Signal Processing 220 (2024) 109464

Contents lists available at ScienceDirect

Signal Processing

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

Phase retrieval from integrated intensity of auto-convolution

Dan Rosen [a,*], Daniel Scarbrough [b], Jeff Squier [b], Michael B. Wakin [a]

[a] Colorado School of Mines, Department of Electrical Engineering, 1500 Illinois St., Golden, CO, 80401, USA
[b] Colorado School of Mines, Department of Physics, 1500 Illinois St., Golden, CO, 80401, USA

ARTICLE INFO

Keywords:
Phase retrieval
Pulse characterization
Microscopy
Ultra-fast optics
Wirtinger descent
Iterative hard tensor thresholding

ABSTRACT

Ultra-fast optical pulses are the most ephemeral sensing paradigm ever devised, examining events over incredibly brief timescales with broadband illumination. A consequence of sensing at timescales lower than a picosecond is that pulse characterization cannot be done with traditional analog-to-digital samplers and must be ascertained from integrating intensity sensors. Techniques for pulse characterization have been constructed using combinations of time-invariant and time-variant filter responses to create non-linear but invertible intensity datasets (Walmsley & Dorrer, 2009). In this paper, we develop a novel high-order phase retrieval technique to perform pulse characterization from a single-pixel integrating sensor measuring integrated intensity of auto-convolution (IIAC). We examine gradient descent's ability to recover signals as a function of signal dimension and measurement count, and we demonstrate the effective use of iterative hard tensor thresholding as an initializer. Finally, we demonstrate IIAC recovery in a laboratory setting to recover the time profile of a complex laser pulse. We assert that the IIAC recovery solution demonstrated here simultaneously provides the optics community with a pulse characterization technique that scales to low-power microscopy systems and provides the optimization community with a physically motivated high-order phase retrieval problem enhanced by low-rank tensor processing.

1. Introduction

Phase retrieval theory has long been motivated by problems in optics because the vast majority of measurements that are available in optics are intensity based. Traditionally, the techniques of phase retrieval have been applied in crystallography [1], coded-diffraction imaging [2], and ptychographic imaging [3], where dense far-field images measure the intensity of a scaled objective plane Fourier transform. Phase retrieval of this form constrains recovery against the Fourier kernel and typically employs known spatial modulation masks as a set of linear constraint equations to recover complex objective plane images [4].

Ultra-fast optics is the study of laser pulses that have been concentrated in time to durations between 10^{-12} and 10^{-18} seconds [5], offering a mechanism to explore events at small time-scales. The narrow time duration of these pulses also offers a broad frequency spectrum with which to study material properties and composition. Attempting to recover the pulse's complex spectrum, called pulse characterization, requires techniques that are more complicated than traditional phase retrieval. Extremely high bandwidth and unique behavior of light at optical frequencies prevent traditional in-phase/quadrature (IQ) sampling and motivate phase retrieval from measurements constructed from time-variant and time-invariant filter sets [6]. A popular technique for creating time-variant filters at time scales matching the pulse is to use non-linear optical elements: crystals that produce output pulses whose phase and amplitude are non-linear functions of input pulses. These non-linear effects can operate on a single pulse or multiple pulses and a commonly employed subset of effects, called second harmonic generating (SHG), are frequently represented as intensity of auto- and cross- convolution or correlation [7]. In the optimization community, FROG pulse characterization has recently been studied as a higher-order phase retrieval problem [8]. This problem was generalized to recover signals from intensity of cross-correlation and formulated as a low-rank phase retrieval [9,10] problem which could reliably be solved with Wirtinger gradient techniques and enhanced with low-rank structured tensor recovery [11].

We view in-series spectrographic (frequency-resolved) pulse characterization in two categories: those modulated in the time domain, and those modulated in the frequency (wavelength) domain. Time-modulation techniques use a probing pulse, either a copy of the pulse-under-test or a time-locked secondary pulse, to element-wise modulate the pulse-under-test in a non-linear medium before being measured by a spectrometer. FROG is the most widely used paradigm of this type [12]. Frequency-modulation techniques transform the pulse into

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

https://doi.org/10.1016/j.sigpro.2024.109464
Received 21 December 2023; Received in revised form 16 February 2024; Accepted 5 March 2024
Available online 11 March 2024

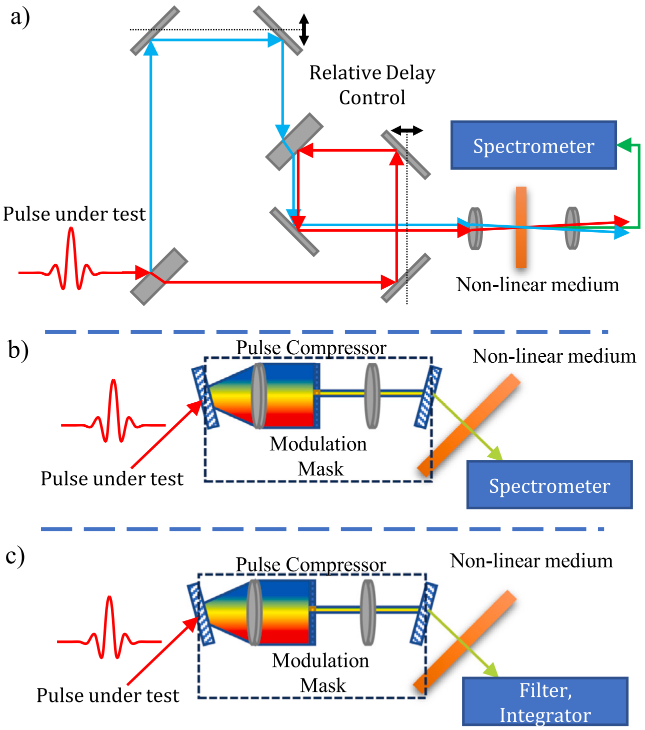

Fig. 1. Three pulse characterization setups. (a) A time-modulated setup that uses physical delay manipulation of pulses to generate a two-dimensional dataset with a spectrometer. (b) A frequency-modulated setup that modulates pulses in the wavelength domain before SHG in a transform domain, generating a two-dimensional dataset over modulation from the spectrometer. (c) The proposed physical setup which integrates the spectral measurement into a single one-dimensional measurement that varies with modulation in the pulse compressor.

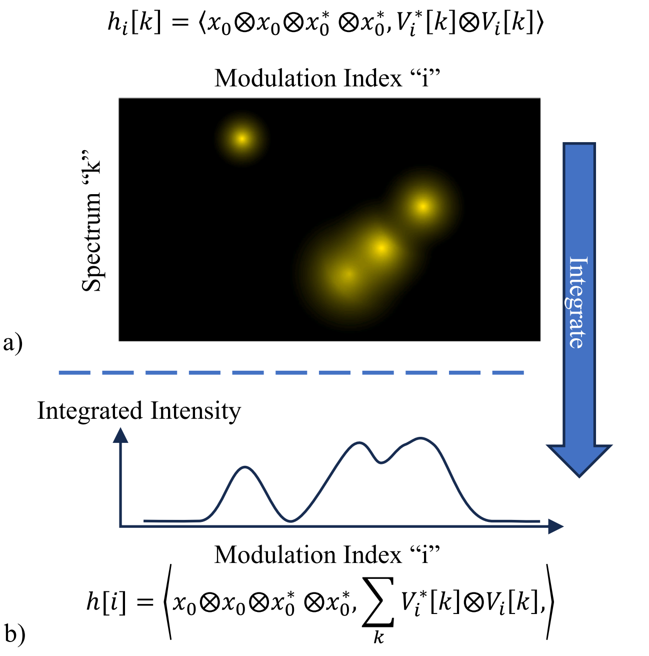
$h_i[k] = \langle x_0 \otimes x_0 \otimes x_0^* \otimes x_0^*, V_i^*[k] \otimes V_i[k] \rangle$
Modulation Index "i"
Spectrum "k"
Integrate
a)
Integrated Intensity
Modulation Index "i"
$h[i] = \left\langle x_0 \otimes x_0 \otimes x_0^* \otimes x_0^*, \sum_k V_i^*[k] \otimes V_i[k], \right\rangle$
b)

Fig. 2. The two dimensional intensity datasets typical for frequency-modulated pulse characterization (a) are indexed in one direction by modulation index i and another by spectrometer pixels around a center k. Our technique is designed to recover the same unknown signal sought after by time-resolved pulse characterization using a spectrally integrated form of these measurements shown in (b).

a spatial wavelength domain in a pulse compressor where the pulse is modulated with a mask. The pulse continues out of the pulse compressor and is element-wise-squared in a non-linear medium before being measured in a spectrometer. Techniques that deploy this include SPARC and DSCAN [13,14]. When the non-linear medium used is a second-harmonic generating (SHG) medium, time-modulated systems produce intensity of cross-correlation measurements and frequency-modulated systems produce intensity of auto-convolution measurements. Time-modulated and frequency-modulated pulse-characterization physical setups are shown in Fig. 1.

Time-modulated and frequency-modulated measurements both produce a two-dimensional dataset, one dimension representing spectrum and the other representing modulation index. For the purposes of this paper, we will interpret a two-dimensional dataset to mean that a measurement using one known modulation produces a series of measurements at the spectrometer, rather than a single measurement. Time-modulation systems take a series of spectra while physically changing pulse overlap. In this case, intensity of cross-correlation is stored in the axis corresponding to pulse shift, while the spectral axis varies phasor modulation.

Frequency-modulated systems modulate a pulse in the wavelength domain before doubling (element-wise square) in the time domain. This action produces an SHG spectrum representing the intensity of auto-convolution of the modulated pulse spectrum. New modulations produce additional auto-convolutions of modulated spectra. The two axis dataset is spectrum on one axis and mask index on the second.

For both time- and frequency- modulated systems, a dataset made from a fixed index in the cross-correlation or auto-convolution dimension corresponds to a unique phase retrieval problem whose ground-truth generating vector is a diagonal of the outer product between the pulse and the probe pulse or the pulse and itself, respectively. Collecting the results of phase retrieval over all lags of correlation or convolution can recover the diagonals of the outer product of interest. Phase retrieval, however, produces estimates of ground-truth that are globally phase-ambiguous, producing phase misalignment between the diagonals of the target outer product matrix. This prevents eigendecomposition from returning the signal of interest and requires a broader optimization to combine diagonal information into a low-rank outer product. The problem of recovering a signal from the intensity of auto-convolution (IAC) or more generally, the intensity of cross-correlation (ICC), was studied using Wirtinger gradient descent and iterative hard tensor thresholding [11] and we expand on the this framework.

While the ICC problem creates a fascinating connection between low-rank phase retrieval and time- and frequency- modulated pulse characterization systems, study of the tensor structure of ICC reveals a closely related problem whose solution may produce a new and powerful tool for optical pulse characterization. We demonstrate that pulse characterization can be done on frequency-modulated datasets that have been integrated along the spectral axis as shown in Fig. 2.

Physically, this corresponds to a frequency-modulated system where the spectrometer is replaced with a single square-law integrator like a photo-diode or photon-counter, shown in Fig. 1(c). We refer to this dataset as integrated intensity of auto-convolution (IIAC), and we briefly discuss a broader definition of the measurement that had not yet been matched to a physical analog (integrated intensity of cross-correlation, IICC) in Appendix C. Our efforts have been concentrated on frequency-modulated systems for two reasons. Firstly, known masks are used in the wavelength domain to modulate the pulse, as opposed to a copy of a potentially unknown pulse in the time domain. Secondly, integration onto a single sensor instead of a spectrometer only provides the integrated intensity of auto-correlation for frequency-modulated systems, and would provide the integrated intensity of delayed element-wise multiplication for time-modulated systems.

In current pulse characterization systems, it is not uncommon to employ spectrometers, cameras, fast-time samplers, or other devices that create large redundant datasets leveraging relatively low amounts of modulation *a priori*. Our proposed technique creates a single unique measurement from every modulation and increases the information in that single measurement relative to ICC. Rather than the typical use of dense or fast sensors, this approach allows the use of a slow single channel sensor.

Mathematically, the IIAC recovery problem presents a new structured high-order phase retrieval problem distinct from both traditional and low-rank phase retrieval. We demonstrate that the problem is effectively solved with Wirtinger gradient techniques, yet substantially improved with an iterative hard thresholding initializer exploiting IIAC's linear fourth-order tensor structure. Tensors are an effective tool to create "lifted" linear problems out of phaseless polynomial measurements that can leverage *a priori* known low-rank of the solution tensor. This development draws parallels to the development of PhaseLift [15] in traditional phase retrieval where a phaseless linear problem is lifted to a low-rank matrix retrieval problem.

We summarize the contributions made in this paper as follows: a new pulse characterization technique constructed from variation of frequency-modulated total SHG pulse energy, a gradient technique to invert these measurements as well as an empirical estimate of required sampling for probable signal recovery, a demonstration of an effective initializer technique based on iterative hard tensor thresholding, and an experimentally demonstrated inversion from laboratory data. To our knowledge, we are the first to group to perform pulse characterization with an integrated fourth order total energy measurement and we present it with connection to structured inverse problems and Wirtinger gradient optimization.

In Section 2 we present the IIAC recovery problem definition and describe the structure in vector, matrix, and low-rank tensor forms. In Section 3 we examine a least-squares cost-function and examine its Wirtinger gradient and Hessian. We also form the IIAC problem as a low-rank tensor recovery and develop an iterative hard tensor thresholding recovery process. In Section 4 we numerically model the convergence rate of randomly initialized Wirtinger descents for generated IIAC datasets. We then compare this random initialization to descents that were initialized from the iterative hard tensor thresholding initializer developed in Section 3. We finish Section 4 with a brief overview of problem complexity and a brief study on the effects of i additive Gaussian noise. Finally, we demonstrate the IIAC recovery technique in a laboratory setting in Section 5 to recover the time delay between two pulses.

2. Problem definition and variations

In this section, we introduce notation and formulate the IIAC problem in several variations. These variations serve different functions, exposing single variables for ease of gradient derivation or formulating the problem as a linear inner product between tensors to demonstrate the rationale behind tensor iterative hard thresholding.

2.1. Notation

These are some common notations that will be used throughout the paper and is summarized below.

- $\circ^T$ represents the transpose of a vector or a matrix.
- $\circ^*$ represents the element-wise conjugate of a vector, matrix, or tensor.
- $\circ^H$ represents the conjugate transpose of a vector or a matrix.
- $a \odot b$ represents the element-wise product of a, b.
- $\langle a, b\rangle$ represents the complex inner product between a, b which can be vectors, matrices, or tensors. This complex inner product is defined as the sum of the element-wise product $a \odot b^*$. For vectors this equates to $b^H a$, for matrices $\mathrm{trace}(AB^H)$.
- aa^H represents the outer product of a column vector a with the row vector a^H.
- $a \otimes a^*$ is equivalent to aa^H, as $\otimes$ represents the outer product between two objects and can be used to represent tensors like $a \otimes a \otimes a^* \otimes a^*$, a fourth order tensor made by the fourth order outer product of a single vector.
- $x_0 \in \mathbb{C}^M$ is the ground-truth objective vector representing the sampled frequency profile $E[\omega]$ of a laser pulse.
- $x \in \mathbb{C}^M$ is current guess or best approximation of x_0.
- $c_i \in \mathbb{C}^M$ is a known modulation variable indexed by $i = [0, 1, \dots, I-1]$. In time-modulated systems, $c_i[n]$ is typically a complex exponential over n with advancing rate over i. In frequency-modulated systems, c_i is whatever modulation can be placed in the pulse compressor: typically an attenuating modulation, but possibly a phase-rotation produced by a spatial light modulator, for example. Pulse compressors can apply an attenuation mask at the modulation plane and then add phase modulation by physically moving the second diffraction grating as done in [14].
- $V_i[k] = \mathrm{antidiag}(c_i c_i^T,\ k)$ represents an $M \times M$ matrix whose values are all zero except for the kth anti-diagonal (where $k = 0$ is the main anti-diagonal, $k = -1$ is toward the top-left of the matrix), which is filled with the lagged product of $c_i[n+k]c_i[M-1-n]$. We will refer to this matrix as the modulation matrix, indexed by modulation index i and lag index k. This matrix linearly encodes the action of modulated convolution when used as a quadratic operator acting on $x_0 x_0^T$, shown in Section 2.3.
- $\mathcal{V}_i = \sum_{k=-M+1}^{M-1} \left(V_i^*[k] \otimes V_i[k]\right)$ is the fourth-order tensor whose entries are zero except where filled as $\mathcal{V}_i[q,r,s,t] = \sum_{k=-M+1}^{M-1} \left(\mathrm{antidiag}(c_i c_i^T,\ k)^*[q,r]\ \mathrm{antidiag}(c_i c_i^T,\ k)[s,t]\right)$. We will refer to this operator as the modulation tensor, indexed by modulation index i and lag index k. This tensor encodes the action of accumulating the magnitude squared result of c_i modulated convolution as a linear operator acting the fourth-order outer product tensor $x_0 \otimes x_0 \otimes x_0^* \otimes x_0^*$. This is shown in Section 2.4.
- $\mathcal{X}_0 = x_0 \otimes x_0 \otimes x_0^* \otimes x_0^*$ is the fourth-order outer product tensor with non-zero entries indexed as $\mathcal{X}_0[q,r,s,t] = x_0[q]x_0[r]x_0^*[s]x_0^*[t]$. $\mathcal{X}$ has an identical construction but with x instead of x_0.
- $\iff$ denotes the "if and only if" symbol to imply bidirectional equivalent statements.

2.2. IIAC quartic vector definition

To formulate the IIAC function, we first examine ICC adapted specifically for frequency-modulated systems measuring intensity of auto-convolution (IAC) and then define IIAC as the sum of IAC over the wavelength (or k axis). In adapting the generalized ICC [11] to IAC we adapt a forward function ingesting two vectors (cross-correlation) into a forward function of a single variable (auto-convolution). The adapted IAC definition is shown in Eq. (1).

$$
\begin{aligned}
h_i[k] = &\left|\sum_{n=\max(0,-k)}^{M-1-\max(0,k)} c_i[n+k]c_i[M-1-n]x_0[n+k]x_0[M-1-n]\right|^2 \\
&+ \eta_i[k], \\
&= \left|x_0^T\,\mathrm{antidiag}(c_i c_i^T,\ k)x_0\right| + \eta_i[k] \\
&= \left|x_0^T V_i[k] x_0\right|^2 + \eta_i[k] \\
&= x_0^T V_i[k] x_0\ x_0^H V_i^*[k] x_0^* + \eta_i[k] \\
&i \in [0, 1, \dots, I-1]
\end{aligned}
\tag{1}
$$

Note that the equivalence of the first and second lines in Eq. (1) stems from the multiple operations encoded in the structure of the matrix $\mathrm{antidiag}(c_i c_i^T,\ k)$. The anti-diagonal structure of the matrix encodes the time reversal and lag shifting required between x_0 and itself for convolution, and the matrix also carries the element-wise modulations imparted with c_i and a lag multiplied copy of c_i. Now, a sum over the k or wavelength axis yields the expression for IIAC measurements $h[i]$ in Eq. (2). This form of the IIAC problem will be referred to as the vector form.

$$\begin{aligned} h[i] &= \sum_{k=-M+1}^{M-1} h_i[k] \\ &= \sum_{k=-M+1}^{M-1} \left(x_0^T V_i[k] x_0 \; x_0^H V_i^*[k] x_0^* + \eta_i[k] \right), \\ &= \sum_{k=-M+1}^{M-1} \left(x_0^T V_i[k] x_0 \; x_0^H V_i^*[k] x_0^* \right) + \eta[i], \\ & i \in \{0, 1, \dots, I-1\} \end{aligned} \tag{2}$$

We note that the relationship of $\eta[i]$ to $\eta_i[k]$ is a sum over k, i.e. $\eta[i] = \sum_k \eta_i[k]$. This vector form is particularly suited for computing the Wirtinger gradient, which is done in Appendix A.

2.3. IIAC quadratic low-rank matrix definition

The definition of IIAC is very nearly a low-rank phase retrieval problem [10], however the sum over wavelength or k prevents the problem from adhering to the low-rank phase retrieval structure. This is shown in Eq. (3) where IIAC is formed as a sum of intensities of inner products between a low-rank matrix $x_0 x_0^T$ and a sensing matrix indexed by i, k.

$$\begin{aligned} h[i] &= \sum_{k=-M+1}^{M-1} \left| x_0^T V_i[k] x_0 \right|^2 + \eta[i] \\ &= \sum_{k=-M+1}^{M-1} \left| \langle x_0 x_0^T, \; V_i^*[k] \rangle \right|^2 + \eta[i] \end{aligned} \tag{3}$$

Were it not for the sum over k in Eq. (3), the problem would fit the description of low-rank phase retrieval as the ICC and IAC problems did. While the definition shown in Eq. (3) is a sum over wavelength, it is possible that the sum could be generalized to a weighted sum or even a mixing matrix to account for dispersive integrators. This approach is not considered here for simplicity.

2.4. IIAC linear tensor definition

Although IIAC cannot be viewed as a low-rank phase retrieval problem, it can be viewed as a low-rank tensor recovery problem. This owes to the fact that IIAC measurements can be written as linear inner products between an objective tensor and a series of measurement tensors as shown in Eq. (4).

$$\begin{aligned} h[i] &= \sum_{k=-M+1}^{M-1} \left| x_0^T V_i[k] x_0 \right|^2 + \eta[i] \\ &= \sum_{k=-M+1}^{M-1} \langle x_0 x_0^T, \; V_i^*[k] \rangle \langle x_0^* x_0^H, \; V_i[k] \rangle + \eta[i] \\ &= \sum_{k=-M+1}^{M-1} \langle x_0 \otimes x_0 \otimes x_0^* \otimes x_0^*, \; V_i^*[k] \otimes V_i[k] \rangle + \eta[i] \\ &= \langle x_0 \otimes x_0 \otimes x_0^* \otimes x_0^*, \; \sum_{k=-M+1}^{M-1} V_i^*[k] \otimes V_i[k] \rangle + \eta[i] \\ &= \langle \mathcal{X}_0, \; \mathcal{V}_i \rangle + \eta[i] \end{aligned} \tag{4}$$

This tensor formulation shows that IIAC is a structured linear problem, where measurements are taken against an $M \times M \times M \times M$ tensor $\mathcal{X}_0$ that is known to be rank-one with outer product symmetry $a \otimes a \otimes a^* \otimes a^*$. This structure will be directly exploited later as a condition in an iterative hard tensor thresholding algorithm.

2.5. IIAC ambiguities

At best, IIAC measurements can be inverted to an estimate x of x_0 with unknown, but trivial ambiguities. Specifically, x may be off by an unknown constant phase and frequency (linear phase) offset. Physically, absolute pulse phase is typically not useful and lost to all intensity-based measurements. Frequency offset when estimating the frequency domain of a pulse represents group delay, where unknown group delay is a consequence of using an integrating sensor with non-linear optics to create time-variation that cannot be externally synchronized like time gating. Mathematically, frequency and phase offsets are unknown in IIAC because of invariance of the IIAC measurement to these ambiguities, i.e. $|x^T V_i[k] x|^2 = |x_a^T V_i[k] x_a|^2$ where $x_a[n] = \phi x[n] \exp(jn\psi)$ is an ambiguous estimate of x as demonstrated in Eq. (5). Eq. (5) demonstrates the invariance of IIAC measurements to arbitrary phase and frequency offsets ϕ, ψ and sets $c_i[n] = 1$ without loss of generality. Note that the equivalence from the third to fourth line in Eq. (5) stems from the constant amplitude of a complex phase offset ϕ and lack of interaction between the rate ψ complex exponential and the sum variable n.

$$\begin{aligned} & |x_a^T V_i[k] x_a|^2 \\ &= \left| \sum_{n=\max(0,-k)}^{M-1-\max(0,k)} x_a[n+k] x_a[M-1-n] \right|^2 \\ &= \left| \sum_{n=\max(0,-k)}^{M-1-\max(0,k)} x[n+k] x[M-1-n] \phi^2 \exp\left(j\psi(k+M-1)\right) \right|^2 \\ &= \left| \phi^2 \exp\left(j\psi(k+M-1)\right) \sum_{n=\max(0,-k)}^{M-1-\max(0,k)} x[n+k] x[M-1-n] \right|^2 \\ &= \left| \sum_{n=\max(0,-k)}^{M-1-\max(0,k)} x[n+k] x[M-1-n] \right|^2 \\ &= |x^T V_i[k] x|^2 \end{aligned} \tag{5}$$

3. Approach

Approaches studied in this paper to recover x_0 from IIAC measurements fall into two categories: Wirtinger gradient descent algorithms, and iterative hard tensor based algorithms.

3.1. Wirtinger gradient and Hessian

A least-squares intensity cost-function is shown in Eq. (6). Gradient descent along the Wirtinger derivative of this function forms this paper's primary tool for recovering x from IIAC measurements. The Wirtinger gradient and Hessian definition and derivation are shown in Appendix A [16].

$$\begin{aligned} f(x) &= \frac{1}{2} \sum_{i=0}^{I-1} \left(h[i] - \sum_{k=-M+1}^{M-1} x^T V_i[k] x \; x^H V_i^*[k] x^* \right)^2 \\ &= \frac{1}{2} \sum_{i=0}^{I-1} e_i^2 \end{aligned} \tag{6}$$

Here, $e_i = h[i] - \sum_{k=-M+1}^{M-1} x^T V_i[k] x \; x^H V_i^*[k] x^*$ will be used represent the error between measurement $h_i[k]$ and the synthesized measurement generated from the current approximation x.

The Wirtinger gradient of the cost-function in Eq. (6) is shown in Eq. (7).

$$\nabla f(x) = -2 \sum_i e_i \sum_{k=-M+1}^{M-1} \begin{pmatrix} (x^T V_i[k] x) \; V_i^*[k] x^* \\ (x^T V_i[k] x)^* \; V_i[k] x \end{pmatrix} \tag{7}$$

We can see from the construction that when error is zero ($e_i = 0, \forall i$), the gradient is zero. The Wirtinger Hessian is shown in Eq. (8).

$$\nabla^2 f(x) = \sum_i \left(a_i a_i^H - e_i \sum_{k=-M+1}^{M-1} A_{i,k} \right) \tag{8}$$

$$a_i = -2 \sum_{k=-M+1}^{M-1} \begin{pmatrix} (x^T V_i[k]x)\ V_i^*[k]x^* \\ (x^H V_i^*[k]x^*)\ V_i[k]x \end{pmatrix}$$

$$A_{i,k} = \begin{pmatrix} 4V_i^*[k](x^T V_i[k]x^*) & 2V_i^*[k](x^T V_i[k]x) \\ 2V_i[k](x^H V_i^*[k]x^*) & 4V_i[k](x^H V_i^*[k]x) \end{pmatrix}$$

From this expression we see that when error is zero, the Hessian is the sum of positive coefficient outer products, meaning that the matrix is positive semi-definite.

3.2. Tensor IHT

As shown in Eq. (4), the IIAC measurement for every unique modulation index i is the result of a linear inner product between two structured fourth-order tensors. The objective tensor $\mathcal{X}_0$ is known *a priori* to be rank-one positive-semidefinite and therefore has structure that can be exposed using a high-order singular value decomposition (HoSVD). Iterative hard tensor thresholding (IHT) is an iterative two step process [17–19]. The first step backprojects error between a measurement set and measurements synthesized by the current approximation of ground-truth. The second step enforces a low rank constraint on the current ground-truth tensor estimate. In IIAC, error backprojection will be combined with the enforcement of rank-one tensor structure using HoSVD.

In this paper, we deploy two forms of this algorithm, IHT constructed with simple alternating minimization (AltMinIIAC), and IHT constructed with relaxed averaged alternating reflections (RAARIIAC) [20]. RAAR has been shown to converge faster for alternating projection style algorithms and was chosen to improve convergence due to the high processing time involved in IHT.

AltMinIIAC is outlined in Algorithm 1. Several supporting algorithms are outlined in Appendix B. Algorithm 1 takes as input IIAC measurements h_i and the known modulation vectors c_i. A rigid index set (ii, jj, kk, ll) is generated that organizes modulations into a linear forward operator along with its pseudo inverse (A, A^+). A randomly initialized rank-one tensor χ serves as our initial tensor estimate. If the L_2 error between the forward projected approximation and the IIAC measurements falls below a user determined error or if the maximum number of iterations has been reached, iteration terminates and the lead eigenvector x is returned as an approximation to x_0.

RAARIIAC is nearly identical to the procedure that is outlined in Algorithm 1 with the exception that the main iteration loop constructs reflections and projections into each constraint set. This updated main iteration loop update is compared with AltMinIIAC in Algorithm 2.

Algorithm 1: IIAC Alternating Minimization (AltMinIIAC)

```
Data:
    $c \in \mathbb{C}^{I \times M}$
    $h[i], i = [0, 1, ..., I-1]$, vectorized IIAC measurements
Parameters:
    thresh $= 1e-3$
    maxIts $= 4000$
Result: $x \in \mathbb{C}^M$
$ii, jj, kk, ll \leftarrow IIACIndex(M,\ N)$
$A \leftarrow \mathbf{0} \in \mathbb{C}^{I \times Q}, Q = \sum_{k=-M+1}^{M-1} (M - |k|)^2$
for( $i = [0, 1, ..., I-1]$ ) {
    $A[i, :] \leftarrow c[i, ii] * c[i, jj] * c^*[i, kk] * c^*[i, ll]$
}
$A^+ \leftarrow pinv(A)$
$\chi \leftarrow IIACStart(M, h[i])$
/* Main iteration loop */
for( $ii = 0, 1, ..., maxIts - 1$ ) {
    $\chi_{back}, error \leftarrow Back-project(\chi, A, A^+, ii, jj, kk, ll, h[i])$
    $\chi \leftarrow IIACRank(\chi_{back})$
    if $error > thresh$ then
        break
}
$x, x, x^*, x^* =$ tucker($\chi$, rank $= [1, 1, 1, 1]$)
$x \leftarrow ScaleIIAC(x, c, h)$
```

Algorithm 2: AltMinIIAC vs. RAARIIAC update step comparison

```
Data:
    $\chi_i$, current iteration tensor
Parameters:
    $\alpha, \beta = .9, .5$
Result: $\chi_{i+1}$, next tensor iteration
// Alt Min Iteration:
$\chi_{back}, error \leftarrow Back-project(\chi_i, A, A^+, ii, jj, kk, ll, h[i])$
$\chi_{i+1} \leftarrow IIACRank(\chi_{back})$
// RAAR Iteration:
$P_2, error \leftarrow Back-project(\chi, A, A^+, ii, jj, kk, ll, h[i])$
$R_1 \leftarrow 2P_2 - \chi_i$
$P_1 \leftarrow IIACRank(R_1)$
$R_2 \leftarrow 2P_1 - P_2$
$\chi_{i+1} \leftarrow \beta(\alpha R_2 + (1-\alpha)\chi_i) + (1-\beta)P_2$
```

4. Performance

Wirtinger descent algorithms are the most effective and scalable tool for returning x from IIAC measurements we have tested so far. This section examines randomly initialized and IHT initialized Wirtinger descents and examines their error statistics as a function of $M,\ I$. Both c_i and x_0 are generated from a complex Gaussian distribution, shown in Algorithm 11.

Wirtinger gradient descent algorithms converge much faster than IHT methods primarily because of decreased complexity. The per iteration memory and computational complexity of both algorithms are discussed in Section 4.4. For ICC [11], it was determined that though gradient methods were more efficient and faster to compute than tensor methods, tensor methods were able to improve the performance of gradient methods acting as an initializer in low measurement count scenarios. To determine the efficacy of tensor methods as an initializer in IIAC we first determine an approximation of the number of measurements I required to return a correct solution for randomly initialized gradient descent as a baseline oversampling requirement. We then run identical experiments comparing randomly initialized versus IHT initialized Wirtinger descents to determine if either have an advantage.

The primary metric used to compare algorithms is the normalized error, specifically the L_2 norm of the error divided by the L_2 norm of the ground truth, as shown in Eq. (10). Before a difference can be taken however, global phase and frequency ambiguity needs to be removed. This is achieved by minimizing the difference between our estimate x and the ground truth x_0 over all possible frequency and phase offsets (Eq. (9)) before taking their difference. In practice, offset correction is performed by dividing our estimate by the ground truth, estimating frequency offset as the median phase step over samples, then estimating phase offset as the angle of the frequency corrected inner product. This is detailed in Algorithm 10.

$$\theta, \phi = \underset{\theta', \phi'}{\operatorname{argmin}} \sum_n \left(\exp(j(\phi' n + \theta'))x[n] - x_0[n] \right)^2 \tag{9}$$

$$x_a[n] = \exp(j(\phi n + \theta))x[n]$$

$$error = \frac{\|x_0 - x_a\|_2}{\|x_0\|_2} \tag{10}$$

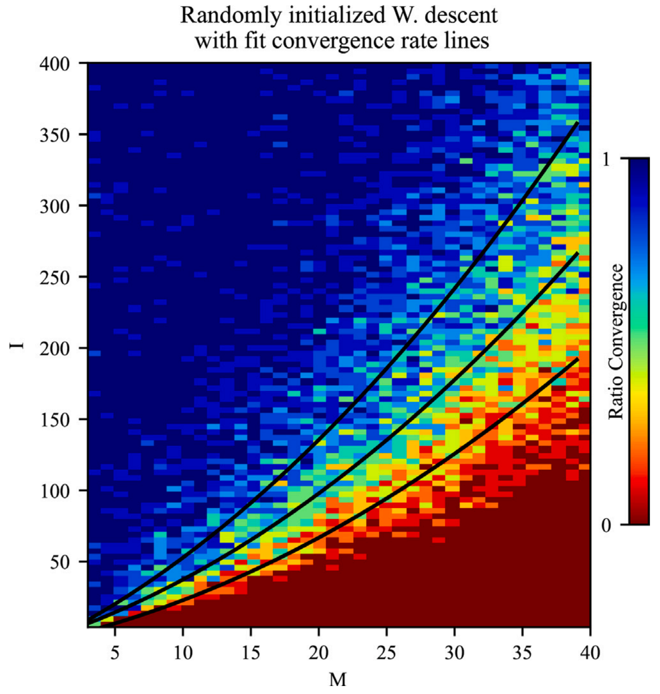

Fig. 3. 10 trials of IIAC Wirtinger descent over measurement count I and signal dimension M. Pixels are shaded by percent of trials that converged in relative error to below .01. The black lines are three fit quadratic functions representing the 25%, 50%, and 75% convergence rates.

With this error metric, we can define another metric employed in this paper: rate of convergence. For a given problem dimension M and measurement count I, we conduct N trials of a randomly initialized IIAC problem set. If a single trial recovers a guess of x_0 that has a normalized error below a set threshold, the trial is declared a success. We define the convergence rate as the fraction of successful trials.

4.1. Randomly initialized sampling requirements

To determine sampling requirements (I) for recovering an arbitrary signal of dimension M, we perform a gradient descent convergence rate test over a grid of M, I values. Each test begins with a randomly initialized x_0 and modulation set c_i and an estimate of x_0 is produced from a randomly initialized Wirtinger descent using only c_i and the IIAC dataset h_i generated from x_0 and c_i. The results of this experiment are shown in Fig. 3. This experiment operates over 10 trials per pixel (I, M pair), with Fig. 3 showing the number of trials per pixel that converged in normalized error to less than .01. This is summarized in Algorithm 3, and uses Scipy's [21] minimize function with "L-BFGS-B" solver, $ftol = 1e{-}6$, and a maximum of 1000 iterations.

4.2. Convergence rate curve-fitting

Fig. 3 shows a quadratic curve that serves as the 25%, 50%, and 75% convergence rate lines for I as a function of M for randomly initialized inversion of IIAC measurements using Wirtinger gradient descent. To determine this curve, and to determine the function that best suits the growth of measurements required to recover a signal reliably, we established the following procedure:

1. For a single M over a range of I, we fit a generalized logistic function: $f(I|M) = (1 + exp(-a * (I - b)))^{-1/c}$. This function is monotonic with three parameters to estimate the gradual transition from low rate of convergence to high rate of convergence. Note the rate a, center b, and inflection c are functions of M.
2. To determine the number of samples required for a desired convergence rate p we solve $I_M = f^{-1}(p|M)$.
3. For a given p, we solve for I_M over all grid M values and fit various functions $f_p(M)$ to the resulting M, I_M set.

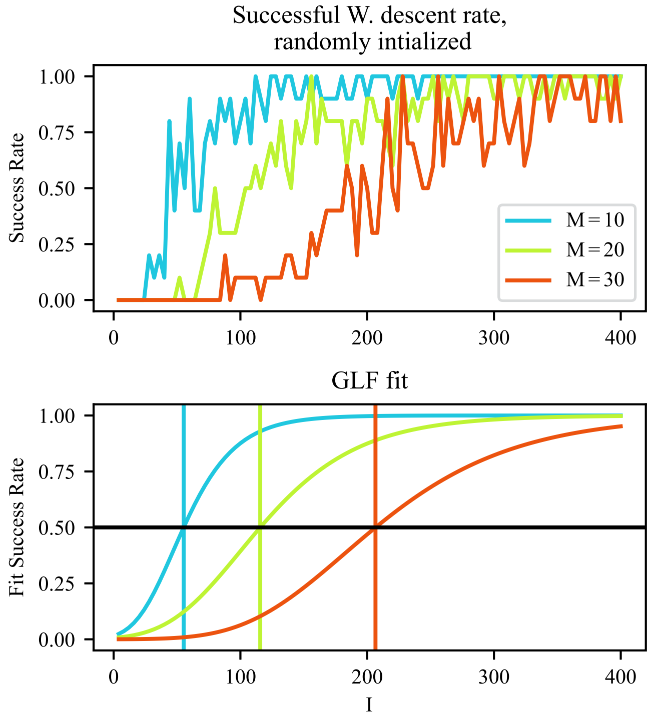

Fig. 4. Convergence rates for various M are fit to GLF curves $p = f(I|M)$. I_M estimates for a desired rate p are determined by solving $I_M = f^{-1}(p|M)$. This has been done in the lower plot with the fit GLF curves for the value $p = .5$.

Table 1
MLE comparison between measured values of I_M and fit values for functions at $p = 25\%, 50\%, 75\%$.

Function	$p = 25\%$	50%	75%
Polynomial (Eq. (11))	17.91	16.23	53.13
Exponential (Eq. (12))	18.85	16.88	53.2

The process of fitting convergence rate data to GLF curves and determining $f^{-1}(.5|M)$ is shown for $M = [10, 20, 30]$ in Fig. 4.

Two candidate functions were considered to determine growth requirements of I as a function of M: a quadratic function to determine polynomial fit in Eq. (11) and an exponential in Eq. (12). Variables a, b, c, d are free variables to adjust fit.

$$f_p(M) = a * M^2 + b * M + c \qquad (11)$$

$$f_p(M) = c * exp(a * M + b) + d \qquad (12)$$

Determining fit quality was judged with mean least-squares error (MLE) between measured I_M values and fit I_M values for $p = [25\%, 50\%, 75\%]$. MLE for Eqs. (11) and (12) are shown in Table 1.

Both functions fit the dataset well; however, the polynomial fit had a small advantage. The values of a, b, c per convergence rate fit are given in Table 2. From Table 2, we see that the quadratic factor grows fairly consistently for increasing convergence rates.

4.2.1. Comparison with state-of-the-art

Pulse characterization from one-dimensional IIAC measurements (indexed only by modulation index i) has no direct comparison with techniques in the literature that rely on two-dimensional datasets. We can, however, compare the technique in this paper to the analogous tensor IHT and gradient formulations applied to traditional

Table 2
Function parameters (Eq. (11)) for polynomial growth fit to 25%, 50%, and 75% convergence rate lines.

p	a	b	c
25%	.0753	2.13	−5.90
50%	.0966	3.15	−3.75
75%	.1178	4.73	−6.33

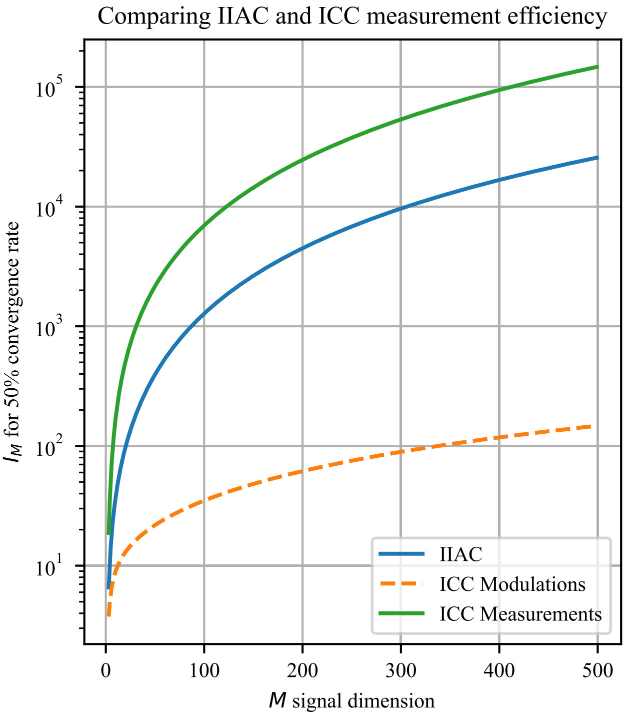

Fig. 5. Comparison of IIAC and ICC median convergence rate $f_5(M)$ in terms of unique modulations and measurements. Here a unique modulation is a new mask at the modulation plane, and a unique measurements is an independent element of the data set $h_i, h_i[k]$ for IIAC and ICC respectively.

two-dimensional pulse characterization datasets. In [11], we demonstrated that gradient and tensor IHT techniques outperformed state-of-the-art alternating minimization techniques in frequency-modulated, frequency-resolved pulse characterization; gradient techniques produced correct results with fewer measurements (measurement efficiency) than alternating minimization. In this section, we compare the measurement efficiency of IIAC to the state-of-the-art technique of ICC. The 50% expected convergence rate lines for IIAC and ICC are listed below and plotted in Fig. 5.

- IIAC: $I_M = .0966M^2 + 3.15M - 3.75$
- ICC (Two variable): $I_M = (.041M + 3.7)\log(.89M)$ [11]

From Fig. 5, we see that ICC can recover signals with a higher modulation efficiency than IIAC. At this point it is important to remember that while ICC requires fewer modulations I than IIAC, each modulation in ICC generates $2M - 1$ measurements. If we compare the total number of measurements, IIAC requires fewer total measurements than ICC. We then see a trade-off between the two system types. If modulations are typically difficult to generate or calibrate, ICC offers a technique that is more efficient in returning signals from fewer modulations. If modulations are easy to generate, IIAC offers a smaller scale optimization than ICC to solve for signals of equivalent size M. We therefore assert that ICC is more information dense per modulation, and IIAC is more information dense per measurement. An important improvement from ICC to IIAC to reiterate is the simplified system setup, as IIAC requires only an integrator instead of a spectrometer. This not only decreases the complexity of the system, but the required power to generate measurements.

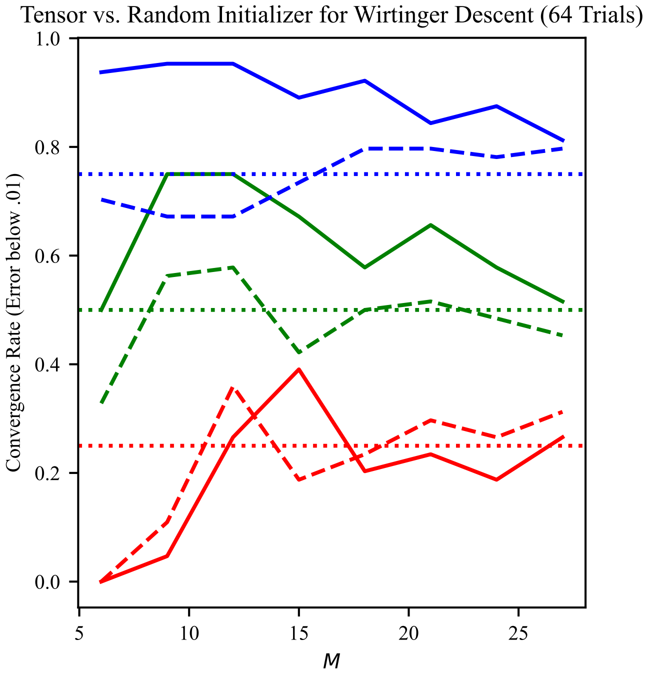

Fig. 6. Tensor initializers (solid lines) are compared to randomly initialized (dashed lines) Wirtinger gradient descents for the IIAC problem. An experiment is conducted for various M at I corresponding to empirically measured $25\%, 50\%, 75\%$ convergence rates corresponding the red, green, and blue sets respectively. Nominal rates are depicted as dotted lines. Wirtinger descents that are initialized with iterative hard tensor thresholding have a consistently higher convergence rate than their randomly initialized counter-parts, with exception of the 25% convergence rate remaining about equal.

4.3. Tensor initialization

Testing fourth-order tensor iterative solvers of the type outlined in Algorithm 1 is very costly in time and compute resources. Instead of a broad I, m investigation, we aimed to demonstrate whether or not the technique was effective at increasing convergence rate for gradient solvers at a small selection of I, M points. Specifically, for identically generated ground-truth and measurement sets, a randomly initialized gradient descent and a tensor initialized gradient descent would run over select M for I corresponding to the hypothetical $25\%, 50\%, 75\%$ convergence rates determined in Section 4.2. This process is outlined in Algorithm 4. For this experiment, $M = [6, 9, 12, 15, 18, 21, 24, 27]$. Results of this experiment are shown in Fig. 6.

Fig. 6 demonstrates that the convergence rates for descents initialized with iterative hard tensor thresholding out-perform randomly initialized descents for identical datasets over 64 trials. The advantage appears to only be noticeable for I, M pairs where randomly initialized Wirtinger descent would achieve 50% or greater convergence rates. While expensive to compute and dense in memory, tensor initializers appear to contribute information complementary to gradient descent. From Fig. 6 it appears that the tensor initializer for the 50% and 75% convergence rates appears to become less effective as M increases. We believe this is because the termination condition *thresh* in Algorithm 1 compares absolute error in the forward projection, rather than error that is normalized by I.

While IHT appears to increase the rate of convergence for gradient based recovery from IIAC measurements, we do not yet have a comprehensive set of theoretical performance guarantees. Previous approaches to providing convergence guarantees [18] have relied on sub-Gaussian and Fourier measurement tensors, which are not as structured as the ones we face in this paper. Measurement tensors $\mathcal{V}_i$ described in this paper are structured in such a way that much of the signal matrix $\mathcal{X}_0$ is in the null space of $\mathcal{V}_i$. This sparse structure prevents much of the signal tensor from being encoded uniquely in measurements and complicates guarantees of uniqueness with unwieldy second and fourth order cross-term conditions on x_0. Attempts to prove recovery uniqueness and stability in future works might start from an analysis and bounding of tensor restricted isometry (TRIP) for the given measurement vectors [18,22]. As a step in that direction, we provide some relevant insight below.

The null space of the IIAC operator is confined to any measurement over which x_0 and c_i have a non-overlapping support; i.e. circumstances under which $x_0 \odot c_i$ is zero everywhere. To conclude this, we consider Eq. (3) in the noiseless case and assert that because $h[i]$ is the sum of sub-terms that are nonnegative, $h[i]$ is zero when all sub-terms are zero; i.e. $h[i] = 0 \iff \left|\langle x_0 x_0^T,\ V_i^*[k]\rangle\right|^2 = 0,\ \forall k$. Trivially, this implies that each term inside the absolute value is also zero; $\left|\langle x_0 x_0^T,\ V_i^*[k]\rangle\right|^2 = 0,\ \forall k \iff \langle x_0 x_0^T,\ V_i^*[k]\rangle = 0, \forall k$. The term $\langle x_0 x_0^T,\ V_i^*[k]\rangle$ is the modulated vector $c_i \odot x_0$ convolved with itself and evaluated at the k lag. Since this convolution must equal zero everywhere, we note that the squared Fourier transform of $c_i \odot x_0$ must also be zero everywhere; $\langle x_0 x_0^T,\ V_i^*[k]\rangle = 0, \forall k \iff \mathcal{F}(c_i \odot x_0)^2[\omega] = 0, \forall \omega$. This implies that the Fourier transform of $c_i \odot x_0$ is also zero; $\mathcal{F}(c_i \odot x_0)^2[\omega] = 0, \forall \omega \iff \mathcal{F}(c_i \odot x_0)[\omega] = 0, \forall \omega$. Finally, a Fourier transform that is zero everywhere implies a signal that is zero everywhere: $\mathcal{F}(c_i \odot x_0)[\omega] = 0, \forall \omega \iff (c_i \odot x_0)[n] = 0 \forall n$. We conclude that $h_i[0] = 0$ and $c_i[n]x_0[n] = 0, \forall n$ are equivalent statements that imply each other. Note that this condition is more aggressive than mere orthogonality of c_i and x_0, which does not guarantee a zero measurement alone.

One interesting property is that, despite the nonlinearity of IIAC measurements, perturbations of x_0 confined to the null space of IIAC do not change the measurement $h[i]$. If we label the IIAC operator as $\mathcal{A}_i(x_0) = \sum_{k=-M+1}^{M-1} \left|\langle x_0 x_0^T,\ V_i^*[k]\rangle\right|^2$, we summarize this fact as: $\mathcal{A}_i(x_0 + x_a) = \mathcal{A}_i(x_0)$ if $(x_a \odot c_i)[n] = 0 \forall n$.

Because the null space of the IIAC operator for measurement i is confined to indices n where $c_i[n] = 0$, constraining $c_i[n] \neq 0, \forall n, i$ will eliminate the non-trivial null space of the IIAC forward operator.

4.4. Complexity

Complexity in this problem breaks down very similarly to the ICC problem for both the gradient and tensor based approach [11]. Memory storage and compute cost per gradient step and alternating minimization iteration are summarized in Table 3.

Memory requirements for the gradient method are very light, benefiting from storing the outer product of modulation vectors to have $V_i[k]$ precomputed. This amounts to $\mathcal{O}(IM^2)$ stored complex floats. Per iteration of gradient descent, a series of I quadratic functions are used to weight the diagonals of added matrices that will be used to multiply a vector, resulting in $\mathcal{O}(IM^2)$ complex multiply-adds per gradient calculation.

Memory requirements for the tensor methods are fairly robust, requiring $\mathcal{O}(IM^3)$ values to store the forward and backward operator, and M^4 complex floats to store the structured tensor χ. Computationally, back-projection of a vector with a matrix of size $I \times M^3$ will require $\mathcal{O}(IM^3)$ operations while the HoSVD requires $\mathcal{O}(M^4)$ [23] with our rank known *a priori*.

Ultimately, the gradient techniques are much more scalable than tensor techniques because of their low-resource requirements. Tensor techniques, however, may provide a performance boost for low-measurement scenarios with low signal dimension at the expense of high memory and compute requirements.

Table 3
Order of magnitude compute operations and memory storage requirements for gradient and tensor based approaches to the IIAC problem.

Algorithm	Memory	Compute
Gradient	$\mathcal{O}(IM^2)$	$\mathcal{O}(IM^2)$
Tensor	$\mathcal{O}(IM^3 + M^4)$	$\mathcal{O}(IM^3 + M^4)$

4.5. Robustness to noise

The experiments in this section were conducted in a noiseless setting, and so a brief examination of the effects of noise is presented here. Choices of noise models for phase retrieval problems are typically specified as Gaussian or Poisson [24–26]; we focus on Gaussian noise as a starting point for studying IIAC inversion. In this experiment, we draw $\eta[i]$ from i.i.d. real Gaussian noise over several noise powers and test the effects of noise on estimate error and rate of convergence. We select two fixed values for M ($M = 10, 16$), and allow I to vary ($I \in [4, 150]$) over 64 trials for SNR values of $20, 30, 40, 50, 60, 70, 80$ dB. We define SNR for this problem in Eq. (13). We note that the values of SNR were confined above 20 dB to maintain the validity of the added Gaussian noise model. Noise powers much higher than this introduce the chance of a negative IIAC value and would require additional constraints or a separate noise model.

$$SNR = 10 \log_{10}\left(\frac{\|x_0\|_2^2}{\|\eta\|_2^2}\right) \tag{13}$$

For each experiment, the resulting normalized squared error is presented in Figs. 7 and 8. Note that for each SNR, there is a bimodal distribution of results. We consider the distribution with lower normalized error to be trials that have successfully converged, while the distribution with higher normalized error to be trials that have failed. As Figs. 11 and 12 show below, the successful trials generally correspond to those with higher numbers of measurements I, and the failed trials generally correspond to lower I. For lower SNR scenarios, it is clear that these distributions begin to merge. We note that the normalized squared error (in dB) for successful trials appears to be linear with the SNR of the signal against additive noise. Another note is that because the experiments were performed for separate signal dimension M values over identical I measurement count, the distribution of successful trials to unsuccessful trials is different for each experiment but the location of their concentration is consistent.

For each SNR, we study the effect of increasing noise power on the rate of convergence, fitting GLF curves to distributions as a function of I and SNR. These results are presented in Figs. 9 and 10. We note that the threshold selected for all trials to be considered successful is a normalized squared error of −10 dB, selected because of its consistent separation between distributions shown in Figs. 7 and 8. We see that the GLF functions fit to the results show that the shape of the rate of convergence does not appear to change until approximately 20 dB SNR, at which point the rate of convergence beings to skew toward more measurements.

Finally, we present the value of normalized error (geometric mean over trials) as a function of I and SNR. These results are in Figs. 11 and 12. We note that for all values of SNR, normalized recovery error is inversely related to the number of measurements.

In summary, we observe that normalized squared error is inversely related to the SNR of the input measurements as well as to the number of measurements I. Results for convergence rate estimates match the noiseless cases presented throughout the section for SNRs higher than 30 dB.

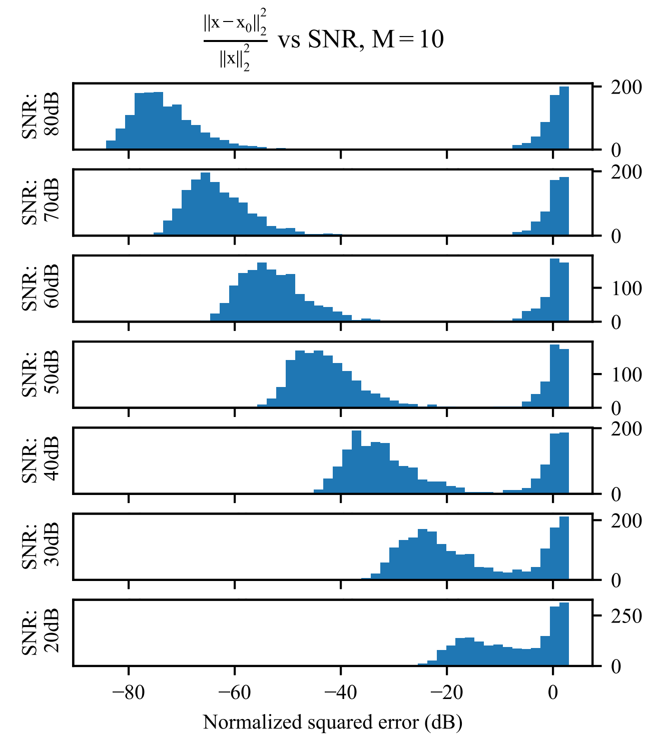

Fig. 7. SNR is varied over measurement count I for a problem dimension $M = 10$. These histograms show the binned normalized squared error for all trials across I for each SNR level.

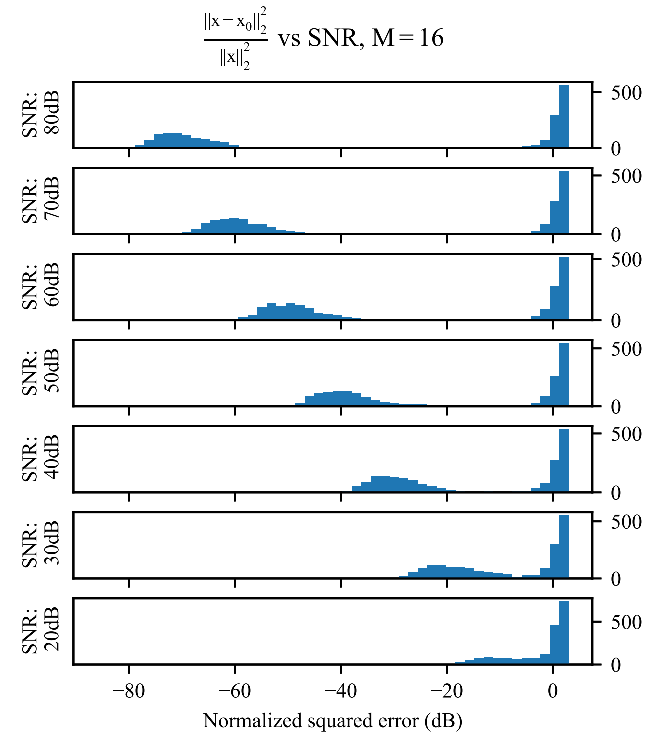

Fig. 8. SNR is varied over measurement count I for a problem dimension $M = 16$. These histograms show the binned normalized squared error for all trials across I for each SNR level.

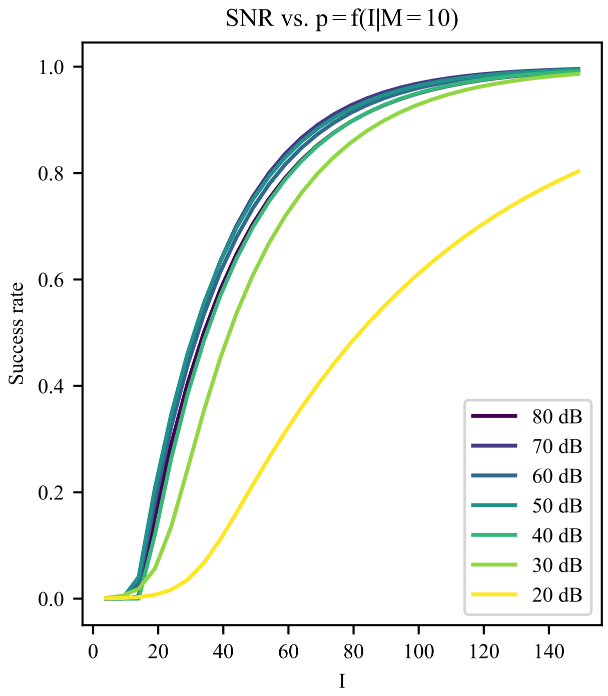

Fig. 9. Curves for estimated rate of convergence lines $p = f(I|M)$ are given as a function of SNR for $M = 10$.

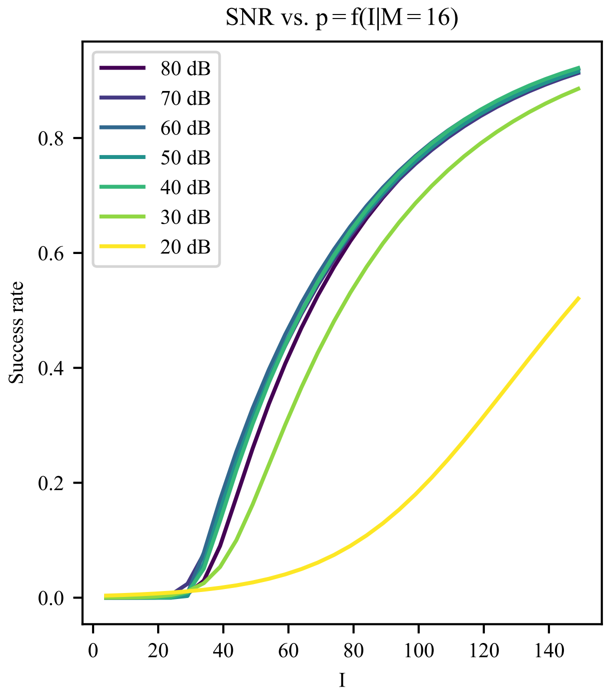

Fig. 10. Curves for estimated rate of convergence lines $p = f(I|M)$ are given as a function of SNR for $M = 16$.

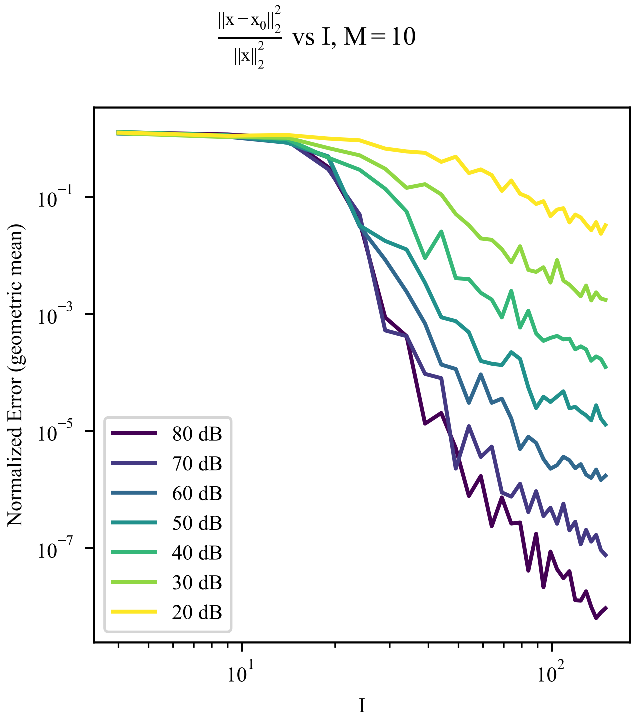

Fig. 11. Normalized error decreases for all SNR as the number of measurements goes up at $M = 10$.

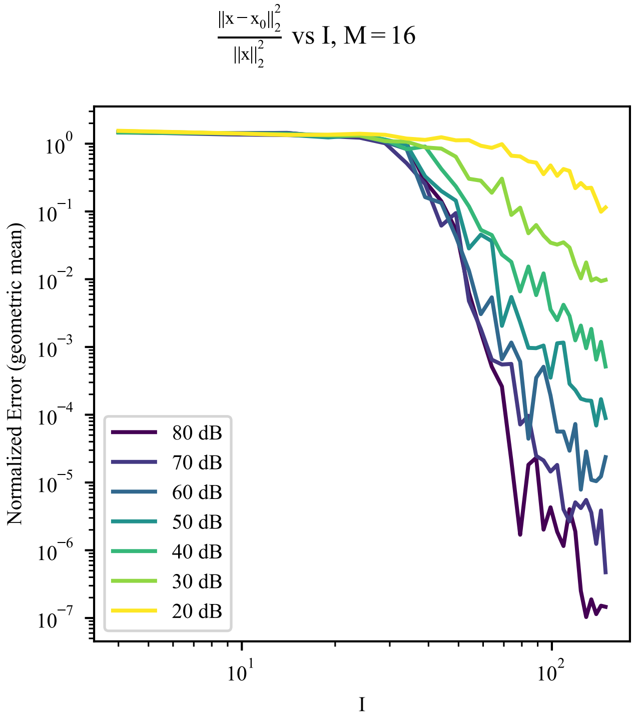

Fig. 12. Normalized error decreases for all SNR as the number of measurements goes up at $M = 16$.

5. Experiment

We constructed a laboratory experiment to determine if a complex pulse spectrum could indeed be recovered from IIAC measurements. The physical setup of the experiment, calibration process, and results are described in this section along with potential improvements. The generating mask and measurements, along with results in both wavelength and time for a synthetic dataset are shown in Figs. 13 and 14, respectively. The mask used in these experiments is a spatial frequency modulated imaging (SPIFI) mask [27–29], a binary continuous cosine basis mask with a radial chirp profile. SPIFI masks are particularly useful for encoding position intensity to a frequency in collected data when the mask is spun and sampled at a constant rate. The mask is well known in the optics community, and used here because of its utility in physically aligning the system. Compared to complex Gaussian modulation masks, SPIFI masks may demonstrate slightly degraded performance, but represent an easily realizable laboratory setup. This purely attenuating mask will be combined with a pure phase mask that results from moving the second grating in the pulse compressor with a linear actuator (DSCAN [14]). While this SPIFI/DSCAN combination produces an IIAC measurement set that is convenient to visualize in two dimensions, it is not a two-dimensional measurement like ICC where every modulation produces a cross-correlation.

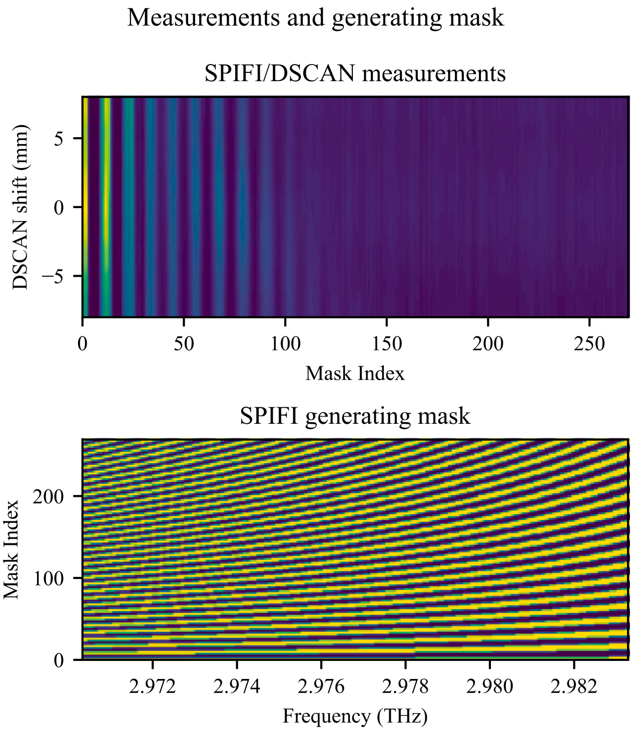

Fig. 13. Synthetic measurement sets were generated to test the feasibility of Wirtinger gradient descent recoveries from IIAC data. The top figure here shows IIAC data generated for two modulators, an alternator referred to as a SPIFI mask [27] and a phase modulation called DSCAN caused by physically translating the pulse compressor's second phase screen. The lower image shows the SPIFI mask profile.

5.1. Physical setup

The experimental pulse characterization system is comprised of several subsections or "arms": spectral modulation arm, delay control arm, calibration arm, and IIAC measurement arm. A diagram of the system is shown in Fig. 15. The source, a ThorLabs FSL1030X1 ytterbium femtosecond fiber laser, produces pulses with a maximum initial pulse duration of 250 fs and a center wavelength of 1030 nm. Flip mirrors (labeled as FM in Fig. 15) are used to select which arms of the system are used without requiring a rebuild and realignment of the system. The first flip mirror can be left up (reflecting) to send the pulses to the spectral modulation subsystem, or down to the delay control arm to generate a copied pair of pulses with a variable time delay. Relative time delay is controlled with a Mach–Zehnder interferometer consisting

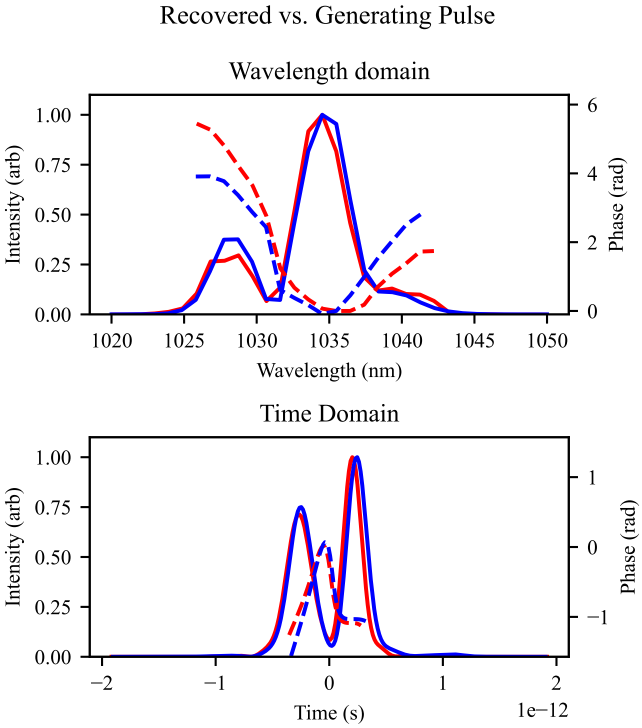

Fig. 14. Recovered pulses in red compare to generating pulses in blue in both time and frequency from the dataset generated in Fig. 13. Frequency measurements have slight variation, but are barely notable in the time domain. Phase in both figures is represented with a dashed line.

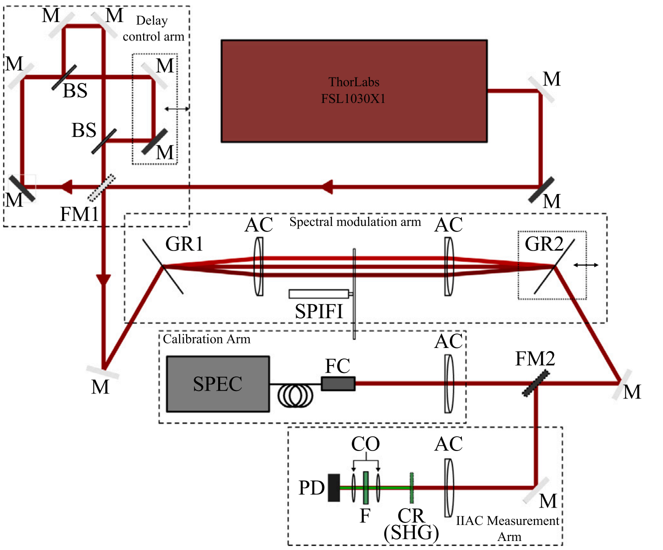

Fig. 15. Diagram of the experimental system. The first flip mirror (FM1) controls whether or not the delay control arm is bypassed for calibration. Passing to the delay control arm creates a duplicate pulse for testing IIAC recovery, bypassing passes a single pulse. The delay line is a Mach–Zhender interferometer with one arm on a motorized stage to move the two mirrors axially, changing the path length to delay a copy of the input pulse. After passing through or bypassing the delay control arm, the pulse travels through the pulse compressor wavelength modulation arm. The SPIFI mask is used to attenuate pulse spectrum and linear stepping of the second pulse compressor grating (GR2) creates a complex phase modulation. The second flip mirror (FM2) directs a pulse to either a calibration spectrometer or the IIAC measurement arm. The spectrometer is used to "image" where the laser pulse is modulated by the SPIFI mask as a function of motor index. M: mirror, FM: flip mirror, BS: 50/50 beam splitter, GR: grating, AC: achromat, FC: fiber coupler, SPEC: spectrometer, CR: crystal (SHG), CO: collection optics, F: filter, PD: photodiode.

of two 50/50 beamsplitters, mirrors, and a linear stage motor (Newport LTA-HS) with a controller (Newport ESP301). This arrangement changes the total path length of one arm of the interferometer to produce a pair of pulses with controlled relative delay.

With or without the use of the delay control arm, the beam goes into the SPIFI modulation arm which is based on a grating and lens pair Martinez pulse compressor [30] consisting of two 1000 lines/mm gratings (Lightsmyth T-1000-1040-31.8 × 24.8-94) rotated to the Littrow angle for the center wavelength of the laser (31 degrees) such that the first diffracted order has the maximum intensity. The first diffracted order of the center wavelength is coaxial with the lens pair consisting of two 100 mm focal length achromatic lenses with an appropriate anti-reflective coating (ThorLabs AC508-100-B-ML). The use of achromatic lenses is important here due to the broad spectrum of the laser: the doublet lenses have lower chromatic aberrations and produce a more consistent focus through the spectrum. At the focal plane between the two lenses, an image of the spectrum of the laser is produced. At this plane, a SPIFI [28] mask mounted on a stepper motor (Trinamic QSH2818) is placed to modulate the spectrum for characterization. The second grating of the pulse compressor is translated along the pulse compressor axis with a linear stage motor and controller (Newport LTS-HS, Newport ESP301) to provide a complex phase modulation to the pulse along with the attenuation provided by the SPIFI mask.

The second flip mirror is then used to select between two different measurement systems: the calibration arm, and the frequency doubling IIAC measurement arm. With the flip mirror down pulses are directed with an achromatic lens through a fiber coupler to a fiber spectrometer (Ocean Optics HR4 series) for calibration mapping of the SPIFI mask indices to particular wavelengths. The IIAC measurement arm uses an achromatic lens to focus pulses into a beta barium borate (BBO) crystal for second harmonic generation (Type I phase matching), converting two photons at 1030 nm to one photon at 515 nm. Collection optics with a filter to block the fundamental 1030 nm light passes the SHG pulse onto a photodiode. The integrated time intensities are digitized using a Digilent Analog Discovery Pro (ADP3450) connected via USB to a PC. Python scripts are used to collect data from the ADP3450 and HR4 spectrometer while synchronizing acquisition with the motor steps of the mask stepper motor.

5.2. Calibration

The modulation mask in the spectral modulation arm needs to be mapped to a measurement index in the collected IIAC dataset. To do this, the spectrometer in the calibration arm takes an image of the modulation plane for a given motor index which maps the attenuation of the SPIFI mask to wavelength.

The phase profile imparted by DSCAN [14] scanning of the second grating mirror does not change the attenuation profile of the SPIFI mask over wavelength. As a function of wavelength, the DSCAN imparted phase profile is known analytically and shown in Eq. (14). Eq. (14) denotes $L[i]$ as the position of the reconstructing grating away from the lens focus. $\theta[i]$ is held at the Littrow angle, and d is the density of the diffraction grating (1000 lines per mm). To disambiguate, c on the right hand of Eq. (14) is the speed of light. These modulations are element-wise multiplied with the attenuation masks produced by the SPIFI mask.

$$c_i[n] = \exp\left(j\frac{nL[i]}{c}\sqrt{1-\left(\frac{2\pi c}{nd}-\sin(\theta[i])\right)^2}\right). \tag{14}$$

To produce the attenuation profile from calibration spectrometer measurements, we remove the pulse wavelength amplitude profile. This flattens attenuation and pass band response so we can measure new spectrum amplitude profiles. The overall calibration procedure is described here:

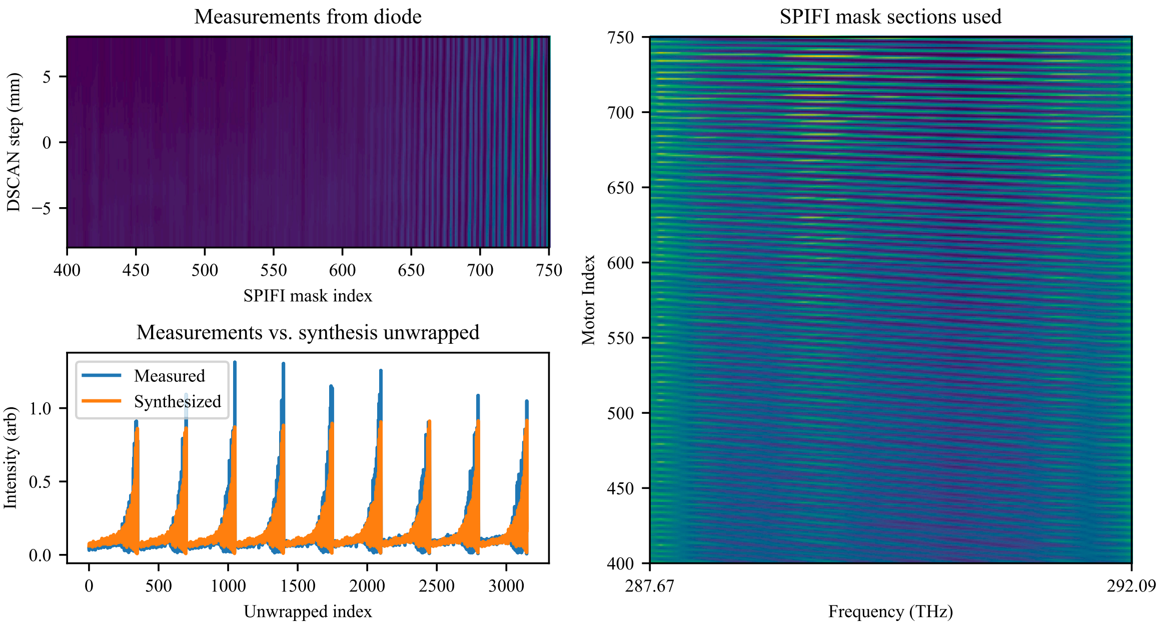

Fig. 16. The top left image is the measurements taken from the diode as a function of DSCAN motor position and SPIFI mask motor index. The right hand picture is taken from the spectrometer at a 0 DSCAN offset and is interpolated to show the mask attenuation as a function of frequency. Note that this has been divided by the max of the profile over the motor index axis to remove the pulse profile. In the bottom left, the unwrapped IIAC dataset from above is compared to the dataset that is generated by the pulse recovered in Fig. 18.

1. Isolate the spectrum illuminated by seeking the left and right bounds at 5% the intensity peak.
2. Take the per-element max over the motor index direction and divide the spectra by this set to remove the pulse spectral profile.
3. Interpolate the wavelength domain values to a frequency domain set, evenly spaced and set by the user for signal dimension M.
4. For each motor mask set, create a set of complex profiles imparted by DSCAN.

This produces a three dimensional calibration set that is M wide, and has two modulation axes including a motor position and a DSCAN position. Vectorizing the motor/DSCAN axes into a single set produces the collection of c_i required to invert h_i IIAC measurements.

5.3. Recovery

A calibration set was taken for the system described in Section 5.1 bypassing the delay control arm. This baseline was used to generate a c_i set with the procedure outlined in Section 5.2. An IIAC dataset was then collected for a pair of pulses spaced 600 fs from each other using the delay control arm.

SPIFI mask motor indices were confined to 400 to 750, shown in Fig. 16, and values of the DSCAN motor shift were $[-8, -6, -4, -2, 0, 2, 4, 6, 8]$ mm. This provides a IIAC measurement set with 3150 measurements, more than six times the measurements needed to expect 50% convergence for $M = 64$ with complex Gaussian measurements ($I_{M=64|p=50\%} = 594$). This oversampling was selected in part because of the unknown behavior that would result from not using complex Gaussian c_i. For two separate optimizations, M was set to 32,64 and their results are shown in Figs. 17 and 18, respectively.

The agreement between the two experiments is very encouraging, and the resulting time domain profile shows two pulses uniquely resolved. The two optimizations were initialized randomly and provide a consistent result over different random initializations. The resulting distance between the pulse peaks is approximately 528.8 fs, low possibly because of interference between the two pulses or due to some experimental error. The path delay error corresponds to a path difference of 21 um. At least 15 fs of error can be attributed to the delay stage motor mount, which lists typical position accuracy at 2.2 μm [31] (4.4 μm total).

5.4. Limitations and future research

From the numerical performance and physical experiments demonstrated in this paper it is clear that while this technique presents a novel approach to pulse characterization, it is not without its limitations. With respect to required measurements for probable inversion, we demonstrate that the number of required measurements I is likely quadratic with respect to M. This implies a limited utility to problems that are substantially larger than the scale tested in this paper. While error in estimates of x_0 is inversely related to SNR, it was also demonstrated that estimate error is inversely related to the number of measurements; thus, the estimate error is limited only by the number of measurements I. Further research would be required to establish a robust rate of convergence that models the effects of SNR. While tensor approaches may improve the ability for a structured recovery, allow for a domain to enforce sparsity conditions, and allow for recovery from fewer measurements, the complexity of such techniques scales as the fourth order of the problem dimension and is extremely limiting for larger problems.

Limitations may exist experimentally at low laser power and/or at low nonlinear signal intensities. Lower levels of signal light following the SHG conversion may be a limitation should the detected intensity be below the noise floor of the detector being used. Attempts to adapt this to lower power experiments will dependent on the detector's sensitivity, quiescent current, background illumination, and sensor noise.

The final limitation we face is sensitivity of inversion to the quality of *a priori* information regarding modulation patterns, as well as potential sensitivity to the modulation basis chosen. Not all SPIFI motor

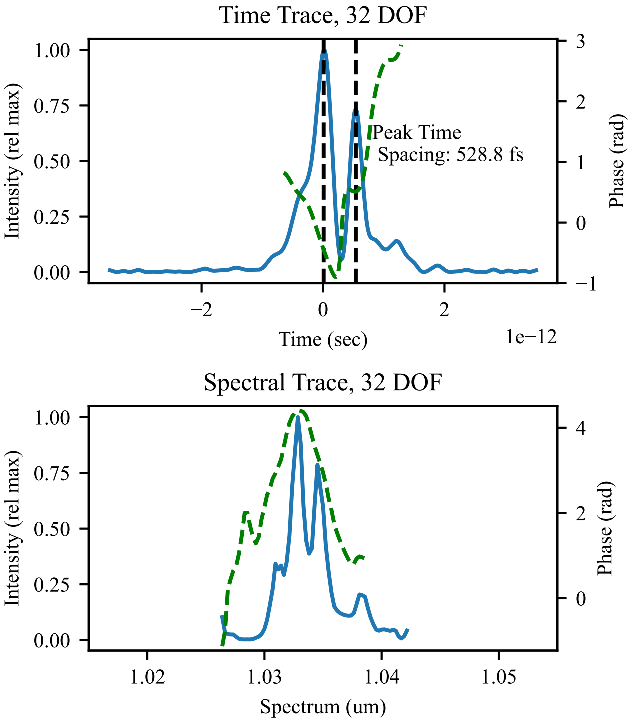

Fig. 17. Double pulse recovered from calibrated setup. $M = 32$ for this recovery. Note that in both the top and the bottom image, the solid blue line represents the intensity of the waveform, while the dashed green line represents the phase.

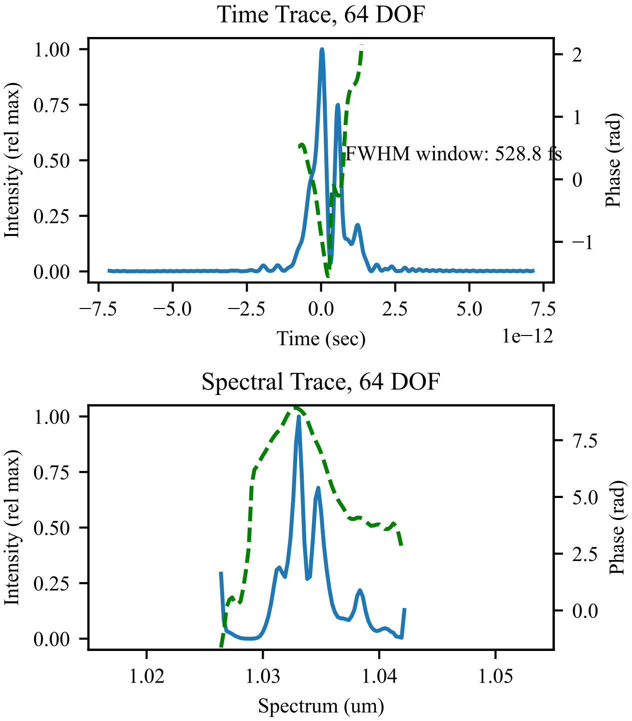

Fig. 18. Double pulse recovered from calibrated setup. $M = 64$ for this recovery. Markers for pulse peaks have been removed for clarity. Note that in both the top and the bottom image, the solid blue line represents the intensity of the waveform, while the dashed green line represents the phase.

indexes were able to produce stable responses for different random initializations and M. SPIFI motor indices could vary between 0 and 800, and represented fading between the high frequency end of the mask and the low frequency end of the mask. Toward the low frequency end, the measurements produced by the largely open sections of mask were much larger in energy than those toward the mid-indices and therefore dominated the least-squares descent. Below about 400, the calibration started to perform poorly as the spectrometer used is a band-limited fiber that could not pass the full harmonic content of the mask at the higher spatial frequencies. The usable range of the SPIFI motor indices then was confined to about 400 to 750. Improved spectrometers for calibration in the future, or aligning the pulse in the spectral modulation arm to be locked to a known mask feature may improve this issue. The motor rotating the SPIFI mask may have also contributed to some error, as even small error in the angular control of the motor toward the higher frequencies of the SPIFI mask would contribute mask phase error. The manufacture lists the step angle accuracy maximum to be 5% [32], making phase error at higher spatial frequencies probable.

In future work we aim to refine the masks used to include a broader set of modulations to more closely mimic the complex Gaussian modulation paradigm for use in a microscopy experiment. We also aim to create a mask that is stable over the directions we expect actuator error to produce a more consistent behavior over modulation indexing.

6. Conclusion

In this paper, we demonstrate that Wirtinger descent and iterative hard tensor thresholding are effective solutions to signal recovery from integrated intensity of auto-convolution. This problem represents a previously unstudied approach to pulse characterization in optics and a new higher-order phase retrieval problem motivating both gradient and structured tensor techniques. Wirtinger descent in particular has demonstrated itself to be a scalable solution to recovering complex laser pulse spectra in a laboratory setting, and the technique may offer a path to pulse characterization in microscopy for the optics community. We provide algorithmic documentation to both the IIAC problem and related IICC problem with the stark belief that new interpretations of non-linear optics sensing paradigms will be solvable with Wirtinger gradient and lifted tensor approaches. We provide theoretical examination of null space conditions of the forward operator and numerical analysis of the techniques performance across problem dimension, measurement count, and SNR. This technique was demonstrated in the laboratory to recover complex pulse time profiles with reasonable accuracy, and plans to improve the technique for use in microscopy are underway. New techniques in spatiotemporal pulse characterization [7] may also be possible with similar gradient techniques allowing for imaging and pulse characterization to happen simultaneously in a calibrated system.

CRediT authorship contribution statement

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

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

Data will be made available on request.

Acknowledgments

We are grateful to Gongguo Tang for his expertise and mentoring in numerical optimization techniques. M. Wakin was supported by National Science Foundation, USA grants CCF-1704204 and CCF-2106834. J. Squier and D. Scarbrough would like to acknowledge support from the Chan Zuckerberg Initiative, USA grant number 2023-321167

Appendix A. Derivation of Wirtinger gradient, Hessian

To derive the Wirtinger gradient and Hessian, we start at their generic definitions [16,33] in Eqs. (15) and (16) and gradually substitute for expressions.

$$\begin{aligned}\nabla f(x) &= \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial x^*} \end{pmatrix}^H \\ &= \begin{pmatrix} \left(\frac{\partial f}{\partial x}\right)^H \\ \left(\frac{\partial f}{\partial x^*}\right)^H \end{pmatrix} \end{aligned} \tag{15}$$

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right)^H & \frac{\partial}{\partial x^*}\left(\frac{\partial f}{\partial x}\right)^H \\ \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x^*}\right)^H & \frac{\partial}{\partial x^*}\left(\frac{\partial f}{\partial x^*}\right)^H \end{pmatrix} \tag{16}$$

We now substitute $f(x)$ for a sum over error from Eq. (6) and can rewrite each sub-expression in Eqs. (15) and (16) as functions of error in Eqs. (17) through (22).

$$\frac{\partial f}{\partial x} = \frac{\partial}{\partial x}\frac{1}{2}\sum_i e_i^2 = \sum_i e_i \frac{\partial e_i}{\partial x} \tag{17}$$

$$\frac{\partial f}{\partial x^*} = \frac{\partial}{\partial x^*}\frac{1}{2}\sum_i e_i^2 = \sum_i e_i \frac{\partial e_i}{\partial x^*} \tag{18}$$

$$\frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right)^H = \sum_i \left(\frac{\partial e_i}{\partial x}^H \frac{\partial e_i}{\partial x} + e_i \frac{\partial}{\partial x}\frac{\partial e_i}{\partial x}^H \right) \tag{19}$$

$$\frac{\partial}{\partial x^*}\left(\frac{\partial f}{\partial x}\right)^H = \sum_i \left(\frac{\partial e_i}{\partial x}^H \frac{\partial e_i}{\partial x^*} + e_i \frac{\partial}{\partial x^*}\frac{\partial e_i}{\partial x}^H \right) \tag{20}$$

$$\frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x^*}\right)^H = \sum_i \left(\frac{\partial e_i}{\partial x^*}^H \frac{\partial e_i}{\partial x} + e_i \frac{\partial}{\partial x}\frac{\partial e_i}{\partial x^*}^H \right) \tag{21}$$

$$\frac{\partial}{\partial x^*}\left(\frac{\partial f}{\partial x^*}\right)^H = \sum_i \left(\frac{\partial e_i}{\partial x^*}^H \frac{\partial e_i}{\partial x^*} + e_i \frac{\partial}{\partial x^*}\frac{\partial e_i}{\partial x^*}^H \right) \tag{22}$$

The six unique expressions differentiating error are broken out and defined in Eqs. (23) through (28).

$$e_i = h[i] - \sum_{k=-M+1}^{M-1} x^T V_i[k]x \; x^H V_i^*[k]x^*$$

$$\begin{aligned}\frac{\partial e_i}{\partial x} &= -x^T \sum_{k=-M+1}^{M-1} \left(V_i[k] + V_i^T[k]\right) \; x^H V_i^*[k]x^* \\ &= -2x^T \sum_{k=-M+1}^{M-1} V_i[k] \; x^H V_i^*[k]x^* \end{aligned} \tag{23}$$

$$\begin{aligned}\frac{\partial e_i}{\partial x^*} &= -x^H \sum_{k=-M+1}^{M-1} \left(V_i^*[k] + V_i^H[k]\right) \; x^T V_i[k]x \\ &= -2x^H \sum_{k=-M+1}^{M-1} V_i^*[k] \; x^T V_i[k]x \end{aligned} \tag{24}$$

$$\begin{aligned}\frac{\partial}{\partial x}\frac{\partial e_i}{\partial x}^H &= -\sum_{k=-M+1}^{M-1} \left(V_i^*[k] + V_i^H[k]\right) \;\; x^T \left(V_i[k] + V_i^T[k]\right) x^* \\ &= -4\sum_{k=-M+1}^{M-1} V_i^*[k] \; x^T V_i[k]x^* \end{aligned} \tag{25}$$

$$\begin{aligned}\frac{\partial}{\partial x^*}\frac{\partial e_i}{\partial x^*}^H &= -\sum_{k=-M+1}^{M-1} \left(V_i[k] + V_i^T[k]\right) \; x^H \left(V_i^*[k] + V_i^H[k]\right) x \\ &= -4\sum_{k=-M+1}^{M-1} V_i[k] \; x^H V_i^*[k]x \end{aligned} \tag{26}$$

$$\begin{aligned}\frac{\partial}{\partial x^*}\frac{\partial e_i}{\partial x}^H &= -\sum_{k=-M+1}^{M-1} \left(V_i^*[k] + V_i^H[k]\right) \;\; x^T V_i[k]x \\ &= -2\sum_{k=-M+1}^{M-1} V_i^*[k] \; x^T V_i[k]x \end{aligned} \tag{27}$$

$$\begin{aligned}\frac{\partial}{\partial x}\frac{\partial e_i}{\partial x^*}^H &= -\sum_{k=-M+1}^{M-1} \left(V_i[k] + V_i^T[k]\right) \;\; x^H V_i^*[k]x^* \\ &= -2\sum_{k=-M+1}^{M-1} V_i[k] \; x^H V_i^*[k]x^* \end{aligned} \tag{28}$$

Collecting terms we arrive at the same terms of gradient and Hessian presented in Eqs. (7) and (8).

$$\begin{aligned}\nabla f(x) &= \sum_i e_i \begin{pmatrix} \frac{\partial e_i}{\partial x}^H \\ \frac{\partial e_i}{\partial x^*}^H \end{pmatrix} \\ &= -2\sum_i e_i \sum_{k=-M+1}^{M-1} \begin{pmatrix} (x^T V_i[k]x) \; V_i^*[k]x^* \\ (x^H V_i^*[k]x^*) \; V_i[k]x \end{pmatrix} \end{aligned}$$

$$\begin{aligned}\nabla^2 f(x) &= \sum_i \left(\begin{pmatrix} \frac{\partial e_i}{\partial z}^H \\ \frac{\partial e_i}{\partial z^*}^H \end{pmatrix} \begin{pmatrix} \frac{\partial e_i}{\partial z} & \frac{\partial e_i}{\partial z^*} \end{pmatrix} + e_i \begin{pmatrix} \frac{\partial}{\partial z}\frac{\partial e_i}{\partial z}^H & \frac{\partial}{\partial z^*}\frac{\partial e_i}{\partial z}^H \\ \frac{\partial}{\partial z}\frac{\partial e_i}{\partial z^*}^H & \frac{\partial}{\partial z^*}\frac{\partial e_i}{\partial z^*}^H \end{pmatrix} \right) \\ &= \sum_i \left(a_i a_i^H - e_i \sum_{k=-M+1}^{M-1} A_{i,k} \right) \\ a_i &= -2\sum_{k=-M+1}^{M-1} \begin{pmatrix} (x^T V_i[k]x) \; V_i^*[k]x^* \\ (x^H V_i^*[k]x^*) \; V_i[k]x \end{pmatrix} \\ A_{i,k} &= \begin{pmatrix} 4V_i^*[k](x^T V_i[k]x^*) & 2V_i^*[k](x^T V_i[k]x) \\ 2V_i[k](x^H V_i^*[k]x^*) & 4V_i[k](x^H V_i^*[k]x) \end{pmatrix} \end{aligned}$$

Appendix B. Collected algorithms, IIAC

Presented in this section are helper algorithms used to construct the larger algorithms throughout the paper.

B.1. Randomly initialized Wirtinger descent experiment

Algorithm 3 details an I vs. M experiment over multiple trials to determine an estimate of convergence rate for randomly initialized IIAC experiments.

Algorithm 3: $I \times M$ convergence tests for randomly initialized Wirtinger Descent

```
Data:
  $I$, measurement count
  $M$, signal dimension
  Result: $success$, True if $error$ is less than .01
  // Generate measurements
  $x_0, c, h = generateComplexGaussianIIAC(M,\ I)$
  $x_{init} = random(M) + 1j\ random(M)$
  $x_{init} = x_{init}/norm(x_{init}, 2)$
  $x = \min_x($Eq. (6)$, gradient =$ Eq. (7)$, init = x_{init})$
  $x = disambiguateIIAC(x, x_0)$
  $error = norm(x - x_0, 2)/norm(x_0, 2)$
  $success = error < .01$
```

B.2. Comparing randomly initialized vs. tensor initialized Wirtinger descent

Algorithm 4 shows the experimental setup to determine if tensor initializers showed any advantage over random initializers for Wirtinger descent against IIAC measurements.

Algorithm 4: $I \times M$ convergence tests comparing randomly initialized and tensor initialized Wirtinger Descent

```
Data:
$I$, measurement count
$M$, signal dimension
Result: $success_{random}, success_{tensor}$, True if $error_{random}, error_{tensor}$
        is less than .01 respectively
// Generate measurements
$x_0, c, h = generateComplexGaussianIIAC(M,\ I)$
// Randomly Initialized Wirtinger Descent
$x_{init} = random(M) + 1j\ random(M)$
$x_{init} = x_{init}/norm(x_{init}, 2)$
$x_{random} = \min_x(\text{Eq. (6)}, gradient = \text{Eq. (7)}, init = x_{init})$
$x_{random} = disambiguateIIAC(x_{random}, x_0)$
$error_{random} = norm(x_{random} - x_0, 2)/norm(x_0, 2)$
// Tensor Initialized Wirtinger Descent
$x_{init} = RAARIIAC(c, h)$
$x_{tensor} = \min_x(\text{Eq. (6)}, gradient = \text{Eq. (7)}, init = x_{init})$
$x_{tensor} = disambiguateIIAC(x_{tensor}, x_0)$
$error_{tensor} = norm(x_{tensor} - x_0, 2)/norm(x_0, 2)$
// Threshold
$success_{random} = error_{random} < .01$
$success_{tensor} = error_{tensor} < .01$
```

B.3. IIACRank

Algorithm 5 accepts a mixed-rank fourth-order tensor and returns a rank-one tensor built from a single fourth-order outer product. Tucker decomposition is used for the decomposition step.

Algorithm 5: IIAC Rank-one tensor projection (IIACRank)

```
Data:
$\chi \in \mathbb{C}^{M\times M\times M\times M}$
Result: $\chi_{rank}$
$X, X_2, X_3^*, X_4^* = \text{tucker}(\chi,\ \text{rank} = [1, 1, 1, 1])$
$\chi_{rank} = X \otimes X \otimes X^* \otimes X^*$
```

B.4. Back-project

Algorithm 6 accepts a fourth-order tensor, forward and backward operators, and an indexing set to back-project error from measurements onto the generating fourth-order tensor.

Algorithm 6: Error back-projection (Back-project)

```
Data:
$\chi \in \mathbb{C}^{M\times N\times M\times N}, \mathbb{C}^{M\times M\times M\times M}$
$A$, forward operator
$B$, backward operator
$ii, jj, kk, ll$, indexing set
$h[i]$, vectorized measurements
Result: $\chi_{back},\ e \in \mathbb{R}$ Frobenius error
$\chi_{back} \leftarrow \mathbf{0}$
error $= h - A\ \chi[ii, jj, kk, ll]$
$\chi_{back}[ii, jj, kk, ll] = \chi[ii, jj, kk, ll] + B$ error
$e = ||\text{error}||_2^2$
```

B.5. IIACIndex

Algorithm 7 accepts signal dimension M, and produces the index set ii, jj, kk, ll that is used to organize the IIAC forward operator, and provides the four index set for the fourth-order guess tensor χ that the forward operator acts on.

Algorithm 7: IIAC Fourth-order index generation (IIACIndex)

```
Data:
$M$
Result: $ii, jj, kk, ll$, Non-zero indices in IIAC measurement
        tensors
$ii, jj, kk, ll \leftarrow [\ ], [\ ], [\ ], [\ ]$
for( $k = [-M+1,\ -M+2,\ \dots,\ M-1$ ) {
    $l_k = \max(0, -k)$
    $u_k = \min(M, N-k) - 1$
    $col_k = k + [l_k : 1 : u_k]$
    $row_k = M - 1 - [l_k : 1 : u_k]$
    $ii = [\ ii,\ repeat(row_k, len(col_k))]$
    $jj = [\ jj,\ repeat(col_k, len(row_k))]$
    $kk = [\ kk,\ tile(row_k, len(col_k))]$
    $ll = [\ ll,\ tile(col_k, len(row_k))]$
}
```

B.6. ScaleIIAC

Algorithm 8 accepts an arbitrary guess x of ground-truth x_0 and produces a scaled x that produces the measurements $h[i] \in \mathbb{R}^I$ from $c \in \mathbb{C}^{M\times I}$ that have the same norm L_2 norm as $h[i]$.

Algorithm 8: Scale IIAC (ScaleIIAC)

```
Data:
$x \in \mathbb{C}^M$
$c \in \mathbb{C}^{M\times I}$
$h[i]$, vectorized IIAC measurements
Result: $X$ scaled to produce measurements $h^*[i]$ s.t. $|h|_2 = |h^*|_2$
$h^*[i] = 0$
for( $i = [0, 1, \dots, I-1$ ) {
    // Written to match Numpy convolve
    $g_i = convolve(x * c[i, :], x * c[i, :], mode = \}full')$
    $h^*[i] = sum(|g_i|^2)$
}
$scale = mean(h^*/h)^{1/4}$
$x/ = scale$
```

B.7. IIACStart

To initialize an alternating minimization algorithm, Algorithm 9 provides a properly scaled rank-one fourth-order tensor. This tensor is constructed from a randomly generated $x \in \mathbb{C}^M$ that is scaled to create measurements with the same norm as those generated by x_0 and then passed through a structured fourth-order outer product.

Algorithm 9: Random Rank-one IIAC tensor, scaled (IIACStart)

```
Data:
$M$
$h[i]$, vectorized IIAC measurements
Result: $\chi \in \mathbb{C}^{M\times M\times M\times M}$ rank-one tensor to initialize
iterative algorithms that produce measurements on the same
scale as $h[i]$
$x = random(M) + \mathbf{j} * random(M)$
$c = random(I, M) + \mathbf{j} * random(I, M)$
$x = ScaleIIAC(x, c, h)$
$\chi = x \otimes x \otimes x^* \otimes x^*$
```

B.8. DisambiguateIIAC

Algorithm 10 aligns phase and frequency offsets between an estimate x and ground-truth x_0 so that difference can be estimated without the phase and frequency ambiguity inherent to IIAC. Frequency offset is estimated as the median phase step between samples. Resulting phase offset is the phase of the inner product between the demixed estimate and ground-truth.

Algorithm 10: IIAC estimate disambiguation (disambiguateIIAC)

```
Data:
x, estimate of ground-truth
x_0, ground-truth
Result: x, phase and frequency aligned estimate of
        ground-truth
// Functions match typical Numpy format and arguments
$offset_{freq} = median(\ diff(\ unwrap(\ angle(\ x/x_0\ ))))$
$x = x\ *\ exp(-\mathbf{j}\ offset_{freq}\ *\ arange(len(x)))$
$offset_{phase} = angle(x^H x_0)$
$x = x\ *\ exp(\mathbf{j} * offset_{phase})$
```

B.9. GeneratecomplexGaussianIIAC

Algorithm 11 generates a starting ground-truth and measurement set, in addition to creating the initial set of measurements.

Algorithm 11: Ground-truth and measurement set generation (generateComplexGaussianIIAC)

```
Data:
M, signal dimension
I, measurement count
Result: x_0, c, h[i], a ground-truth signal, complex Gaussian
        measurement set, and IIAC measurements
$x_0 = random(M) + \mathbf{j} * random(M)$
$x_0/ = norm(x_0, 2)$
$c = random(I, M) + \mathbf{j} * random(I, M)$
$h[i] = \mathbf{0}^I$
for( $i = [0, 1, ..., I-1$ ) {
    // Written to match Numpy convolve
    $g_i = convolve(x * c[i, :], x * c[i, :], mode = \}full')$
    $h[i] = sum(|g_i|^2)$
}
```

Appendix C. Integrated intensity of cross-correlation

The definition of ICC in Eq. (2) is constructed from a modified form of the intensity of cross-correlation problem, where a technique designed to recover two vectors is modified to recover a single vector. This adaptation stems from the fact that in time-resolved systems, a pulse is multiplied by itself in a transform domain rather than being probed with another pulse as is the case in frequency resolved systems. As such, IIAC is a system that recovers a single unknown vector.

Currently, a physical system that uses integrated intensity measurements to recovery two vectors has not been designed and therefore a technique to recover two vectors from the integrated intensity of cross-correlation (IICC) has not been physically motivated. However, it stands to reason that if IIAC recovery techniques are successful in optical pulse characterization, future research may benefit from having some established techniques for recovery from IICC as well as IIAC. This section defines the IICC problem, and outlines modified algorithms for the recovery of two signals from IICC measurements.

C.1. Formulations

The IICC measurement is defined in quartic vector form for two ground-truth vectors $x_0 \in \mathbb{C}^M, y_0 \in \mathbb{C}^N$ in Eq. (29). Here we assume that each vector can receive I independent modulations c_i, d_i respectively.

$$\begin{aligned} h[i] &= \sum_{k=-M+1}^{N-1} \left| x_0^T \operatorname{diag}(c_i d_i^H,\ k) y_0^* \right|^2 + \eta[i] \\ &= \sum_{k=-M+1}^{N-1} \left| x_0^T \Lambda_i[k] y_0^* \right|^2 + \eta[i] \\ &= \sum_{k=-M+1}^{N-1} x_0^T \Lambda_i[k] y_0^*\ x_0^H \Lambda_i^*[k] y_0 + \eta[i] \end{aligned} \tag{29}$$

We will continue to use the notation that $\Lambda_i[k] = \operatorname{diag}(c_i d_i^H,\ k)$. This definition can also be viewed as the sum of phaseless quadratic matrix inner products as shown in Eq. (30).

$$\begin{aligned} h[i] &= \sum_{k=-M+1}^{N-1} \left| x_0^T \Lambda_i[k] y_0^* \right|^2 + \eta[i] \\ &= \sum_{k=-M+1}^{N-1} \left| \langle x_0 y_0^H, \Lambda_i^*[i] \rangle \right|^2 + \eta[i] \end{aligned} \tag{30}$$

Finally, IICC can be shown to be the linear inner product between a series of modulation fourth-order tensors and a ground-truth fourth-order tensor $\mathcal{Z} = x_0 \otimes y_0^* \otimes x_0^* \otimes y_0$. This adaptation is shown in Eq. (31).

$$\begin{aligned} h[i] &= \sum_{k=-M+1}^{N-1} \left| x_0^T \Lambda_i[k] y_0^* \right|^2 + \eta[i] \\ &= \sum_{k=-M+1}^{N-1} \langle x_0 y_0^H,\ \Lambda_i^*[k] \rangle \langle x_0^* y_0^T,\ \Lambda_i[k] \rangle + \eta[i] \\ &= \sum_{k=-M+1}^{N-1} \langle x_0 \otimes y_0^* \otimes x_0^* \otimes y_0,\ \Lambda_i^*[k] \otimes \Lambda_i[k] \rangle + \eta[i] \\ &= \langle x_0 \otimes y_0^* \otimes x_0^* \otimes y_0,\ \sum_{k=-M+1}^{N-1} \Lambda_i^*[k] \otimes \Lambda_i[k] \rangle + \eta[i] \\ &= \langle \mathcal{Z}, \mathcal{L}_i \rangle + \eta[i] \end{aligned} \tag{31}$$

C.2. Cost-function, gradient, Hessian

With the definition of IICC given in Eq. (29), we construct a least-squares cost-function in Eq. (32) for estimates of ground-truth x, y.

$$\begin{aligned} f(x, y) &= \frac{1}{2} \sum_{i=0}^{I-1} \left(h[i] - \sum_{k=-M+1}^{N-1} x^T \Lambda_i[k] y^*\ x^H \Lambda_i^*[k] y \right)^2. \\ &= \frac{1}{2} \sum_{i=0}^{I-1} e_i^2 \end{aligned} \tag{32}$$

Because our cost-function is a function of two variables, we construct a concatenated vector $z = [x^T, y^T]^T$, form the cost-function as a function of z, take the Wirtinger gradient and Hessian in identical fashion to Appendix A, and replace z with x, y constituents. We have omitted this derivation for sake of brevity, and we present the results for Wirtinger gradient in Eq. (33) and Wirtinger Hessian in Eq. (35). Note the shorthand for modulated cross-correlation at lag variable k is denoted $g_i[k] = x^T \Lambda_i[k] y^*$. or in sum form as $g_i[k] = \sum_{n=\max(0,-k)}^{\min(M,N-k)-1} c_i[n] d_i^*[n+k] x[n] y^*[n+k]$.

$$\nabla f(x, y) = -\sum_{i=0}^{I-1} \left(e_i \sum_{k=-M+1}^{N-1} \begin{pmatrix} g_i[k] \Lambda_i^*[k] y \\ g_i^*[k] \Lambda_i^T[k] x \\ g_i^*[k] \Lambda_i[k] y^* \\ g_i[k] \Lambda_i^H[k] x^* \end{pmatrix} \right) \tag{33}$$

$$= \sum_{i=0}^{I-1} e_i a_i \tag{34}$$

$$\nabla^2 f(x, y) = \sum_{i=0}^{I-1} \left(a_i a_i^H - e_i \left(b_{i,k} b_{i,k}^H + c_{i,k} c_{i,k}^H + D_{i,k} \right) \right) \tag{35}$$

$$b_{i,k},\ c_{i,k} = \begin{pmatrix} \Lambda_i^*[k]y \\ \mathbf{0} \\ \mathbf{0} \\ \Lambda_i^H[k]x^* \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ \Lambda_i^T[k]x \\ \Lambda_i[k]y^* \\ \mathbf{0} \end{pmatrix}$$

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

C.3. Additional ambiguity

Only one additional ambiguity exists for the IICC problem that is not present for the IIAC problem, ambiguous scaling between x, y. Because the IICC measurement is a integrated intensity of cross-correlation, no measurements are made without coupling between x_0, y_0. This means that the true energy of each cannot be determined, only the energy of their cross-correlation thus making the scale of x, y ambiguous to one another.

C.4. Algorithm variation

The tensor alternating minimization algorithm AltMinIIAC can be modified to initialize the IICC problem, and the resulting AltMinIICC is shown in Algorithm 12. Helper algorithms IICCStart, ScaleIICC, IICCRank, and IICCIndex are presented in Algorithms 13, 14, 15, and 16 respectively.

Algorithm 12: IICC Alternating Minimization (AltMinIICC)

```
Data:
$c \in \mathbb{C}^{I\times M}$
$d \in \mathbb{C}^{I\times N}$
$h[i], i = [0, 1, ..., I-1]$, vectorized IICC measurements
Parameters:
thresh = $1e-6$
maxIts = 1000
Result: $x \in \mathbb{C}^{M}, y \in \mathbb{C}^{N}$
$ii, jj, kk, ll \leftarrow IICCIndex(M, N)$
$A \leftarrow \mathbf{0} \in \mathbb{C}^{I\times Q}, Q = \sum_{k=-M+1}^{N-1} (\min(M, N-k) - \max(0, -k))^2$
for( $i = [0, 1, ..., I-1]$ ) {
    $A[i, :] \leftarrow c[i, ii] * d^*[i, jj] * c^*[i, kk] * d[i, ll]$
}
$A^+ \leftarrow pinv(A)$
$\chi \leftarrow IICCStart(M, N, h[i])$
for( $ii = 0, 1, ..., maxIts-1$ ) {
    $\chi_{back}, error \leftarrow Backproject(\chi, A, A^+, ii, jj, kk, ll, h[i])$
    $\chi \leftarrow IICCRank(\chi_{back})$
    if $error > thresh$ then
        break
}
$x, y^*, x^*, y = $ tucker($\chi$, rank = [1, 1, 1, 1])
$x, y \leftarrow ScaleIICC(x, y, c, d, h[i])$
```

Algorithm 13: Random Rank-one IICC tensor, scaled (IICCStart)

```
Data:
$M, N$
$h[i]$, vectorized IICC measurements
Result: $\chi \in \mathbb{C}^{M\times N\times M\times N}$ rank-one tensor to initialize
    iterative algorithms that produce measurements on the same
    scale as $h[i]$
$x = random(M) + 1j * random(M)$
$y = random(N) + 1j * random(N)$
$c = random(I, M) + 1j * random(I, M)$
$d = random(I, N) + 1j * random(I, N)$
$x, y = ScaleIICC(x, y, c, d)$
$\chi = x \otimes y^* \otimes x^* \otimes y$
```

Algorithm 14: Scale IICC (ScaleIICC)

```
Data:
$X, Y \in \mathbb{C}^{M}, \mathbb{C}^{N}$
$c, d \in \mathbb{C}^{M\times I}, \mathbb{C}^{N\times I}$
$h[i]$, vectorized IICC measurements
Result: $X, Y$ scaled to produce measurements on same scale as
        $h[i]$
$h_{tmp}[i] = 0$
for( $i = [0, 1, ..., I-1]$ ) {
    // Written to match Numpy correlate
    $h_{tmp}[i] = sum(|correlate(y * d[i, :], x * c[i, :], mode =' full')|^2)$
}
$scale = mean(h_{tmp}/h)^{1/4}$
$X/ = scale$
$Y/ = scale$
```

Algorithm 15: IICC Rank-one tensor projection (IICCRank)

```
Data:
$\chi \in \mathbb{C}^{M\times N\times M\times N}$
Result: $\chi_{rank}$
$X, Y^*, X_2, Y_2 = $ tucker($\chi$, rank = [1, 1, 1, 1])
$\chi_{rank} = X \otimes Y^* \otimes X^* \otimes Y$
```

Algorithm 16: IICC Fourth-order index generation (IICCIndex)

```
Data:
$M, N$
Result: $ii, jj, kk, ll$, Non-zero indices in IICC measurement
        tensors
$ii, jj, kk, ll \leftarrow [\ ], [\ ], [\ ], [\ ]$
for( $k = [-M+1,\ -M+2,\ ...\ ,\ N-1$ ) {
    $l_k = \max(0, -k)$
    $u_k = \min(M, N-k) - 1$
    $row_k = [l_k\ :\ 1\ :\ u_k]$
    $repeated_k = repeat(rows_k,\ len(rows_k))$
    $tiled_k = tile(rows_k,\ len(rows_k)$
    $ii = [\ ii,\ repeated_k]$
    $jj = [\ jj,\ repeated_k + k]$
    $kk = [\ kk,\ tiled_k]$
    $ll = [\ ll,\ tiled_k + k]$
}
```

References

[1] T. Bendory, D. Edidin, Toward a mathematical theory of the crystallographic phase retrieval problem, SIAM J. Math. Data Sci. 2 (3) (2020) 809–839, http://dx.doi.org/10.1137/20M132136X, arXiv:https://doi.org/10.1137/20M132136X.

[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] P.C. Konda, L. Loetgering, K.C. Zhou, S. Xu, A.R. Harvey, R. Horstmeyer, Fourier ptychography: current applications and future promises, Opt. Express 28 (7) (2020) 9603–9630, http://dx.doi.org/10.1364/OE.386168, URL http://www.osapublishing.org/oe/abstract.cfm?URI=oe-28-7-9603.

[4] J. Dong, L. Valzania, A. Maillard, T.-a. Pham, S. Gigan, M. Unser, Phase retrieval: From computational imaging to machine learning: A tutorial, IEEE Signal Process. Mag. 40 (1) (2023) 45–57, http://dx.doi.org/10.1109/MSP.2022.3219240.

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

[6] I.A. Walmsley, C. Dorrer, Characterization of ultrashort electromagnetic pulses, Adv. Opt. Photon. 1 (2) (2009) 308–437, http://dx.doi.org/10.1364/AOP.1.000308, URL https://opg.optica.org/aop/abstract.cfm?URI=aop-1-2-308.

[7] S.W. Jolly, O. Gobert, F. Quéré, Spatio-temporal characterization of ultrashort laser beams: a tutorial, J. Opt. 22 (10) (2020) 103501, http://dx.doi.org/10.1088/2040-8986/abad08.

[8] T. Bendory, D. Edidin, Y.C. Eldar, On signal reconstruction from FROG measurements, Appl. Comput. Harmon. Anal. 48 (3) (2020) 1030–1044, http://dx.doi.org/10.1016/j.acha.2018.10.003, URL https://www.sciencedirect.com/science/article/pii/S1063520318300915.

[9] 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.

[10] 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.

[11] D. Rosen, M.B. Wakin, Bivariate retrieval from intensity of cross-correlation, Signal Process. 215 (2024) 109267, http://dx.doi.org/10.1016/j.sigpro.2023.109267, URL https://www.sciencedirect.com/science/article/pii/S0165168423003419.

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

[13] 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.

[14] 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.

[15] 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, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21432.

[16] 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.

[17] T. Blumensath, M.E. Davies, Iterative hard thresholding for compressed sensing, Appl. Comput. Harmon. Anal. 27 (3) (2009) 265–274, http://dx.doi.org/10.1016/j.acha.2009.04.002, URL https://www.sciencedirect.com/science/article/pii/S1063520309000384.

[18] H. Rauhut, R. Schneider, Ž. Stojanac, Low rank tensor recovery via iterative hard thresholding, Linear Algebra Appl. 523 (2017) 220–262, http://dx.doi.org/10.1016/j.laa.2017.02.028, URL https://www.sciencedirect.com/science/article/pii/S0024379517301295.

[19] 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.

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

[21] 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.

[22] R. Grotheer, S. Li, A. Ma, D. Needell, J. Qin, Iterative hard thresholding for low CP-rank tensor models, Linear Multilinear Algebra 70 (22) (2022) 7452–7468, http://dx.doi.org/10.1080/03081087.2021.1992335, arXiv:https://doi.org/10.1080/03081087.2021.1992335.

[23] 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.

[24] L.-H. Yeh, L. Tian, Z. Liu, M. Chen, J. Zhong, L. Waller, Experimental robustness of Fourier ptychographic phase retrieval algorithms, in: Imaging and Applied Optics 2015, Optica Publishing Group, 2015, p. CW4E.2, http://dx.doi.org/10.1364/COSI.2015.CW4E.2, URL https://opg.optica.org/abstract.cfm?URI=COSI-2015-CW4E.2.

[25] J. Dong, L. Valzania, A. Maillard, T.-a. Pham, S. Gigan, M. Unser, Phase retrieval: From computational imaging to machine learning: A tutorial, IEEE Signal Process. Mag. 40 (1) (2023) 45–57, http://dx.doi.org/10.1109/MSP.2022.3219240.

[26] Z. Li, K. Lange, J.A. Fessler, Poisson phase retrieval with wirtinger flow, in: 2021 IEEE International Conference on Image Processing, ICIP, 2021, pp. 2828–2832, http://dx.doi.org/10.1109/ICIP42928.2021.9506139.

[27] G. Futia, P. Schlup, D.G. Winters, R.A. Bartels, Spatially-chirped modulation imaging of absorbtion and fluorescent objects on single-element optical detector, Opt. Express 19 (2) (2011) 1626–1640, URL http://www.opticsexpress.org/abstract.cfm?URI=oe-19-2-1626.

[28] N. Worts, J. Czerski, J. Jones, J. Field, R. Bartels, J. Squier, Simultaneous multidimensional spatial frequency modulation imaging, Int. J. Optomechatron. 14 (2020) 1–17, http://dx.doi.org/10.1080/15599612.2019.1694610.

[29] J.J. Field, K.A. Wernsing, S.R. Domingue, A.M. Allende Motz, K.F. DeLuca, D.H. Levi, J.G. DeLuca, M.D. Young, J.A. Squier, R.A. Bartels, Superresolved multiphoton microscopy with spatial frequency-modulated imaging, Proc. Natl. Acad. Sci. 113 (24) (2016) 6605–6610, http://dx.doi.org/10.1073/pnas.1602811113.

[30] O. Martinez, 3000 Times grating compressor with positive group velocity dispersion: Application to fiber compensation in 1.3-1.6 μm region, IEEE J. Quantum Electron. 23 (1) (1987) 59–64, http://dx.doi.org/10.1109/JQE.1987.1073201, URL http://ieeexplore.ieee.org/document/1073201/.

[31] Precision motorized actuators: LTA series, 2023, URL https://www.newport.com/mam/celum/celum_assets/resources/LTA_Data_Sheet.pdf?2.

[32] QSH2818-x-8k hardware manual | document revision V1.50, 2023, URL https://www.analog.com/media/en/technical-documentation/data-sheets/QSH2818-x-8k_datasheet_rev1.50.pdf.

[33] 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.