagonal entries divided by the product of the two anti-diagonal entries produces a ratio that is invariant to x and y and depends only on a product of phase terms of the form $\phi_n \phi_n \phi_{n-1}^* \phi_{n+1}^*$, as shown in Fig. 4. In a rank-one matrix, this ratio must always be equal to 1; therefore this quantity gives a local regularity condition that measures how well the phases are synchronized to give a rank-one matrix. Moreover, this leads to a constructive procedure for synchronizing the phases. Computing the ratio for every 2×2 sub-matrix, we obtain a series of cross terms that each depend on no less than 3 unique entries of ϕ. This leaves two elements of ϕ as free parameters. For algorithms in this paper, we set $\phi_0 = \phi_1 = 1$ arbitrarily. With this assumption in hand, ϕ_2 can be solved for using the products $\phi_1 \phi_1 \phi_0 \phi_2^* = \phi_2^*$, and so on.

More generally, this problem can be thought of as a phase synchronization problem [31] where we have MN third-order cross-term measurements of three vectors x, y, ϕ. While it may thus be feasible to pose another optimization at this stage, where the rank of the resulting misaligned outer product matrix is minimized by varying diagonal phase offsets, we select the local regularity approach because it is largely algebraic and requires only simple assumptions. We note that the ambiguities discussed in Section 3.2 remain and cannot be resolved by any algorithm without additional assumptions. In particular, the unknown phase offset between ϕ_0 and ϕ_1 propagates across all values of ϕ, which manifests as an unknown frequency offset in the recovery of x and y.

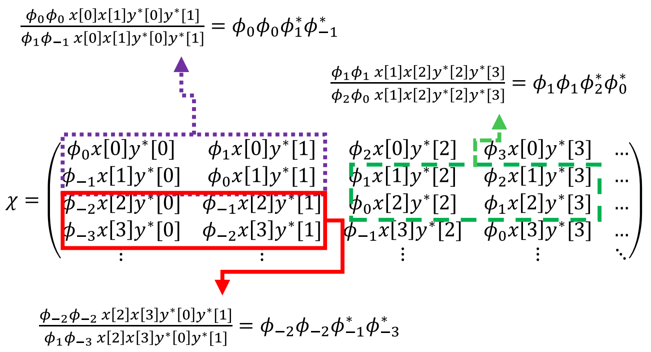
$\frac{\phi_0\phi_0\, x[0]x[1]y^*[0]y^*[1]}{\phi_1\phi_{-1}\, x[0]x[1]y^*[0]y^*[1]} = \phi_0\phi_0\phi_1^*\phi_{-1}^*$
$\frac{\phi_1\phi_1\, x[1]x[2]y^*[2]y^*[3]}{\phi_2\phi_0\, x[1]x[2]y^*[2]y^*[3]} = \phi_1\phi_1\phi_2^*\phi_0^*$
$\chi =$
$\phi_0 x[0]y^*[0]$
$\phi_1 x[0]y^*[1]$
$\phi_2 x[0]y^*[2]$
$\phi_3 x[0]y^*[3]$
$\phi_{-1} x[1]y^*[0]$
$\phi_0 x[1]y^*[1]$
$\phi_1 x[1]y^*[2]$
$\phi_2 x[1]y^*[3]$
$\phi_{-2} x[2]y^*[0]$
$\phi_{-1} x[2]y^*[1]$
$\phi_0 x[2]y^*[2]$
$\phi_1 x[2]y^*[3]$
$\phi_{-3} x[3]y^*[0]$
$\phi_{-2} x[3]y^*[1]$
$\phi_{-1} x[3]y^*[2]$
$\phi_0 x[3]y^*[3]$
$\frac{\phi_{-2}\phi_{-2}\, x[2]x[3]y^*[0]y^*[1]}{\phi_1\phi_{-3}\, x[2]x[3]y^*[0]y^*[1]} = \phi_{-2}\phi_{-2}\phi_{-1}^*\phi_{-3}^*$

Fig. 4. When xy^H is multiplied by an arbitrary phase offset ϕ_k on each diagonal, any 2×2 submatrix can be used to compute a ratio that depends only on the phase offsets. Ensuring these ratios are equal to 1 ensures the phases are synchronized so the matrix can have rank 1.

In practice, an estimate of $x_0 y_0^H$ may contain both magnitude and phase errors. Algorithmically, we align magnitude and phase in two different steps. Magnitude errors in the entries of the misaligned xy^H are corrected iteratively because of the coupled nature of the ratio measurements, and the algorithm used for this process is detailed in Algorithm 1. Phase is solved for directly as detailed in Algorithm 2. We adopt the following notation: χ_{spec} represents the collection of the largest $M + N - 1$ eigenvectors of the spectral matrix collected as misaligned diagonals of an approximation of $x_0 y_0^H$, χ_{mag} represents the local magnitude aligned version of χ_{spec}, and $\chi_{aligned}$ represents the phase and magnitude aligned estimate of $x_0 y_0^H$.

Algorithm 1: Iterative magnitude correction for diagonally disjoint recoveries (MagReg)

```
Data:
$\chi_{spec} \in \mathbb{C}^{M \times N}$
Parameters:
maxIts = 1000
thresh = 1e − 4
Result: $\chi_{mag} \in \mathbb{C}^{M \times N}$
$\chi \leftarrow \chi_{spec}$
for( $ii = 0,\ 1,\ \ldots,\ maxIts - 1$ ) {
    // $adjust = \mathbf{1} \in \mathbb{R}^{M \times N},\ diff \in \mathbb{R}^{(M-1)\times(N-1)}$
    $\chi_{old} \leftarrow \chi$
    $adjust = \mathbf{1}$
    $diff = \left|\frac{\chi[:-1,1:]\chi[1:,:-1]}{\chi[:-1,:-1]\chi[1:,1:]}\right|^{1/8}$
    $adjust[:-1,:-1] *= diff$
    $adjust[1:,1:] *= diff$
    $adjust[:-1,1:] /= diff$
    $adjust[1:,:-1] /= diff$
    $\chi = \chi * adjust$
    if $||\chi - \chi_{old}||_2^2 / ||\chi||_2^2 < thresh$ then
        break
    else if $ii == maxIts$ then
        break
}
$\chi_{mag} \leftarrow \chi$
```

Finally, we present a complete spectral initializer in Algorithm 3 that uses the magnitude and phase alignment techniques described in Algorithms 1 and 2. Fig. 5 shows results from spectral initializers with and without magnitude and phase alignment as well as the result of a spectral initializer that uses the single largest eigenvector [7]. The phase and magnitude alignment concentrates the eigenspectrum similar to that of the ground truth, and produces a much finer estimate of magnitude than the unaligned attempt. The estimate of phase from the aligned version does appear noisy in comparison to the unaligned initializer, however this is only in the region where the unaligned initializer produced estimates significantly larger than zero. Across the

Algorithm 2: Iterative phase correction for diagonally disjoint recoveries (PhaseReg)

```
Data:
    $\chi_{mag} \in \mathbb{C}^{M \times N}$
Result: $\chi_{aligned} \in \mathbb{C}^{M \times N}$
$adjust = \mathbf{1} \in \mathbb{C}^{M \times N}$
for( $k = [-M+1, \ -M+2, \ \dots, \ N-1$ ) {
    $compare = exp\left(j\angle\left(\frac{\chi[:-1,1:]\chi[1:,:-1]}{\chi[:-1,:-1]\chi[1:,1:]}\right)\right)$
    if $k < 0$ then
        $\mathrm{diag}(adjust, \ k)[: -1] *= \mathrm{diag}(compare, \ k+1)$
    else if $k > 1$ then
        $\mathrm{diag}(adjust, \ k)[: -1] *= \mathrm{diag}(compare, \ k-1)$
}
for( $k = [-M+1, \ -M+2, \ \dots, \ N-1$ ) {
    if $k < 0$ then
        $\mathrm{diag}(adjust, \ k) *= \frac{\mathrm{diag}(adjust, \ k+1)[:-1]^2}{\mathrm{diag}(adjust, \ k+2)[:-2]}$
    else if $k > 1$ then
        $\mathrm{diag}(adjust, \ k) *= \frac{\mathrm{diag}(adjust, \ k-1)[:-1]^2}{\mathrm{diag}(adjust, \ k-2)[:-2]}$
}
$\chi_{aligned} = \chi_{mag} * exp(-j\angle(adjust))$
```

full signal width, the aligned estimate is more consistent. This higher concentration of eigenspectrum in the aligned initializer aligns with a priori knowledge that the spectral initializer here should be rank-one, and was chosen to initialize Wirtinger descent later. It should be noted that the traditional approach ("spectral") of using only a single eigenvector from the spectral matrix fails almost completely here, seemingly able only to estimate a single large value of x.

Algorithm 3: Spectral initializer with iterative magnitude and phase correction

```
Data:
    $h_i[k] = \left|c_i x_0 \bigotimes_k d_i y_0\right|^2 + \eta_i[k] \in \mathbb{R}^{I \times (M+N-1)}$
Result: $x \in \mathbb{C}^M, \ y \in \mathbb{C}^N$
$\chi_{spec} \leftarrow \mathbf{0} \in \mathbb{C}^{M \times N}$
for( $k = [-M+1, \ -M+2, \ \dots, \ N-1$ ) {
    // $A_k \in \mathbb{C}^{I \times (u-l)}, \ h_k \in \mathbb{R}^I, \ D \in \mathbb{C}^{(u-l) \times (u-l)}$
    $A_k = \begin{pmatrix} \mathrm{diag}(c_0 d_0^H, \ k) \\ \mathrm{diag}(c_1 d_1^H, \ k) \\ \vdots \\ \mathrm{diag}(c_{I-1} d_{I-1}^H, \ k) \end{pmatrix}$
    $h_k = \begin{pmatrix} h_0[k] \\ h_1[k] \\ \vdots \\ h_{I-1}[k] \end{pmatrix}$
    $D = A_k^H \, \mathrm{diag}(h_k) A_k$
    $u, s, v^H = SVD(D)$
    $\mathrm{diag}(\chi_{spec}, \ k) = u[:, 0]\sqrt{s[0]}$
}
$\chi_{mag} = \mathrm{MagReg}(\chi_{spec})$
$\chi_{aligned} = \mathrm{PhaseReg}(\chi_{mag})$
$U, S, V^H = SVD(\chi_{aligned})$
$x = U[:, 0]\sqrt{S[0]}$
$y = V[:, 0]\sqrt{S[0]}$
```

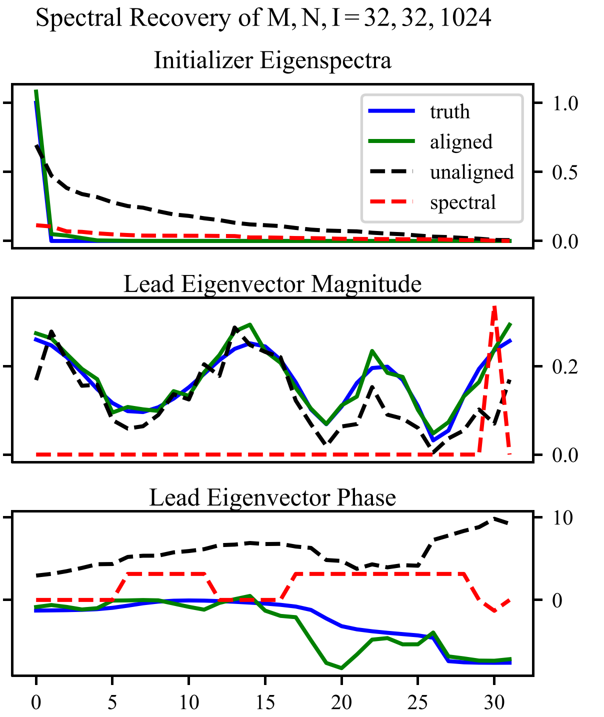

Fig. 5. Several spectral initializer approaches are applied to an $M, N, I = 32, 32, 1024$ ICC measurement set of a low-bandwidth signal with full-bandwidth complex Gaussian i.i.d. measurements. Here the eigenvectors corresponding to the x recovery are shown. Selecting only the largest eigenvector, as is common in the literature [7] produces the result labeled "spectral". Recovering every diagonal and collecting them into a single matrix without diagonal phase alignment is labeled "unaligned". Finally, applying the magnitude and phase alignment techniques discussed in this paper is shown under the label "aligned". The concentration of eigenvalues for the aligned technique demonstrates the strengths of the local regularity condition at correcting the unknown diagonal phase offsets.

4.3. Matrix projective algorithms

PCGPA represented an early projective algorithm in the optics community [9] that alternated a magnitude constraint with a rank-one enforcement to invert FROG measurements. For our approach, we alternate an AltMinPhase step [32] that is outlined in Algorithm 4 and an SVD enforced rank-one condition that is outlined in Algorithm 5. This technique is modeled after [6], with appropriate substitutions made for known rank-one a priori. To distinguish this technique, we will identify it as Matrix AltMin throughout the rest of the paper. This approach is extremely close in nature to PCGPA and is an important baseline to compare against because of its common use. It is important to note that this algorithm still has the identical phase misalignment problem that the spectral methods mentioned earlier encounter. Here iterative applications of rank-one projections and AltMinPhase projections are applied to a null-initialized matrix until a stable result is produced. The complete alternating minimization algorithm is detailed in Algorithm 6. Some additional notation useful here: χ is the current guess of the outer-product matrix $x_0 y_0^H$, χ_{mag} is the output of χ when AltMinPhase is applied to every diagonal, and χ_{rank} is the rank-one projection of χ.

It should be noted here that in Algorithm 6, we use pseudo-inverse operators instead of adjoints for our magnitude back-projection. Because our problem is disjoint per diagonal on the outer product of xy^H, we found that computing these pseudo-inverses added little time to computation and typically provided faster convergence compared to using adjoints. Using the adjoint typically works well when properly scaled, but takes more iterations to converge.

4.4. Tensor projective algorithms

As outlined in Section 2.5.3, ICC can be viewed as taking linear measurements of a rank-one, fourth order tensor $\mathcal{X} = x_0 \otimes y_0^* \otimes x_0^* \otimes y_0$. As an alternative to matrix-based projective algorithms, which still involve phase retrieval, one can attempt to directly estimate the tensor $\mathcal{X}$ and then factor this tensor to obtain estimates of x_0 and y_0.

Algorithm 4: Matrix magnitude back-projection (MatrixMagProject)

```
Data:
    $\chi \in \mathbb{C}^{M\times N}$
    $A_k$, forward operator for all $k$
    $A_k^+$, pseudo-inverse of each forward operator
    $h_i[k] = \left|c_i x_0 \bigotimes_k d_i y_0\right|^2 + \eta_i[k] \in \mathbb{R}^{I\times(M+N-1)}$
    $h_k = \{h_0[k],\ h_1[k],\ \ldots,\ h_{I-1}[k]\}$
Result: $\chi_{mag}$
for( $k = [-M+1,\ -M+2,\ \ldots,\ N-1$ ) {
    forward = $A_k$ diag($\chi$, $k$)
    phasedMag = $\sqrt{h_k}exp(j\angle(\text{forward}))$
    diag($\chi_{mag}$, $k$) = $A_k^+$phasedMag
}
```

Algorithm 5: Single rank matrix projection (MatrixRankProject)

```
Data:
    $\chi \in \mathbb{C}^{M\times N}$
Result: $\chi_{rank}$
$U, S, V^H = SVD(\chi)$
$x = U[:,0]\sqrt{S[0]}$
$y = V[:,0]\sqrt{S[0]}$
$\chi_{rank} = xy^H$
```

Algorithm 6: Alternating Magnitude Backprojection and Rank Constrained Minimization (Matrix AltMin)

```
Data:
    $h_i[k] = \left|c_i x_0 \bigotimes_k d_i y_0\right|^2 + \eta_i[k] \in \mathbb{R}^{I\times(M+N-1)}$
    $c \in \mathbb{C}^{I\times M}$, $d \in \mathbb{C}^{I\times N}$
Parameters:
    thresh = $1e-6$
    maxIts = 1000
Result: $x \in \mathbb{C}^M$, $y \in \mathbb{C}^N$
$\chi \leftarrow \mathbf{0} \in \mathbb{C}^{M\times N}$
for( $k = [-M+1,\ -M+2,\ \ldots,\ N-1$ ) {
    $rows = l[k] : 1 : u[k]$
    $cols = l[k] : 1 : u[k] + k$
    $A_k = c[:, rows] * d^*[:, cols]$ // $A_k \in \mathbb{C}^{I\times(u-l)}$
    $A_k^+ = \text{pinv}(A_k)$
}
for( $ii = 0,\ 1,\ \ldots,\ maxIts - 1$ ) {
    $\chi_{old} \leftarrow \chi$
    $\chi_{mag}$ = MatrixMagProject($\chi$, $A_k$, $A_k^+$, $h$)
    $\chi$ = MatrixRankProject($\chi_{mag}$)
    if $||\chi - \chi_{old}||_2^2/||\chi||_2^2 < thresh$ then
        break
    else if $ii == maxIts$ then
        break
}
$U, S, V^H = SVD(\chi)$
$x = U[:,0]\sqrt{S[0]}$
$y = V[:,0]\sqrt{S[0]}$
```

Tensor iterative hard thresholding [33,34] is a two step protective algorithm that alternates between back-projecting error and promoting low tensor rank. To promote low tensor rank, we use the Tucker decomposition to reduce our tensor, then use only the top eigenvector to reconstruct a rank-one tensor as shown in Algorithm 7. Measurements in the back-projection step are still disjoint over k, and therefore this step can be broken into $M + N - 1$ back-projections as shown in Algorithm 8; we again choose pseudo-inverses over adjoints. The complete alternating minimization algorithm that employs both of these projections is detailed in Algorithm 6. At low measurement counts, Algorithm 6 may converge to tensors with smaller or larger norm magnitude if this algorithm is initiated at zero or randomly. For this reason, results are scaled by Algorithm 10 rather than by core tensor value, as it typically provides a better estimate of scale. At higher measurement counts, when Algorithm 6 converges to a tighter estimate of x, y, the scaling factor is typically identical to $\sqrt{c/2}$, where c is the single value of the core tensor. Tensor decompositions in this project were performed using the Python tensor decomposition package Tensorly [35]. Conceptually, in the same way PhaseLift [21] lifts the non-linear inversion of linear phase retrieval to linear low-rank matrix recovery, we have lifted the non-linear inversion of low-rank phase retrieval to linear low-rank tensor recovery. Additional notation useful in this section includes: χ is the current estimate of $x_0 \otimes y_0^* \otimes x_0^* \otimes y_0$, χ_{back} is χ added to the backprojection of its forward error, and χ_{rank} is the rank-one projection of χ.

Algorithm 7: Rank-one tensor projection (TensorRankProject)

```
Data:
    $\chi \in \mathbb{C}^{M\times N\times M\times N}$
Result: $\chi_{rank}$
$v[0], v[1], v[2], v[3] = \text{tucker}(\chi,\ \text{rank} = [1,1,1,1])$
$\chi_{rank} = v[0] \otimes v[1] \otimes v[0]^* \otimes v[1]^*$
```

Algorithm 8: Tensor error back-projection (TensorBackProject)

```
Data:
    $\chi \in \mathbb{C}^{M\times N\times M\times N}$
    $A_k$, forward operator for all $k$
    $A_k^+$, pseudo-inverse of each forward operator
    $h_i[k] = \left|c_i x_0 \bigotimes_k d_i y_0\right|^2 + \eta_i[k] \in \mathbb{R}^{I\times(M+N-1)}$
Result: $\chi_{back}$, $e \in \mathbb{R}$ Frobenius error
$e = 0$
$\chi_{back} \leftarrow \mathbf{0} \in \mathbb{C}^{M\times N\times M\times N}$
for( $k = [-M+1,\ -M+2,\ \ldots,\ N-1$ ) {
    $rows = l[k] : 1 : u[k]$
    $ii = \text{repeat}(rows,\ u - l)$
    $jj = \text{tile}(rows,\ u - l) + k$
    $kk = ii + k$
    $ll = jj + k$
    flatKSection = $\chi[ii, jj, kk, ll]$
    error = $h_k - A_k$flatKSection
    backProject = $A_k^+$error
    $\chi_{back}[ii, jj, kk, ll] = \chi[ii, jj, kk, ll]$ + backProject
     $e = e + ||\text{error}||_F^2$
}
```

4.5. Alternating projections and variants

Projective algorithms have been studied in the context of phase retrieval for many years, and variants studied dating back to the early 2000's demonstrated superior rates of convergence. Relaxed Averaged Alternating Reflections (RAAR) [36] is one such algorithm and is widely used. We have adapted RAAR to both the matrix and tensor projective algorithms in an attempt to improve rates. RAAR algorithms used here assume there are two projection operators P_1, P_2, which for us are the functions MatrixRankProject and MatrixMagProject for the matrix case, and TensorRankProject and TensorBackProject for the tensor case. With P_1, P_2 so defined, RAAR variants of our matrix and tensor initializers are identical to their alternating minimization counter-parts, with the exception of a new update step per iteration shown for RAAR in Algorithm 11. This update step depends on two parameters α, β; in numerical tests, we find that setting $\alpha = 0.9$ and $\beta = 0.5$ works well.

Algorithm 9: Iterative hard tensor thresholding (Tensor AltMin)

```
**Data:**
$h_i[k] = \left|c_i x_0 \bigotimes_k d_i y_0\right|^2 + \eta_i[k] \in \mathbb{R}^{I\times(M+N-1)}$
$c \in \mathbb{C}^{I\times M},\ d \in \mathbb{C}^{I\times N}$
**Parameters:**
thresh $= 1e-6$
maxIts $= 1000$
**Result:** $x \in \mathbb{C}^M,\ y \in \mathbb{C}^N$
$\chi \leftarrow C(0,1) \in \mathbb{C}^{M\times N\times M\times N}$
for( $k = [-M+1,\ -M+2,\ \dots,\ N-1$ ) {
    $rows = l[k] : 1 : u[k]$
    $ii = \text{repeat}(rows,\ u-l)$
    $jj = \text{tile}(rows,\ u-l) + k$
    $kk = ii + k$
    $ll = jj + k$
    for( $i = [\ 0,\ 1,\ \dots,\ I-1]$ ) {
        $v_k[i] = \text{diag}(c[i,:]d[i,:]^H,\ k)$
        $A_k[i,:] = \text{flatten}(v_k[i] \otimes v_k[i]^*)$
        // $A_k \in \mathbb{C}^{I\times(u-l)^2}$
    }
    $A_k^+ = \text{pinv}(A_k)$
}
for( $ii = 0,\ 1,\ \dots,\ maxIts - 1$ ) {
    $\chi_{back}, e = \text{TensorErrorProject}(\chi,\ A_k,\ A_k^+,\ h)$
    $\chi = \text{TensorRankProject}(\chi_{back})$
    if $e < thresh$ then
        break
    else if $ii == maxIts$ then
        break
}
$X, Y^*, X^*, Y = \text{tucker}(\chi,\ \text{rank} = [1,1,1,1])$
$x, y = \text{measurementScale}(\ X,\ Y,\ h,\ c,\ d)$
```

Algorithm 10: measurementScale

```
**Data:**
$x \in \mathbb{C}^M,\ y \in \mathbb{C}^N$
$c \in \mathbb{C}^{I\times M},\ d \in \mathbb{C}^{I\times N}$
$h_i[k] = \left|c_i x_0 \bigotimes_k d_i y_0\right|^2 + \eta_i[k] \in \mathbb{R}^{I\times(M+N-1)}$
**Result:** $x, y$ correct relative scaling
$H_i[k] = \left|c_i x \bigotimes_k d_i y\right|^2$
ratio $= \text{mean}(H_i[k]/h_i[k] \forall i, k)$
scale $= \text{ratio}^{1/4}$
$x = x/\text{scale}$
$y = y/\text{scale}$
```

Algorithm 11: RAAR update for single iteration

```
**Data:**
$\chi_i$ current guess
**Parameters:**
$\alpha = .9$
$\beta = .5$
**Result:** $\chi_{i+1}$
/* At iteration $i$ */
$r_1 = 2P_2(\chi_i) - \chi_i$
$r_2 = 2P_1(r_1) - P_2(\chi_i)$
$m_1 = \beta(\alpha r_2 + (1-\alpha)\chi_i)$
$m_2 = (1-\beta)P_2(\chi_i)$
$\chi_{i+1} = m_1 + m_2$
```

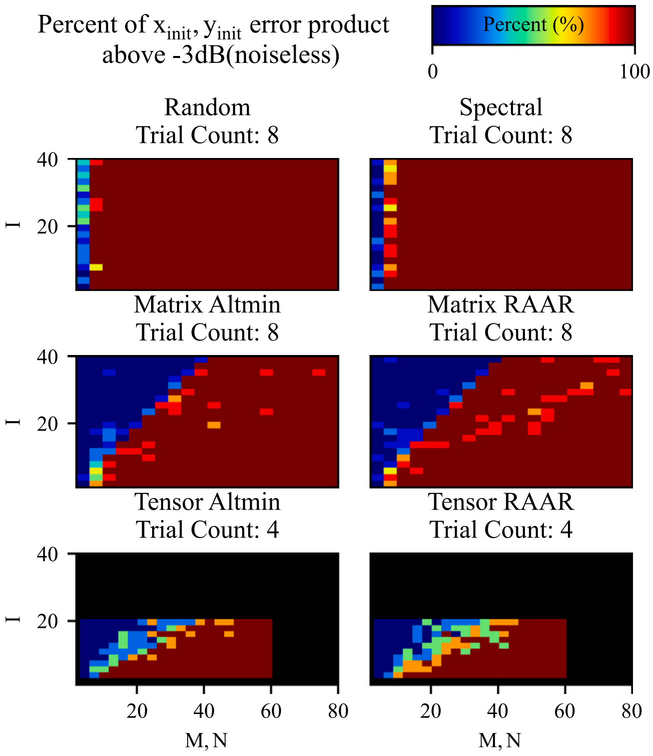

Fig. 6. Various initializers are compared, and the resulting error in the x, y estimates are multiplied and shown as a function of $M = N$ and I. With respect to problem size M, N, tensor methods require less measurements I than matrix methods. At the scales presented here, spectral methods do not show any dramatic improvement over the random initializer. Trial spaces presented in black were not tested due to prohibitive processing time.

5. Performance results

5.1. Numerical testing

Initial results for numerical Monte-Carlo testing of algorithms are presented in this section. For the purposes of testing, $x_0 \sim C(\mathbf{0}, \mathbb{I}^M)$, $y_0 \sim C(\mathbf{0}, \mathbb{I}^N)$, and each $c_i \sim C(\mathbf{0}, \mathbb{I}^M)$ and $d_i \sim C(\mathbf{0}, \mathbb{I}^N)$ for all $i = [0, 1, \dots, I-1]$.

Fig. 6 displays the resulting error of the initializers developed in Section 4 over several trials with free variables $M = N$ and I. Error in Fig. 6 for a given estimate x, y of x_0, y_0 is calculated as $\|x_0 - \min_{\theta,\phi}(\phi x e^{j\theta n})\|_2 / \|x_0\|_2$ times $\|y_0 - \min_{\theta,\phi}(\phi y e^{j\theta n})\|_2 / \|y_0\|_2$, the product of unambiguous, normalized error in x and the unambiguous, normalized error in y. Fig. 6 shows the percentage over which this error product was below -3 dB over several experiments. Each initializer was allowed to iterate $maxIts = 1000$ times and was required to terminate at a threshold condition of $thresh = 1e-6$. The maximum number of iterations was typically reached for low numbers of measurements I, while the threshold condition was typically reached in higher measurement regimes.

Based on the output from experiments in Fig. 6, we run L-BFGS-B Wirtinger descent initialized with the result of the various initialization algorithms. The final product of unambiguous errors (after Wirtinger descent) is shown in Fig. 7. Fig. 8 shows the median total runtime from initializer through descent for the tests run in Fig. 7.

From the numerical results presented here, tensor initializers allow convergence from a reduced number of ICC measurements, but this comes at the expense of a much longer run time. Otherwise, there appears to be no definite advantage in using an initializer before Wirtinger descent for this problem.

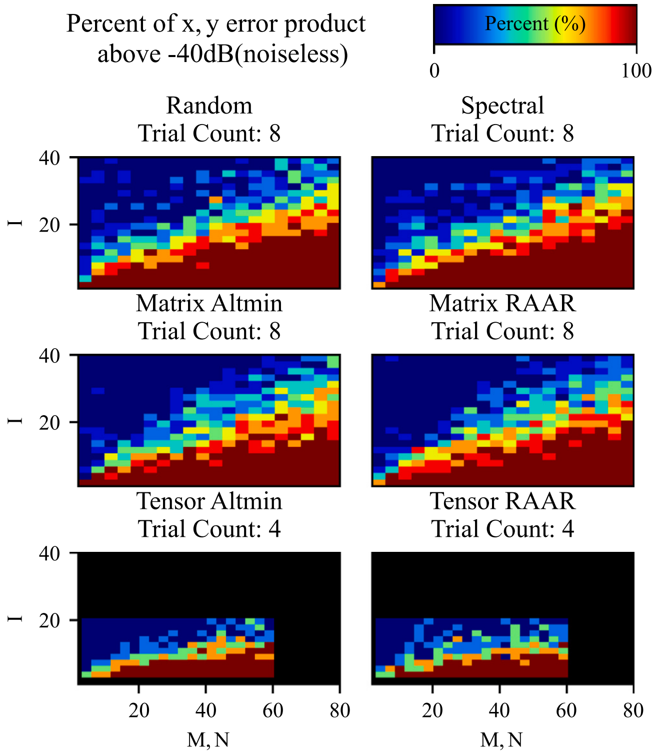

Fig. 7. Various initializers are followed with Wirtinger descent, and the resulting error in the x, y estimates are multiplied and shown as a function of $M = N$ and I. Compared to random initialization, there does not appear to be a strong improvement from the spectral initializer or the matrix alternating minimization methods when followed by Wirtinger descent. Only the iterative hard tensor thresholding initialization methods appear to place the Wirtinger descent in a more advantaged initial point than random.

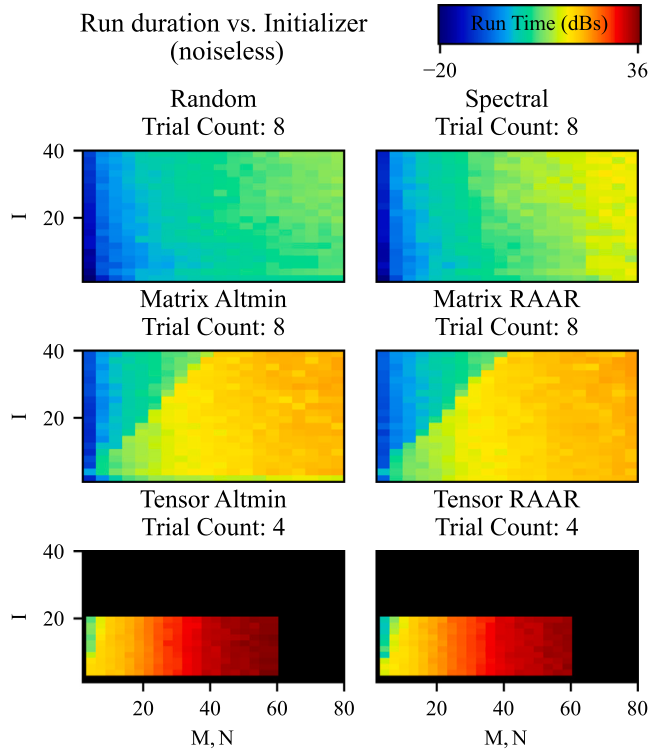

Fig. 8. Various initializers are followed with Wirtinger Descent, and resulting runtimes are shown as a function of $M = N$ and I. Tensor methods take significantly longer than other approaches studied.

Table 1
Memory and Computational complexity per iterations of presented algorithms.

Algorithm	Memory	Compute
Matrix AltMin	$\mathcal{O}\left(IK^2\right)$	$\mathcal{O}\left(IK^2+K^3\right)$
Tensor AltMin	$\mathcal{O}\left(IK^3+K^4\right)$	$\mathcal{O}\left(IK^3+K^4\right)$
Gradient	$\mathcal{O}\left(IK^2\right)$	$\mathcal{O}\left(IK^2\right)$

5.2. Complexity

As demonstrated in Fig. 8, tensor methods typically take much longer to compute than matrix alternating methods and randomly initialized gradient descent. This merits a discussion on the memory and per iteration computational complexity of the presented methods, along with a discussion on their scalability to practical problem sets. For the purposes of this discussion, we focus comparison to Matrix AltMin, Tensor AltMin, and Wirtinger descent. RAAR per iteration complexity will be a constant multiple more complex than alternating minimization methods, and spectral initializers are not discussed here because of their ineffectiveness demonstrated in Section 5.1. Common terms appear in both the discussion of computational and memory complexity, which are listed and simplified here.

- $Q = \min(M, N) \geq \min(M, N-k) - \min(0, k)$ bounds the length of each diagonal in the matrix xy^H.
- $P = \max(M, N)$.
- SVD computational complexity for a matrix $xy^H \in \mathbf{C}^{M\times N}$ is $\mathcal{O}\left(P^2Q + Q^3\right)$ [37].
- Tensor t-HOSVD computational complexity for a tensor $xy^Hx^Hy \in \mathbb{C}^{M\times N\times M\times N}$ is $\mathcal{O}\left(P^2Q^2\right)$ [38].
- K is used to simplify expressions when Q, P are roughly the same value, e.g. $Q \approx P \approx K$.

5.2.1. Memory complexity

Persistent memory storage between each iteration is presented in Table 1.

Matrix AltMin stores only two leading guess vectors (P, Q terms), a matrix that is their outer product (QP terms), and the adjoint or pseudo-inverse matrices that relate measurements to diagonals ($(P + Q - 1)\times Q\times I$ terms). The adjoint matrix dominates memory complexity and scales as $\mathcal{O}(IK^2)$.

Tensor AltMin stores two leading guess vectors (P, Q terms), a tensor that represents the fourth-order outer product terms (P^2Q^2 terms), and a collection of matrices that relate measurements to subsections of the outer product tensor ($P+Q-1$ matrices of $I\times Q^2$ terms). The tensor and measurement matrices dominate here and scale as $\mathcal{O}(IK^3 + K^4)$.

Basic Wirtinger methods per iteration store current best guess and scale linearly as $P+Q$. However, to achieve the computation complexity listed earlier, it is convenient to pre-compute outer products of c_i, d_i and store them. This requires a memory structure with size $IPQ \rightarrow \mathcal{O}(IK^2)$. More complex algorithms may store additional information like previous guesses or an estimate of Hessian, but are not presented here.

5.2.2. Computational complexity

Operations per iteration for each algorithm is presented in Table 1 in units of complex floats.

Matrix AltMin is a series of matrix multiplications back-projecting error from measurements to diagonals ($P + Q - 1$ matrix multiplies of size $I \times Q$) followed by an SVD ($\mathcal{O}\left(P^2Q + Q^3\right)$ listed above). SVD and matrix multiplies both contribute similar complexity, and gathering terms yields $\mathcal{O}(IK^2 + K^3)$.

Tensor AltMin is a series of matrix multiplications back-projecting error from measurements to sub-sections of a fourth-order tensor ($P + Q - 1$ matrix multiplications of size $I \times Q^2$) followed by a HOSVD

($\mathcal{O}\left(P^2Q^2\right)$ from above). Matrix multiplication and HOSVD both contribute similar complexity, and gathering terms yields $\mathcal{O}(IK^3 + K^4)$.

Gradient calculations produce an estimate of $g_i[k]$ ($2Q$ multiplies per i,k), followed by a subtraction of $e_i[k] = h_i[k] - |g_i[k]|^2$ (Q multiplies per i,k). Weighting each diagonal $\Lambda_i[k]$ is a multiplication by e_i and g_i ($2Q$ multiplies per i,k). This process is repeated four times for each subsection of the gradient to produce a total of $20Q$ operations per i,k. Accounting for each diagonal and measurement, we arrive at a total of $16Q \times (P+Q-1) \times I$ yielding a generic complexity of $\mathcal{O}(IK^2)$.

5.2.3. Scalability

Based on the computational complexity and memory complexity per iteration, gradient techniques are roughly as complex as Matrix AltMin, while tensor techniques require vastly more resources than gradient descent and Matrix AltMin. In our own studies, the resource intensity of tensor problems quickly grew to be prohibitive as is visible in the cropping in Fig. 7.

Note from Table 1 that each term depends on K or on both K and I. Based on the phase-transition diagrams presented in Fig. 7, we see that all algorithms converge when I is on the order of K (ignoring log factors). Based on the assumption that $I \approx K$, we see that Matrix AltMin and gradient techniques are roughly equal in memory complexity and are both an order of K less complex than tensor techniques.

For practical problems moving forward, gradient descent approaches offer both an efficient and scalable technique for solving for signals from the intensity of cross-correlation with tensor initializers offering slight benefit only when the problem size is small enough that the computational expense can be spared. Comparing to existing state-of-the-art algorithms built from Matrix Altmin, gradient methods are equivalently complex in both memory and computation but appear to have superior recovery rates for lower measurement count as shown in Figs. 6 and 7.

Using tensors to solve for low-dimensional approximations of x,y may be an effective way to initialize gradient descent while maintaining low computational cost, but remains a topic of future research.

5.3. Convergence

5.3.1. Toward theoretical bounds

The intensity-based cost function studied in ICC was selected in part because it is possible to take an expectation value with respect to measurement vectors c,d when these vectors are assumed to be constructed using random distributions in phase and amplitude. For this discussion, as in the numerical experiments in Section 5.1, we treat c,d as independent and identically distributed (i.i.d.) complex Gaussian vectors $c,d \sim \mathcal{CN}(0,\mathbb{I}^M), \mathcal{CN}(0,\mathbb{I}^N)$. This practice is typical in phase retrieval studies because complex Gaussian measurements provide a generic measurement type with known moments and because recovery from Gaussian measurements has been demonstrated to be near optimal [39]. Expectations can be calculated by lifting polynomial interactions in the cost function to appropriate tensor orders. Bounding deviation from this asymptotic landscape is a straightforward approach to creating theoretical requirements for the number of measurements required for convergence [28]. The gradient and Hessian of the expected value of cost can provide insight into regions over which convergence could be expected, impossible, or plagued by erroneous local minima.

Toward this end we have obtained derived the expectation of the cost function presented in Eq. (8), as well as the gradient of the expectation. The results are constructed of tightly coupled terms of x, y, x_0, y_0 and diagonals of the outer product of xy^H and $x_0y_0^H$. This tight coupling is not surprising considering that the Wirtinger gradient of cost for x,y is a linear function of y,x respectively modified by cross terms of x,y contained in $g_i[k]$. This dependence on cross-terms makes it difficult to analyze the function, and further work is needed to determine the regions of convergence of the cost function. Our derivations are presented in Appendices D and E. We also present a brief numerical study of the convergence of gradient descent in Section 5.3.2.

Outside of our studies of asymptotic cost function, we can make two comments on convergence based on the Wirtinger gradient and Hessian shown in Eqs. (9) and (10).

First, the point $x = \mathbf{0}, y = \mathbf{0}$ is a point of both zero gradient and zero Hessian. All terms in the gradient and Hessian are either directly dependent on x,y or indirectly through $g_i[k]$. This point has no gradient and no curvature and is therefore unsuitable as a starting point for our descents, prompting random initialization. This also implies that a global convergence guarantee cannot be established, though a local guarantee may still be possible.

Second, the matrix $U(x,y) = \sum_{i=0}^{I-1} \sum_{k=-M+1}^{N-1} e_i[k]g_i[k]\Lambda_i^*[k]$ contains significant geometric information of the cost function due to its ubiquity in the gradient and Hessian alike. Examining the gradient in Eq. (9), we see that gradient will go to zero at points other than ambiguities of the ground truth if x,y are in the left and right-hand null-space of $U(x,y)$ respectively. Were it not for the presence of $g_i[k]$, this problem would largely be solvable with statements about the relationship between x,y and c,d vectors; however this $g_i[k]$ term modifies what would be an ordinary quadratic $\langle \mathrm{diag}(c_i, d_i^H, k), xy^H \rangle$ to a higher-order interaction. We believe that understanding the behavior of $U(x,y)$ and its variations present in the Hessian is critical to understanding the regions of convergence.

5.3.2. Numerical convergence analysis

By fitting models against the phase transition plots shown in Fig. 7, we aim to provide some estimate of the convergence rate of randomly initialized Wirtinger descent using the procedure outlined below.

1. For each M,N fit the phase transition plot for randomly initialized Wirtinger descent to a generalized logistic function (GLF) as a function of I:

$$GLF(I,\ a,\ b,\ c) = \frac{1}{(1+e^{-a*(I-b)})^{1/c}}. \tag{12}$$

2. For every GLF function fit, determine where the GLF crosses the points .25, .5, and .75 to provide estimates of the 25%, 50%, and 75% convergence rates of randomly initialized ICC problems converging to a correct answer.
3. Fit a linear, square-root, and log-linear function to these crossings as a function of M,N, and determine which function fits data with least mean-squared error.

The GLF function was used here to approximate columns of the phase transition plot because of its low complexity and generalizability for monotonic functions transitioning between 0 and 1. The linear, square-root, and log-linear functions fit to resulting GLF crossings are outlined in Eqs. (13), (14), and (15). All variables that are not M are free variables.

$$f_{linear}(M) = aM + b \tag{13}$$

$$f_{sqrt}(M) = \sqrt{aM+b} + c \tag{14}$$

$$f_{loglin}(M) = \log(aM+b)(cM+d) \tag{15}$$

Functions in Eqs. (13), (14), and (15) were fit to the resulting GLF crossings and the residual mean squared error is listed in Table 2. The log-linear function fits the crossing data with least error consistently for the 25%, 50%, and 75% convergence rate. The fits extracted for the log-linear function are plotted over the phase-transition diagram in Fig. 9.

We suspect based off these brief numerical experiments that the number of measurements I required to converge with constant success rate is approximately log-linear with respect to problem complexity M,N with parameters listed in Table 3. Note that the additive parameter b in the log-linear expression consistently solved to be insignificant and is not listed.

Table 2
Residual MSE for various fit functions vs. convergence rate.

	25%	50%	75%
Linear	2.54	2.42	2.94
Square-root	1.06	1.43	2.91
Log-linear	.76	.92	1.82

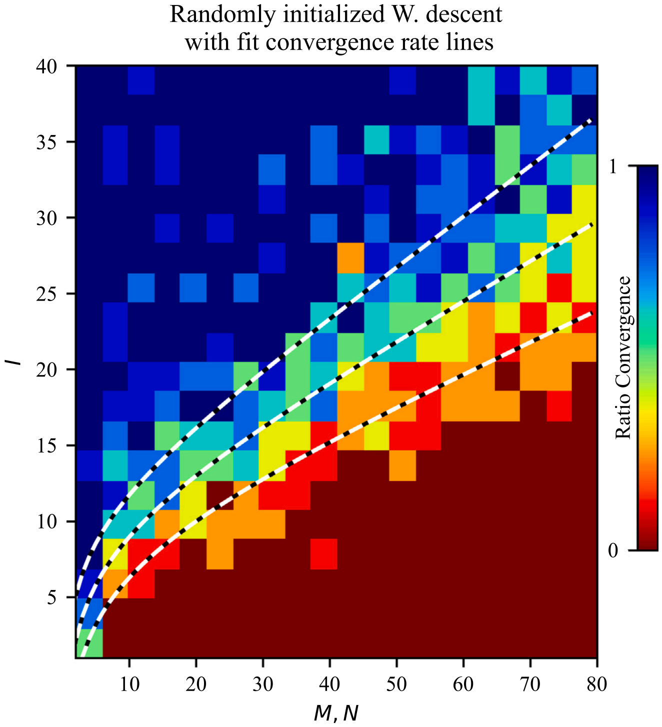

Fig. 9. 25%, 50%, and 75% log-linear functions plotted against the phase-transition plot. Note the phase transition plot shows the average number of trials that converge to within 1% relative error of ground truth.

Table 3
Measurement count per convergence rate fit to log-linear function of problem complexity.

$I_{25\%}$	$\log(.44M)(.035M + 3.9)$
$I_{50\%}$	$\log(.89M)(.041M + 3.7)$
$I_{75\%}$	$\log(2.3M)(.047M + 3.3)$

5.4. Amplitude cost

Recent advances in gradient based phase retrieval have studied nuanced initializers, reweighting and truncation steps, and additional cost functions [22,23]. Useful comparisons of existing reweighting, truncation, and cost function variations have been studied in [19,25]. One of the more broadly recognized improvements to gradient techniques has studied the transition to an amplitude cost function like that in Eq. (17) as opposed to an intensity cost function like the one shown in Eq. (16).

$$l_{int}(z) = \frac{1}{2M}\sum_{i=1}^{M}\left(y[i] - |A[i,:]z|^2\right)^2 \tag{16}$$

$$l_{amp}(z) = \frac{1}{2M}\sum_{i=1}^{M}\left(\sqrt{y[i]} - |A[i,:]z|\right)^2 \tag{17}$$

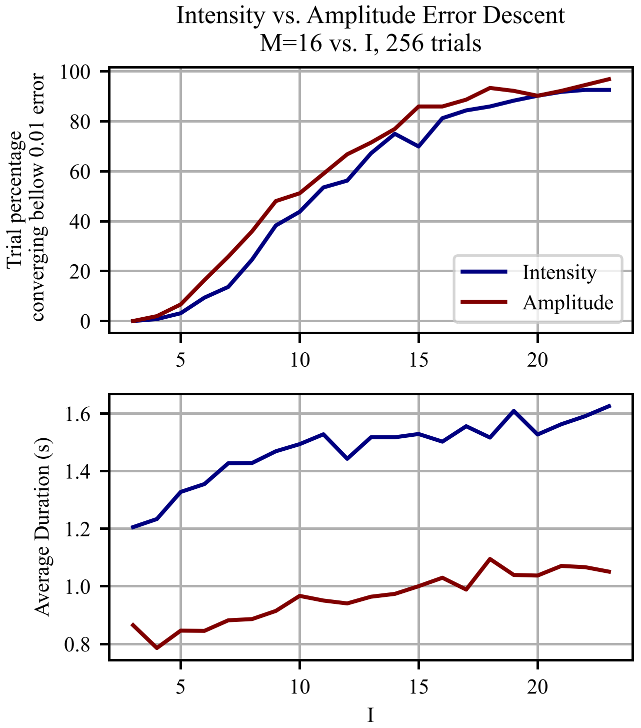

Fig. 10. Over 256 trials, amplitude and intensity based cost functions solve for objective variables from phaseless modulated cross-correlation measurements. The top plot shows the percentage of trials that meet a minimum ground truth error requirement, and the bottom plot shows the average duration for each gradient descent.

Despite performance improvements demonstrated in ordinary phase retrieval by amplitude techniques, the research presented in this paper examines unweighted intensity based approaches because the relative simplicity of the intensity cost function seemed a more straightforward route toward theoretical guarantees via geometric analysis. As demonstrated in Appendices D and E, the intensity cost function can be studied in the asymptotic landscape with expectations because its polynomial admits an expectation when lifted to a tensor inner product. The amplitude construction cannot be made into a linear inner product by lifting because it is not constructed from polynomials.

While this paper primarily addresses the efficacy of an intensity based cost function presented in Eq. (8), it is worth demonstrating that amplitude based cost functions present a viable approach to the pulse characterization problem. To demonstrate this, we devised a head-to-head comparison of the cost function in Eq. (8) and a cost function formulated in Eq. (18).

$$f(x,y) = \frac{1}{2}\sum_{i=0}^{I-1}\sum_{k=-M+1}^{N-1}\left(\sqrt{h_i[k]} - \sqrt{x^T\Lambda_i[k]y^*x^H\Lambda_i^*[k]y}\right)^2. \tag{18}$$

We run an experiment on a fixed problem size $M = 16$ for varying measurement counts I and record the error of their solutions as well as the run-time over many trials. Each experiment is initialized randomly and ground truth and measurements are constructed from i.i.d. complex Gaussian random variables. Both intensity and amplitude gradient descents are initialized from precisely the same x_{init}, y_{init}. To provide a fair head-to-head for the intensity and amplitude cost functions, gradient is estimated numerically with identical function parameters using the L-BFGS-B algorithm in the Scipy optimization library [29] as equations for amplitude gradient have not yet been verified and implemented. The measurements used in this experiment are provided to each algorithm without noise.

Results of this experiment demonstrated in Fig. 10 show that the amplitude based cost does marginally improve the percentage of trials that converge to ground truth over the intensity based cost function. Amplitude based trials appear to converge on their solution at a faster rate, typically in about two thirds the amount of time of the intensity approach in this experiment. This result encourages future examination of amplitude based cost functions applied to the pulse characterization problem.

6. DSCAN

DSCAN [16] is a simple time-resolved phaseless auto-convolution technique that modulates the spectrum of a pulse with a known phase-profile. This pulse is squared in its time domain and the spectral intensity is measured at a spectrometer, resulting in the intensity of modulated pulse spectrum auto-convolution. Our technique is adapted to this problem where x_0 and y_0 (and conversely, c, d) are conjugate time reversals of each other.

6.1. Prior art comparison

Wilhelm et al. [16] introduce techniques in DSCAN to resolve multi-mode phase retrieval. Multi-mode retrieval is a generalized problem to ICC in which multiple pulses exist in a pulse train (i.e, multiple $x_0, x_1, \ldots$ and $y_0, y_1, \ldots$) and integrate in the same measurements $h_i[k]$. Multi-mode may be possible with techniques in this paper, for example by changing the rank solved for in iterative algorithms and concatenating additional x, y as descent variables, but is not tested in this paper.

When applied to singular modes, as done in this paper, the technique presented in [16] is an alternating minimization (referred to in this section as AltMin) that uses a forward and backward model to iterate between applying an average in the pulse spectral domain and applying a magnitude constraint in the measurement domain. With a minor adaptation, our algorithm can be compared directly with this algorithm to determine its suitability for pulse retrieval.

Each algorithm was seeded identically, starting with a vector of $\mathbf{1}$. Pulses of length $N = 64$ were generated with a Gaussian magnitude envelope and random spline phase profile. Grating position bounds on DSCAN generation were fixed to 1 mm in each scan direction. With fixed scan bounds, scan steps span the measurement range uniformly with gradually increasing step counts between 20 and 100 grating positions. Results presented here are conducted for the noiseless case. Results comparing error in retrieved x_0 are shown in Fig. 11 against identical synthetic data. Results show that Wirtinger descent has more consistent success at lower measurement count than AltMin. In Fig. 11, some final errors displayed in the histogram for AltMin are lower than Wirtinger descent different because of the termination criteria tuning. In this experiment, success rates for Wirtinger descent begin to decline for higher measurement count. It may be that decreasing distinction between c_i vectors by decreasing step size (DSCAN grating boundaries are fixed) complicates convergence, but the cause is still under investigation.

6.2. Evaluation on real data

Finally, we have applied Wirtinger descent to the single mode dataset used in [16]. The data set represents a spectrometer intensity measurement at 500 positions of the grating screen in the DSCAN system; these positions are symmetric about its zero position and stepped across 5 mm in each direction. Grating density d is 1400 lines/mm and angles in and out of the system were held at 34.05°. Spectral density of the measurement permitted a recoverable pulse width of $M = 320$ samples. The initial vector state for both the Wirtinger descent and AltMin was held as a complex vector of $x = \mathbf{1}^M$. At convergence, the recovered vector was able to regenerate the measured dataset to 4.3% Frobenius error, with input and recreated data set presented in Fig. 12. AltMin is able to regenerate measured dataset to 10.1% Frobenius error.

When the ambiguous linear component of the spectral phase is subtracted off, the recovered vector highly resembles the AltMin recovery and accompanying spectrometer measurement shown in Fig. 13. The recovered trace comparison in shows the efficacy of the Wirtinger descent techniques of this paper as a drop in alternative to existing recovery algorithms.

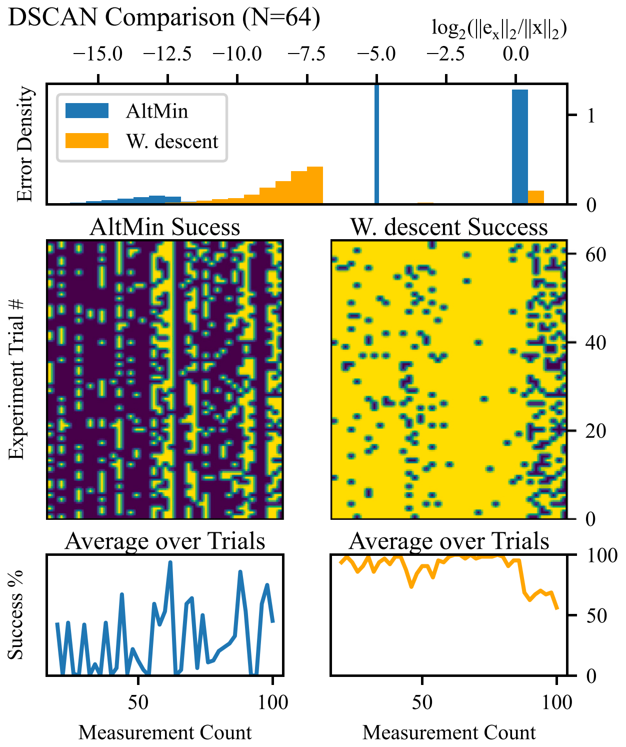

Fig. 11. Alternating minimization current state-of-the-art is compared with Wirtinger descent presented in this paper. Final results of each algorithm are binned and displayed in the top figure; failure and success distributions are on either side of a chosen success threshold 2^{-5}. Over 64 trials, each algorithm was seeded identically with identical data, and their ability to return a result below the threshold is displayed against measurement count I in the two middle figures. Their average (over the vertical dimension of the middle figures) is shown in the bottom two plots.

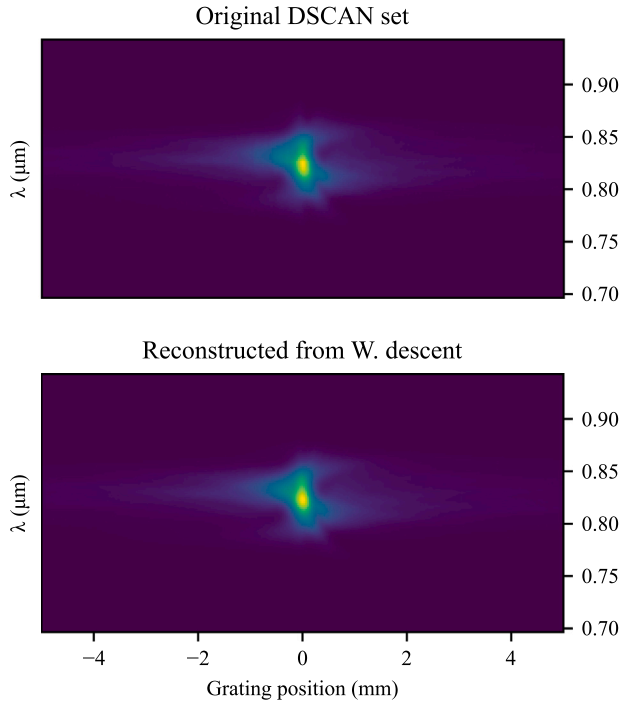

Fig. 12. Following the practice of pulse recovery papers [8,12,14,16], we show the measured DSCAN dataset against the forward model for the DSCAN setup on our recovered spectrum. The two results are nearly identical, and differ in Frobenius norm by only 4.3%.

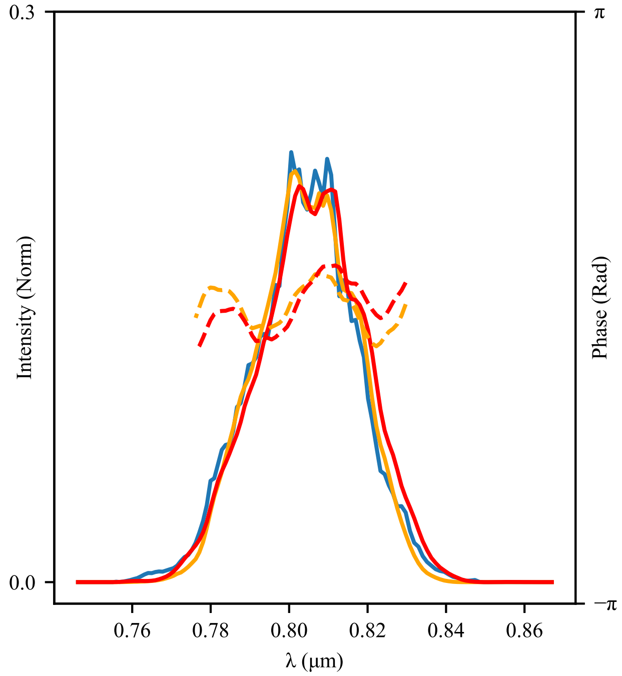

Fig. 13. Two algorithms recover the spectral profile of a pulse from DSCAN measurements. The solid blue line represents the spectrometer measurement of the intensity of the spectrum of the pulse, a direct measurement of ground truth intensity for the spectrum. The solid orange line shows the recovery of intensity made by the alternating minimization algorithm presented in [16]. The broken orange line is the phase of the recovered pulse. The solid red line is the intensity profile recovered by Wirtinger descent initialized with Algorithm 6. The broken red line is the recovered pulse phase using Wirtinger descent.

7. Conclusion

In this paper, we adapted the phase retrieval techniques of Wirtinger descent to a new intensity of cross-correlation problem. We highlighted a key difference in this problem from existing low-rank phase retrieval literature in the lack of phase-coherence between the recovered diagonals of the outer product matrix xy^H. This lack of coherence makes traditional spectral initializers fail and motivates techniques that are able to combine this phase information coherently. We achieved this using a novel low-rank tensor initializer based on iterative hard tensor thresholding. We demonstrated that when used as an initializer to Wirtinger descent, low-rank tensor initialization is superior to matrix alternating minimization and adapted spectral initializers with respect to measurement count at the expense of greater computational burden. Wirtinger descent over intensity based cost functions were demonstrated to out-perform state-of-the-art matrix alternating-minimization by inverting ICC problems with fewer measurements for comparable memory and computational cost. Finally we demonstrate the ability of our recovery algorithms to operate on real data, recovering objective variables comparable to current state-of-the-art with little adaptation.

Further analysis of the forward operator is in progress, where examinations of the geometric landscape of the Wirtinger objective function may lend insight into convergence properties of Wirtinger descent in the style of [28].

CRediT authorship contribution statement

Dan Rosen: Methodology, Conceptualization, Software, Investigation, Writing – original draft. **Michael B. Wakin:** Conceptualization, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

Our collaboration with the Squier Optics Group at Colorado School of Mines [40] was the primary motivator to explore algorithms for pulse characterization. We are grateful to the group for helpful conversations and to Daniel Scarbrough and Alex Wilhelm who collected and shared the original DSCAN data. We are also grateful to Gongguo Tang for his expertise and mentoring in numerical optimization techniques. This work was partially supported by National Science Foundation, USA grants CCF-1704204 and CCF-2106834.

Appendix A. Cost, gradient, hessian with a single variable

Concatenating the variables x and y into one vector $z = \left[x^T,\ y^T\right]^T \in \mathbb{C}^{M+N}$, we may rewrite the cost function (8) simply as

$$\begin{aligned} f(z) &= \frac{1}{2}\sum_{i,k}\left(h_i[k] - z^T\beta_i[k]z^*z^H\beta_i^*[k]z\right)^2 \\ &= \frac{1}{2}\sum_{i,k} e_i[k]^2, \end{aligned} \tag{19}$$

where

$$\beta_i[k] = \begin{pmatrix} \mathbf{0}_{M\times M} & \Lambda_i[k] \\ \mathbf{0}_{N\times M} & \mathbf{0}_{N\times N} \end{pmatrix}. \tag{20}$$

This new cost function is quartic with respect to the variable z and has a well defined Wirtinger gradient and Hessian. Similarly, we may express $g_i[k]$ (defined above (8) as $x^T\Lambda_i[k]y^*$) in terms of z as $g_i[k] = z^T\beta_i[k]z^*$.

Producing a Wirtinger gradient will proceed much in the same manner as [28]. The Wirtinger gradient of $f(z)$, with z complex-valued, is defined as

$$\nabla f(z) = \begin{pmatrix} \frac{\partial f}{\partial z} & \frac{\partial f}{\partial z^*} \end{pmatrix}^H. \tag{21}$$

Note that the partial derivatives in (21) are taken to be row vectors. Moreover, according to the Wirtinger calculus, the partial derivative $\frac{\partial f}{\partial z}$ is computed assuming z^* (and z^H) is fixed, while the partial derivative $\frac{\partial f}{\partial z^*}$ is computed assuming z (and z^T) is fixed [41].

Differentiating $f(z)$ in (19) with respect to z yields

$$\frac{\partial f}{\partial z} = \frac{\partial}{\partial z}\frac{1}{2}\sum_{i,k} e_i[k]^2 = e_i[k]\frac{\partial e_i[k]}{\partial z}, \tag{22}$$

where

$$\begin{aligned} \frac{\partial e_i[k]}{\partial z} &= -z^T\left(\beta_i[k]z^*z^H\beta_i^*[k] + \beta_i^H[k]z^*z^H\beta_i^T[k]\right) \\ &= -\left(g_i[k]z^H\beta_i^*[k] + g_i^*[k]z^H\beta_i^T[k]\right). \end{aligned} \tag{23}$$

Differentiating with respect to z^* yields

$$\frac{\partial f}{\partial z^*} = \frac{\partial}{\partial z^*}\frac{1}{2}\sum_{i,k} e_i[k]^2 = e_i[k]\frac{\partial e_i[k]}{\partial z^*}, \tag{24}$$

where

$$\begin{aligned} \frac{\partial e_i[k]}{\partial z^*} &= -z^H\left(\beta_i^*[k]zz^T\beta_i[k] + \beta_i^T[k]zz^T\beta_i^H[k]\right) \\ &= -\left(g_i^*[k]z^T\beta_i[k] + g_i[k]z^T\beta_i^H[k]\right). \end{aligned} \tag{25}$$

Collecting terms, we arrive at the following expression for the Wirtinger gradient:

$$\nabla f(z) = \begin{pmatrix} {\frac{\partial f}{\partial z}}^H \\ {\frac{\partial f}{\partial z^*}}^H \end{pmatrix} = \sum_{i,k} e_i[k] \begin{pmatrix} {\frac{\partial e_i[k]}{\partial z}}^H \\ {\frac{\partial e_i[k]}{\partial z^*}}^H \end{pmatrix}, \tag{26}$$

where

$$\begin{pmatrix} {\frac{\partial e_i[k]}{\partial z}}^H \\ {\frac{\partial e_i[k]}{\partial z^*}}^H \end{pmatrix} = - \begin{pmatrix} g_i[k]\beta_i^*[k]z + g_i^*[k]\beta_i^T[k]z \\ g_i^*[k]\beta_i[k]z^* + g_i[k]\beta_i^H[k]z^* \end{pmatrix}.$$

The Wirtinger Hessian of $f(z)$, with z complex-valued, is defined as

$$\nabla^2 f(z) = \begin{pmatrix} \frac{\partial}{\partial z}\left(\frac{\partial f}{\partial z}\right)^H & \frac{\partial}{\partial z^*}\left(\frac{\partial f}{\partial z}\right)^H \\ \frac{\partial}{\partial z}\left(\frac{\partial f}{\partial z^*}\right)^H & \frac{\partial}{\partial z^*}\left(\frac{\partial f}{\partial z^*}\right)^H \end{pmatrix} \tag{27}$$

Using (26), we have that the individual components of the Wirtinger Hessian are given by

$$\frac{\partial}{\partial z}\left(\frac{\partial f}{\partial z}\right)^H = \sum_{i,k} {\frac{\partial e_i[k]}{\partial z}}^H \frac{\partial e_i[k]}{\partial z} + e_i[k]\frac{\partial}{\partial z}{\frac{\partial e_i[k]}{\partial z}}^H, \tag{28}$$

$$\frac{\partial}{\partial z^*}\left(\frac{\partial f}{\partial z}\right)^H = \sum_{i,k} {\frac{\partial e_i[k]}{\partial z}}^H \frac{\partial e_i[k]}{\partial z^*} + e_i[k]\frac{\partial}{\partial z^*}{\frac{\partial e_i[k]}{\partial z}}^H, \tag{29}$$

$$\frac{\partial}{\partial z}\left(\frac{\partial f}{\partial z^*}\right)^H = \sum_{i,k} {\frac{\partial e_i[k]}{\partial z^*}}^H \frac{\partial e_i[k]}{\partial z} + e_i[k]\frac{\partial}{\partial z}{\frac{\partial e_i[k]}{\partial z^*}}^H, \tag{30}$$

and

$$\frac{\partial}{\partial z^*}\left(\frac{\partial f}{\partial z^*}\right)^H = \sum_{i,k} {\frac{\partial e_i[k]}{\partial z^*}}^H \frac{\partial e_i[k]}{\partial z^*} + e_i[k]\frac{\partial}{\partial z^*}{\frac{\partial e_i[k]}{\partial z^*}}^H. \tag{31}$$

The partial derivatives of $e_i[k]$ with respect to z and z^* appear in Eqs. (28)–(31) and have already been computed in Eqs. (23) and (25). Computing the remaining terms in Eqs. (28)–(31) requires the following second derivatives:

$$\begin{aligned} \frac{\partial}{\partial z}{\frac{\partial e_i[k]}{\partial z}}^H &= - g_i[k]\beta_i^*[k] - g_i^*[k]\beta_i^T[k] \\ &\quad - \beta_i^*[k]zz^H\beta_i^T[k] - \beta_i^T[k]zz^H\beta_i^*[k], \end{aligned} \tag{32}$$

$$\frac{\partial}{\partial z}{\frac{\partial e_i[k]}{\partial z^*}}^H = - \beta_i^H[k]z^*z^H\beta_i^T[k] - \beta_i[k]z^*z^H\beta_i^*[k], \tag{33}$$

$$\frac{\partial}{\partial z^*}{\frac{\partial e_i[k]}{\partial z}}^H = - \beta_i^*[k]zz^T\beta_i[k] - \beta_i^T[k]zz^T\beta_i^H[k], \tag{34}$$

$$\begin{aligned} \frac{\partial}{\partial z^*}{\frac{\partial e_i[k]}{\partial z^*}}^H &= - g_i[k]\beta_i^H[k] - g_i^*[k]\beta_i[k] \\ &\quad - \beta_i[k]z^*z^T\beta_i^H[k] - \beta_i^H[k]z^*z^T\beta_i[k]. \end{aligned} \tag{35}$$

Finally, it is very useful to decompose the Wirtinger Hessian into two sections based on their structure within the sum. As shown below, one of these terms is a direct sum of outer products and the other is a sum that contains several terms that decompose to an outer product. We have:

$$\nabla^2 f(z) = A_1 + A_2 \tag{36}$$

where

$$A_1 = \sum_{i,k} \begin{pmatrix} {\frac{\partial e_i[k]}{\partial z}}^H \frac{\partial e_i[k]}{\partial z} & {\frac{\partial e_i[k]}{\partial z}}^H \frac{\partial e_i[k]}{\partial z^*} \\ {\frac{\partial e_i[k]}{\partial z^*}}^H \frac{\partial e_i[k]}{\partial z} & {\frac{\partial e_i[k]}{\partial z^*}}^H \frac{\partial e_i[k]}{\partial z^*} \end{pmatrix} = \sum_{i,k} a_1 a_1^H$$

with

$$a_1 = \begin{pmatrix} {\frac{\partial e_i[k]}{\partial z}}^H \\ {\frac{\partial e_i[k]}{\partial z^*}}^H \end{pmatrix} = - \begin{pmatrix} g_i[k]\beta_i^*[k]z + g_i^*[k]\beta_i^T[k]z \\ g_i^*[k]\beta_i[k]z^* + g_i[k]\beta_i^H[k]z^* \end{pmatrix},$$

and

$$\begin{aligned} A_2 &= \sum_{i,k} e_i[k] \begin{pmatrix} \frac{\partial}{\partial z}{\frac{\partial e_i[k]}{\partial z}}^H & \frac{\partial}{\partial z^*}{\frac{\partial e_i[k]}{\partial z}}^H \\ \frac{\partial}{\partial z}{\frac{\partial e_i[k]}{\partial z^*}}^H & \frac{\partial}{\partial z^*}{\frac{\partial e_i[k]}{\partial z^*}}^H \end{pmatrix} \\ &= - \sum_{i,k} e_i[k]\left(a_{2,1}a_{2,1}^H + a_{2,2}a_{2,2}^H + A_{2,3}\right) \end{aligned}$$

with

$$a_{2,1}, a_{2,2} = \begin{pmatrix} \beta_i^*[k]z \\ \beta_i^H[k]z^* \end{pmatrix}, \begin{pmatrix} \beta_i^T[k]z \\ \beta_i[k]z^* \end{pmatrix}$$

$$A_{2,3} = \begin{pmatrix} \beta_i^T[k]g_i^*[k] + \beta_i^*[k]g_i[k] & \mathbf{0} \\ \mathbf{0} & \beta_i[k]g_i^*[k] + \beta_i^H[k]g_i[k] \end{pmatrix}.$$

Appendix B. Cost function ambiguities

Let x_a and y_a be as defined in (11), and note that to prove $f(x,y) = f(x_a, y_a)$ for f defined in (8), it suffices to prove that

$$x^T \Lambda_i[k]y^* x^H \Lambda_i^*[k]y = x_a^T \Lambda_i[k]y_a^* x_a^H \Lambda_i^*[k]y_a$$

for all i, k. To that end, write

$$\begin{aligned} &x_a^T \Lambda_i[k]y_a^* x_a^H \Lambda_i^*[k]y_a \\ &= \sum_{r=l[k]}^{u[k]} \sum_{s=l[k]}^{u[k]} x_a[r]c_i[r]y_a^*[r+k]d_i^*[r+k] \cdot x_a^*[s]c_i^*[s]y_a[s+k]d_i[s+k] \\ &= \sum_{r=l[k]}^{u[k]} \sum_{s=l[k]}^{u[k]} x_a[r]x_a^*[s]y_a[s+k]y_a^*[r+k] \cdot c_i[r]c_i^*[s]d_i[s+k]d_i^*[r+k] \\ &= \sum_{r=l[k]}^{u[k]} \sum_{s=l[k]}^{u[k]} \frac{|a|^2}{|a|^2} x[r]x^*[s]y[s+k]y^*[r+k]e^{j\zeta} \cdot c_i[r]c_i^*[s]d_i[s+k]d_i^*[r+k] \\ &= \sum_{r=l[k]}^{u[k]} \sum_{s=l[k]}^{u[k]} x[r]x^*[s]y[s+k]y^*[r+k] \cdot c_i[r]c_i^*[s]d_i[s+k]d_i^*[r+k] \\ &= x^T \Lambda_i[k]y^* x^H \Lambda_i^*[k]y, \end{aligned}$$

where the fourth equality follows because

$$\zeta = \phi_x - \phi_x + \phi_y - \phi_y + s\psi - s\psi + r\psi - r\psi + k\psi - k\psi = 0.$$

Appendix C. Adaptation to real solvers

For numerical experiments, expressions for gradient and Hessian were provided to Scipy's minimize function [29]. Because the function does not optimize over complex variables, the following translations from Wirtinger gradient and Hessian to real and imaginary portions of the gradient and Hessian are provided in Eqs. (37) and (38) for a generic complex function $f(z)$.

$$\nabla f_{real,imag}(z) = \begin{pmatrix} \nabla_z f + \nabla_{z^*} f \\ j\left(\nabla_{z^*} f - \nabla_z f\right) \end{pmatrix} \tag{37}$$

$$\begin{aligned} \nabla^2 f_{real,imag}(z) &= \begin{pmatrix} f_1 + f_2 + f_3 + f_4 & j\left(f_2 + f_4 - f_1 - f_3\right) \\ j\left(f_1 + f_2 - f_3 - f_4\right) & f_1 + f_4 - f_2 - f_3 \end{pmatrix} \\ f_1,\ f_2 &= \frac{\partial}{\partial z}\left(\nabla_z f\right),\ \frac{\partial}{\partial z^*}\left(\nabla_z f\right) \\ f_3,\ f_4 &= \frac{\partial}{\partial z}\left(\nabla_{z^*} f\right),\ \frac{\partial}{\partial z^*}\left(\nabla_{z^*} f\right) \end{aligned} \tag{38}$$

Here, $\nabla_z f, \nabla_{z^*} f$ are the first and second half of the Wirtinger gradient ∇f, differentiated respectively with respect to z, z^* then conjugate-transposed.

Appendix D. Expectation of intensity-based cost function

Determining the expectation of the cost function presented in Eq. (8) is done by lifting the non-linear function to a linear inner product between higher-order tensors. The linearity that allows this expectation to be evaluated is demonstrated in the tensor notation shown in Section 2.5.3. Here we assume c_i, d_i are i.i.d. and drawn from $\mathcal{CN}(0, \mathbb{I})$. We start from the cost function and gradually replace terms in Eq. (39). To reduce reuse of outer product operators, throughout this section we interpret concatenation of tensor objects to be a tensor outer product.

Recall here that $\mathcal{X} = x \otimes y^* \otimes x^* \otimes y$, $\mathcal{X}_0 = x_0 \otimes y_0^* \otimes x_0^* \otimes y_0$, and $\mathcal{L}_i^k = \Lambda_i^*[k] \otimes \Lambda_i[k]$.

$$\begin{aligned} f(x,y) =& \frac{1}{2}\sum_{i=0}^{I-1}\sum_{k=-M+1}^{N-1} \left(h_i[k] - x^T \Lambda_i[k] y^* x^H \Lambda_i^*[k] y\right)^2 \\ =& \frac{1}{2}\sum_{i=0}^{I-1}\sum_{k=-M+1}^{N-1} \left(\langle \mathcal{L}_i^k,\ \mathcal{X}_0\rangle - \langle \mathcal{L}_i^k,\ \mathcal{X}\rangle\right)^2 \\ =& \frac{1}{2}\sum_{i=0}^{I-1}\sum_{k=-M+1}^{N-1} \langle \mathcal{L}_i^k \mathcal{L}_i^k,\ \mathcal{X}_0\mathcal{X}_0 +\ \mathcal{X}\mathcal{X} -\ \mathcal{X}_0\mathcal{X} -\ \mathcal{X}\mathcal{X}_0\rangle \\ =& \left\langle \frac{1}{2}\sum_{i=0}^{I-1}\sum_{k=-M+1}^{N-1} \mathcal{L}_i^k \mathcal{L}_i^k,\ \mathcal{X}_0\mathcal{X}_0 +\ \mathcal{X}\mathcal{X} -\ \mathcal{X}_0\mathcal{X} -\ \mathcal{X}\mathcal{X}_0 \right\rangle \end{aligned} \tag{39}$$

$$\begin{aligned} \mathbb{E}\left[f(x,y)\right] =& \left\langle \mathbb{E}\left[\frac{1}{2}\sum_{i=0}^{I-1}\sum_{k=-M+1}^{N-1} \mathcal{L}_i^k \mathcal{L}_i^k\right],\ \mathcal{X}_0\mathcal{X}_0 +\ \mathcal{X}\mathcal{X} \right\rangle \\ &- \left\langle \mathbb{E}\left[\frac{1}{2}\sum_{i=0}^{I-1}\sum_{k=-M+1}^{N-1} \mathcal{L}_i^k \mathcal{L}_i^k\right],\ \mathcal{X}_0\mathcal{X} +\ \mathcal{X}\mathcal{X}_0 \right\rangle \\ =& \langle \mathbb{L},\ \mathcal{X}_0\mathcal{X}_0 +\ \mathcal{X}\mathcal{X} -\ \mathcal{X}_0\mathcal{X} -\ \mathcal{X}\mathcal{X}_0\rangle \end{aligned} \tag{40}$$

The expectation of this problem shown in Eq. (40) then solely revolves on the expectation of the eighth-order tensor $\frac{1}{2}\sum_{k=-M+1}^{N-1}(\mathcal{L}_i^k \mathcal{L}_i^k)$. While the tensor $(\mathcal{L}_i^k \mathcal{L}_i^k)$ has eight indexing variables, each variable is coupled with another as a function of k, meaning that the only non-zero entries of the tensor can be written in Eq. (41).

$$\begin{aligned} (\mathcal{L}_i^k \mathcal{L}_i^k)[q,q,r,r,s,s,t,t] =& c_i^*[q] d_i[q+k] c_i[r] d_i^*[r+k] \\ & \times c_i^*[s] d_i[s+k] c_i[t] d_i^*[t+k] \\ & q,r,s,t = [-\min(0,k), \dots, \min(M, N-k) - 1] \end{aligned} \tag{41}$$

The uniform phase distribution of the complex random vectors c, d gives an expectation of zero for any collection of variables that are of odd conjugate matching. This leaves only three combinations of q, r, s, t in Eq. (41) that result in non-zero expectations:

1. $q = r = s = t$ which produces $A_1 = \mathbb{E}\left[|c_q|^4 |d_{q+k}|^4\right]$ when $\mathbb{E}_i\left[(\mathcal{L}_i^k \otimes \mathcal{L}_i^k)[q,q,q,q,q,q,q,q]\right]$
2. $q = r \neq s = t$ which produces $A_2 = \mathbb{E}\left[|c_q|^2 |c_s|^2 \| d_{q+k}|^2 d_{s+k}|^2\right]$ when $\mathbb{E}_i\left[(\mathcal{L}_i^k \otimes \mathcal{L}_i^k)[q,q,q,q,s,s,s,s]\right]$
3. $q = t \neq r = s$ which produces $A_2 = \mathbb{E}\left[|c_q|^2 |c_r|^2 \| d_{q+k}|^2 d_{r+k}|^2\right]$ when $\mathbb{E}_i\left[(\mathcal{L}_i^k \otimes \mathcal{L}_i^k)[q,q,r,r,r,r,q,q]\right]$

Under our assumption that c, d are drawn i.i.d. from $\mathcal{CN}(0, \mathbb{I})$, $A_1, A_2 = 4, 1$. Note here that in the expectation, the sum over i multiplies now just a multiplication by I. Collecting indicator terms, we arrive at an expression for $\mathbb{L}$ in Eq. (42). Here, concatenated e_i indexing terms are joined as an outer product.

$$\begin{aligned} \mathbb{L} =& I\ (\mathbb{L}_0 + \mathbb{L}_1 + \mathbb{L}_2) \\ \mathbb{L}_0 =& \frac{A_1 - 2A_2}{2} \sum_{k=-M+1}^{N-1} \sum_{q=-\min(0,k)}^{\min(M,N-k)-1} e_q\ e_{q+k}\ e_q\ e_{q+k}\ e_q\ e_{q+k}\ e_q\ e_{q+k} \\ \mathbb{L}_1 =& \frac{A_2}{2} \sum_{k=-M+1}^{N-1} \sum_{q=-\min(0,k)}^{\min(M,N-k)-1} \sum_{s=-\min(0,k)}^{\min(M,N-k)-1} e_q\ e_{q+k}\ e_q\ e_{q+k}\ e_s\ e_{s+k}\ e_s\ e_{s+k} \\ \mathbb{L}_2 =& \frac{A_2}{2} \sum_{k=-M+1}^{N-1} \sum_{q=-\min(0,k)}^{\min(M,N-k)-1} \sum_{r=-\min(0,k)}^{\min(M,N-k)-1} e_q\ e_{q+k}\ e_r\ e_{r+k}\ e_r\ e_{r+k}\ e_q\ e_{q+k} \end{aligned} \tag{42}$$

We note here that in Eq. (42), $\mathbb{L}_0$ donates a term to $\mathbb{L}_1$ and $\mathbb{L}_2$ to simplify the sums and avoid the use of indicator expressions that prevent $q = s, q = r$ respectively. Evaluating Eq. (40) with $A_1, A_2 = 4, 1$ with the expression for expectation in Eq. (42), we arrive at the following expectation of the intensity cost function in Eq. (43).

$$\begin{aligned} \mathbb{E}[f(x,y)] =& I \| |x_0 y_0^H|^2 - |x y^H|^2 \|_F^2 \\ &+ I \sum_{k=-M+1}^{N-1} \left(\|z_0^k\|_2^4 + \|z^k\|_2^4 - \|z_0^k\|_2^2 \|z^k\|_2^2 - |\langle z_0^k, z^k\rangle|^2\right) \end{aligned} \tag{43}$$

The expression in Eq. (43) is dense with functions over $z^k = \mathrm{diag}(x y^H, k)$, $z_0^k = \mathrm{diag}(x_0 y_0^H, k)$ as well as absolute values of point-wise multiplications between x, y, x_0, y_0, a difficulty that carries on to the gradient of this function.

Appendix E. Gradient of expectation

Derivation of Wirtinger gradient of the expectation of ICC cost is shown in Eq. (44). Because the expressions are not cleanly representable as vectors, partial derivatives F_x, F_y shown in Eqs. (45) and (46) are indexed by i, j respectively. Only F_x and F_y are presented here for brevity; F_{x^*} and F_{y^*} will be their conjugate respectively as Eq. (40) is a real valued function. As a shorthand here, we use concatenation to imply outer products (e.g. $x \otimes y^* \otimes x^* \otimes y \rightarrow x y^* x^* y$).

$$\begin{aligned} \nabla \mathbb{E}[f(x,y)] = \begin{pmatrix} \left(\frac{\partial}{\partial x}\mathbb{E}[f(x,y)]\right)^* [i] \\ \left(\frac{\partial}{\partial y}\mathbb{E}[f(x,y)]\right)^* [j] \\ \left(\frac{\partial}{\partial x^*}\mathbb{E}[f(x,y)]\right)^* [i] \\ \left(\frac{\partial}{\partial y^*}\mathbb{E}[f(x,y)]\right)^* [j] \end{pmatrix} = \begin{pmatrix} F_x[i] \\ F_y[j] \\ F_{x^*}[i] \\ F_{y^*}[j] \end{pmatrix} \\ i, j = [0, 1, \dots, M-1], [0, 1, \dots, N-1] \end{aligned} \tag{44}$$

where

$$\begin{aligned} F_x[i] =& \langle \mathbb{L}[i,:,:,:,:,:,:,:], y x y^* (x^* y x y^* - x_0^* y_0 x_0 y_0^*)\rangle \\ &+ \langle \mathbb{L}[:,:,:,:,i,:,:,:], (x^* y x y^* - x_0^* y_0 x_0 y_0^*) y x y^*\rangle \\ =& 2\ I\ x[i] \left(|x[i]|^2 \|y\|_4^4 - |x_0[i]|^2 \|y_0 y\|_2^2\right) \\ &+ I\ x[i] \sum_{k=-i}^{N-i-1} |y[i+k]|^2 \left(\|z^k\|_2^2 - \|z_0^k\|_2^2\right) \\ &+ I\ x[i] \sum_{k=-i}^{N-i-1} |y[i+k]|^2 \|z^k\|_2^2 \\ &- I\ x_0[i] \sum_{k=-i}^{N-i-1} y[i+k] y_0^*[i+k] (z_0^{kH} z^k) \end{aligned} \tag{45}$$

and

$$\begin{aligned} F_y[j] =& \langle \mathbb{L}[:,:,:,j,:,:,:,:], x^* y x (x^* y x y^* - x_0^* y_0 x_0 y_0^*)\rangle \\ &+ \langle \mathbb{L}[:,:,:,:,:,:,:,j], (x^* y x y^* - x_0^* y_0 x_0 y_0^*) x^* y x\rangle \\ =& 2\ I\ y[j] \left(|y[j]|^2 \|x\|_4^4 - |y_0[j]|^2 \|x_0 x\|_2^2\right) \\ &+ I\ y[j] \sum_{k=j-M+1}^{j} |x[j-k]|^2 \left(\|z^k\|_2^2 - \|z_0^k\|_2^2\right) \\ &+ I\ y[j] \sum_{k=j-M+1}^{j} |x[j-k]|^2 \|z^k\|_2^2 \\ &- I\ y_0[j] \sum_{k=j-M+1}^{j} x[j-k] x_0^*[j-k] (z^{kH} z_0^k). \end{aligned} \tag{46}$$

Determination of where these expressions are positive, negative, and zero relies on the relationship between x, y, x_0, y_0 as well as second-order cross terms stored in z^k, z_0^k.

References

[1] T. Bendory, D. Edidin, Toward a mathematical theory of the crystallographic phase retrieval problem, 2020, arXiv:2002.10081.

[2] E.J. Candès, X. Li, M. Soltanolkotabi, Phase retrieval from coded diffraction patterns, Appl. Comput. Harmon. Anal. 39 (2) (2015) 277–299, http://dx.doi.org/10.1016/j.acha.2014.09.004.

[3] G. Steinmeyer, A review of ultrafast optics and optoelectronics, J. Opt. A: Pure Appl. Opt. 5 (1) (2002) R1.

[4] A. Weiner, Ultrafast Optics, Vol. 72, John Wiley & Sons, 2011.

[5] T. Bendory, P. Sidorenko, Y.C. Eldar, On the uniqueness of FROG methods, IEEE Signal Process. Lett. 24 (5) (2017) 722–726, http://dx.doi.org/10.1109/LSP.2017.2690358.

[6] S. Nayer, N. Vaswani, Y.C. Eldar, Low rank phase retrieval, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 4446–4450, http://dx.doi.org/10.1109/ICASSP.2017.7952997.

[7] K. Lee, S. Bahmani, Y.C. Eldar, J. Romberg, Phase retrieval of low-rank matrices by anchored regression, Inf. Inference: J. IMA 10 (1) (2020) 285–332, http://dx.doi.org/10.1093/imaiai/iaaa018, arXiv:https://academic.oup.com/imaiai/article-pdf/10/1/285/36580350/iaaa018.pdf.

[8] D.J. Kane, R. Trebino, Characterization of arbitrary femtosecond pulses using frequency-resolved optical gating, IEEE J. Quantum Electron. 29 (2) (1993) 571–579.

[9] D. Kane, Real-time measurement of ultrashort laser pulses using principal component generalized projections, IEEE J. Sel. Top. Quantum Electron. 4 (2) (1998) 278–284, http://dx.doi.org/10.1109/2944.686733.

[10] D.J. Kane, Principal components generalized projections: A review, J. Opt. Soc. Amer. B 25 (6) (2008) A120–A132, http://dx.doi.org/10.1364/JOSAB.25.00A120, URL https://opg.optica.org/josab/abstract.cfm?URI=josab-25-6-A120.

[11] T. Bendory, D. Edidin, Y.C. Eldar, On signal reconstruction from FROG measurements, 2018, arXiv:1706.08494.

[12] S. Pinilla, T. Bendory, Y.C. Eldar, H. Arguello, Frequency-resolved optical gating recovery via smoothing gradient, IEEE Trans. Signal Process. 67 (23) (2019) 6121–6132, http://dx.doi.org/10.1109/tsp.2019.2951192.

[13] R. Koumans, A. Yariv, Time-resolved optical gating based on dispersive propagation: a new method to characterize optical pulses, IEEE J. Quantum Electron. 36 (2) (2000) 137–144, http://dx.doi.org/10.1109/3.823457.

[14] A.M.A. Motz, J.A. Squier, C.G. Durfee, D.E. Adams, Spectral phase and amplitude retrieval and compensation technique for measurement of pulses, Opt. Lett. 44 (8) (2019) 2085–2088, http://dx.doi.org/10.1364/OL.44.002085, URL http://www.osapublishing.org/ol/abstract.cfm?URI=ol-44-8-2085.

[15] A.M. Weiner, Ultrafast optical pulse shaping: A tutorial review, Opt. Commun. 284 (15) (2011) 3669–3692.

[16] A.M. Wilhelm, D.D. Schmidt, D.E. Adams, C.G. Durfee, Multi-mode root preserving ptychographic phase retrieval algorithm for dispersion scan, Opt. Express 29 (14) (2021) 22080–22095, http://dx.doi.org/10.1364/OE.426859, URL https://opg.optica.org/oe/abstract.cfm?URI=oe-29-14-22080.

[17] J. Fienup, Phase retrieval algorithms: A comparison, Appl. Opt. 21 (1982).

[18] R.W. Gerchberg, A practical algorithm for the determination of phase from image and diffraction plane pictures, Optik 35 (1972) 237–246, URL https://api.semanticscholar.org/CorpusID:55691159.

[19] J.-W. Liu, Z.-J. Cao, J. Liu, X.-L. Luo, W.-M. Li, N. Ito, L.-C. Guo, Phase retrieval via wirtinger flow algorithm and its variants, in: 2019 International Conference on Machine Learning and Cybernetics (ICMLC), 2019, pp. 1–9, http://dx.doi.org/10.1109/ICMLC48188.2019.8949170.

[20] T. Goldstein, C. Studer, PhaseMax: Convex phase retrieval via basis pursuit, IEEE Trans. Inform. Theory 64 (4) (2016).

[21] E.J. Candès, T. Strohmer, V. Voroninski, PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming, Comm. Pure Appl. Math. 66 (8) (2013) 1241–1274, http://dx.doi.org/10.1002/cpa.21432, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.21432.

[22] G. Wang, G.B. Giannakis, Y. Saad, J. Chen, Phase retrieval via reweighted amplitude flow, IEEE Trans. Signal Process. 66 (11) (2018) 2818–2833, http://dx.doi.org/10.1109/TSP.2018.2818077.

[23] G. Wang, G.B. Giannakis, Y.C. Eldar, Solving systems of random quadratic equations via truncated amplitude flow, IEEE Trans. Inform. Theory 64 (2) (2018) 773–794, http://dx.doi.org/10.1109/TIT.2017.2756858.

[24] O. Dhifallah, C. Thrampoulidis, Y.M. Lu, Phase retrieval via polytop optimization: Geometry, phase transitions, and new algorithms, 2018, arXiv.

[25] R. Chandra, Z. Zhong, J. Hontz, V. McCulloch, C. Studer, T. Goldstein, PhasePack: A phase retrieval library, in: Asilomar Conference on Signals, Systems, and Computers, 2017.

[26] E.J. Candes, X. Li, M. Soltanolkotabi, Phase retrieval via wirtinger flow: Theory and algorithms, IEEE Trans. Inform. Theory 61 (4) (2015) 1985–2007, http://dx.doi.org/10.1109/TIT.2015.2399924.

[27] S. Nayer, P. Narayanamurthy, N. Vaswani, Provable low rank phase retrieval, IEEE Trans. Inform. Theory 66 (9) (2020) 5875–5903, http://dx.doi.org/10.1109/TIT.2020.2984478.

[28] J. Sun, Q. Qu, J. Wright, A geometric analysis of phase retrieval, Found. Comput. Math. 18 (2017) 1131–1198, http://dx.doi.org/10.1007/s10208-017-9365-9.

[29] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, İ. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental algorithms for scientific computing in python, Nature Methods 17 (2020) 261–272, http://dx.doi.org/10.1038/s41592-019-0686-2.

[30] Y. Chen, Y. Chi, J. Fan, C. Ma, Spectral methods for data science: A statistical perspective, Found. Trends Mach. Learn. 14 (5) (2021) 566–806, http://dx.doi.org/10.1561/2200000079.

[31] A. Singer, Angular synchronization by eigenvectors and semidefinite programming, 2009, arXiv:0905.3174.

[32] P. Netrapalli, P. Jain, S. Sanghavi, Phase retrieval using alternating minimization, IEEE Trans. Signal Process. 63 (18) (2015) 4814–4826, http://dx.doi.org/10.1109/TSP.2015.2448516.

[33] H. Rauhut, R. Schneider, Z. Stojanac, Low rank tensor recovery via iterative hard thresholding, 2016, arXiv:1602.05217.

[34] S. Li, G. Tang, M.B. Wakin, Simultaneous blind deconvolution and phase retrieval with tensor iterative hard thresholding, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 2977–2981, http://dx.doi.org/10.1109/ICASSP.2019.8683575.

[35] J. Kossaifi, Y. Panagakis, A. Anandkumar, M. Pantic, Tensorly: Tensor learning in python, J. Mach. Learn. Res. 20 (26) (2019) 1–6, URL http://jmlr.org/papers/v20/18-277.html.

[36] D.R. Luke, Relaxed averaged alternating reflections for diffraction imaging, Inverse Problems 21 (1) (2005).

[37] X. Li, S. Wang, Y. Cai, Tutorial: Complexity analysis of singular value decomposition and its variants, 2019, arXiv: Numerical Analysis, URL https://api.semanticscholar.org/CorpusID:203951880.

[38] C. Xiao, C. Yang, M. Li, Efficient alternating least squares algorithms for low multilinear rank approximation of tensors, J. Sci. Comput. 87 (3) (2021) 67, http://dx.doi.org/10.1007/s10915-021-01493-0.

[39] Y. Chen, Y. Chi, J. Fan, C. Ma, Gradient descent with random initialization: fast global convergence for nonconvex phase retrieval, Math. Program. 176 (1) (2019) 5–37, http://dx.doi.org/10.1007/s10107-019-01363-6.

[40] Squier Group, 2019, URL https://ultrafastoptics.mines.edu.

[41] K. Kreutz-Delgado, The complex gradient operator and the CR-calculus, 2009, http://dx.doi.org/10.48550/ARXIV.0906.4835, URL https://arxiv.org/abs/0906.4835.