optica

Common pulse retrieval algorithm: a fast and universal method to retrieve ultrashort pulses: supplementary material

NILS C. GEIB[1,*], MATTHIAS ZILK[1], THOMAS PERTSCH[1,2,3], AND FALK EILENBERGER[1,2,3]

[1] *Institute of Applied Physics, Abbe Center of Photonics, Friedrich Schiller University, Albert-Einstein-Str. 15, 07745 Jena, Germany*
[2] *Fraunhofer Institute for Applied Optics and Precision Engineering IOF, Center for Excellence in Photonics, Albert-Einstein-Str. 7, 07745 Jena, Germany*
[3] *Max Planck School of Photonics, Germany*
[*] *Corresponding author: nils.geib@uni-jena.de*

Published 12 April 2019

This document provides supplementary information to "Common pulse retrieval algorithm: a fast and universal method to retrieve ultrashort pulses," https://doi.org/10.1364/OPTICA.6.000495. It shows additional results and contains additional information that facilitates the re-implementation of COPRA, e.g., the expressions for the gradients used in the main paper. Additionally, we provide information on the creation of the test pulses and the removal of ambiguities to enable the reproduction of our results.

1. ADDITIONAL PNPS METHODS

In the main paper we showed only a selection of the published PNPS methods. Additional non-collinear PNPS methods can be found in Tab. S1. Tab. S2 gives parametrization filters for additional collinear PNPS methods. More can be found in the literature, although they are mostly variants of schemes shown here. In non-comprehensive numerical tests we found that COPRA can be applied to all of them.

2. DISCRETE CALCULATIONS

The discrete calculations are performed by approximating the integral of the continuous Fourier transform by its Riemann sum. This leads to

$$\tilde{E}_n = \mathrm{FT}_{k\to n}(E_k) \equiv \sum_k D_{nk} E_k \quad \text{with} \quad D_{nk} = \frac{\Delta t}{2\pi}\mathrm{e}^{\mathrm{i}\omega_n t_k} \tag{S1}$$

$$E_k = \mathrm{FT}^{-1}_{n\to k}(\tilde{E}_n) \equiv \sum_n D^{-1}_{kn}\tilde{E}_n \quad \text{with} \quad D^{-1}_{kn} = \Delta\omega\,\mathrm{e}^{-\mathrm{i}t_k\omega_n}. \tag{S2}$$

Here, the summations are *not* the usual discrete Fourier transforms (DFT), and D_{nk} and D^{-1}_{kn} are *not* the usual DFT matrices. Rather, by requiring the reciprocity relation $\Delta t\,\Delta\omega = 2\pi/N$ of the grid spacings the sums can be calculated by using a DFT *and* two multiplications with appropriate phase factors that depend on the simulation grid. We have

$$\tilde{E}_n = \frac{\Delta t}{2\pi}\exp(\mathrm{i}nt_0\Delta\omega)\sum_k[\exp(\mathrm{i}t_k\omega_0)E_k]\exp(\mathrm{i}nk\zeta), \tag{S3}$$

$$E_k = \Delta\omega\exp(-\mathrm{i}t_k\omega_0)\sum_n[\exp(-\mathrm{i}nt_0\Delta\omega)\tilde{E}_n]\exp(-\mathrm{i}kn\zeta), \tag{S4}$$

with $\zeta = \Delta t\Delta\omega = 2\pi/N$. Only the last exponential in the sum belongs to the DFT and the other phase factors are required to approximate the continuous Fourier transform on the chosen grid. For the specific choice of $t_0 = -\lfloor N/2\rfloor\Delta t$ and $\omega_0 = -\lfloor N/2\rfloor\Delta\omega$ both phase factors can be replaced by circular shifts of the input and output arrays by $\pm\lfloor N/2\rfloor$. This is what is done by the functions `fftshift` and `ifftshift` that are available in some programming languages. For any other non-trivial choice of t_0 and ω_0 a circular shift by a different amount or the explicit expression in Eq. (S4) has to be used. Detailed expositions of this issue can be found in [1, 2].

Using the discrete approximations for the Fourier transform we can then calculate the discrete PNPS signal from $\tilde{\mathbf{E}}$. We give two examples below. In the non-collinear case it fully depends on the method. For example, for SHG-FROG it is given by

$$S_{mk} = \mathrm{FT}^{-1}_{n\to k}[\exp(\mathrm{i}\tau_m\omega_n)\,\tilde{E}_n]\,\mathrm{FT}^{-1}_{n\to k}[\tilde{E}_n] \approx \mathcal{S}_{\tau_m}[\tilde{E}](t_k). \tag{S5}$$

Table S1. Signal operator for additional non-collinear schemes.

Method	$\mathcal{S}_\delta[\tilde{E}]$
SD-FROG	$\mathcal{F}^{-1}\left[e^{i\tau\omega}\tilde{E}\right]^2\,\mathcal{F}^{-1}\left[\tilde{E}\right]^*$
THG-FROG	$\mathcal{F}^{-1}\left[e^{i\tau\omega}\tilde{E}\right]^2\,\mathcal{F}^{-1}\left[\tilde{E}\right]$

The pulse delay τ is the parameter δ in these methods.

Table S2. Parametrization filter for additional collinear schemes.

Scheme	Parameter δ	$\tilde{H}_\delta(\omega)$
chirp scan [3]	chirp C	$\exp(iC/2\,\omega^2)$
bFROG [4]	delay τ	$[1+\exp(i\tau\omega)]/2$
πFROG [5]	delay τ	$[B(\omega+\Omega_0)+\exp(i\tau\omega)]/2$

In the collinear case we define the discretely evaluated parametrization filter

$$\tilde{H}_{mn} = \tilde{H}_{\delta_m}(\omega_n). \tag{S6}$$

With that we can calculate S_{mk} for all collinear methods using SHG as

$$S_{mk} = (\mathrm{FT}^{-1}_{n\to k}[\tilde{H}_{mn}\tilde{E}_n])^2. \tag{S7}$$

This shows that once the calculation of SHG-iFROG is implemented, it can be used to calculate SHG-d-scan traces by simply exchanging $\tilde{H}_{mn}$.

3. GRADIENTS

COPRA requires the calculation of $\nabla_n Z_m$, which is the gradient of Z_m with respect to $\tilde{\mathbf{E}}$. As Z_m is not holomorphic the expression $\nabla_n Z_m$ represents a Wirtinger derivative [6]. It can be calculated by viewing $\tilde{\mathbf{E}}$ and its complex conjugate $\tilde{\mathbf{E}}^*$ as independent variables and evaluating

$$\nabla_n Z_m = 2\frac{\partial Z_m}{\partial \tilde{E}_n^*}. \tag{S8}$$

Although the calculation is straightforward we think that providing the expressions here will make the implementation of the algorithm and the reproduction of our results easier. We will provide the calculation of the gradient for SHG-FROG in full and only state the result in the other cases.

In general $\nabla_n Z_m$ takes the following form

$$\begin{aligned}\nabla_n Z_m &= 2\frac{\partial}{\partial \tilde{E}_n^*}\sum_k \Delta S_{mk}\,\Delta S_{mk}^* \\ &= -2\sum_k \Delta S_{mk}^*\frac{\partial S_{mk}}{\partial \tilde{E}_n^*} + \Delta S_{mk}\left(\frac{\partial S_{mk}}{\partial \tilde{E}_n}\right)^*,\end{aligned} \tag{S9}$$

where

$$\Delta S_{mk} = S'_{mk} - S_{mk}. \tag{S10}$$

For methods in which S_{mk} does not explicitly depend on E_n^*, e.g., those based on SHG and THG, the first term in Eq. (S9) vanishes. In the calculations we use the orthogonality of the Fourier matrices

$$\sum_k D_{mk}D^{-1}_{kn} = \delta_{mn}. \tag{S11}$$

Furthermore, we use that the matrices for forward and backward transformation are related by

$$D^{-1}_{nm} = \frac{2\pi\Delta\omega}{\Delta t}\,[D_{mn}]^*. \tag{S12}$$

A. Non-collinear Methods

We define the shifted pulse (usually denoted $E(t_k-\tau_m)$ in the FROG literature) as

$$\begin{aligned}A_{mk} &= \mathrm{FT}^{-1}_{n\to k}[\exp(i\tau_m\omega_n)\tilde{E}_n] \qquad\qquad (\mathrm{S13})\\ &= \sum_n D^{-1}_{kn}\exp(i\tau_m\omega_n)\tilde{E}_n \qquad\qquad (\mathrm{S14})\end{aligned}$$

SHG-FROG We have $S_{mk} = A_{mk}E_k$. The derivatives of S_{mk} are

$$\frac{\partial S_{mk}}{\partial \tilde{E}_n^*} = 0 \tag{S15}$$

$$\begin{aligned}\frac{\partial S_{mk}}{\partial \tilde{E}_n} &= \frac{\partial}{\partial \tilde{E}_n}A_{mk}E_k \qquad\qquad (\mathrm{S16})\\ &= \frac{\partial}{\partial \tilde{E}_n}\left[\sum_l D^{-1}_{kl}\exp(i\tau_m\omega_l)\tilde{E}_l\right]\left[\sum_j D^{-1}_{kj}\tilde{E}_j\right] \qquad (\mathrm{S17})\\ &= D^{-1}_{kn}\exp(i\tau_m\omega_n)E_k + D^{-1}_{kn}A_{mk}. \qquad\qquad (\mathrm{S18})\end{aligned}$$

Plugging this result in Eq. (S9) we obtain

$$\nabla_n Z_m = -2\sum_k \Delta S_{mk}\left[D^{-1}_{kn}\exp(i\tau_m\omega_n)E_k + D^{-1}_{kn}A_{mk}\right]^* \tag{S19}$$

and using Eq. (S12)

$$\begin{aligned}&= -\frac{4\pi\Delta\omega}{\Delta t}\{\exp(-i\tau_m\omega_n)\sum_k D_{nk}\Delta S_{mk}E_k^* \\ &\qquad + \sum_k D_{nk}\Delta S_{mk}A_{mk}^*\} \qquad\qquad (\mathrm{S20})\\ &= -\frac{4\pi\Delta\omega}{\Delta t}\{\exp(-i\tau_m\omega_n)\mathrm{FT}_{k\to n}[\Delta S_{mk}\,E_k^*] \\ &\qquad + \mathrm{FT}_{k\to n}[\Delta S_{mk}\,A_{mk}^*]\}. \qquad\qquad (\mathrm{S21})\end{aligned}$$

PG-FROG We have

$$S_{mk} = |A_{mk}|^2E_k. \tag{S22}$$

which leads to

$$\begin{aligned}\nabla_n Z_m = &-\frac{4\pi\Delta\omega}{\Delta t}\{2\,\exp(-i\tau_m\omega_n)\mathrm{FT}_{k\to n}[A_{mk}\,\mathrm{Re}\,(\Delta S_{mk}\,E_k^*)] \\ &+ \mathrm{FT}_{k\to n}[\Delta S_{mk}\,|A_{mk}|^2]\}.\end{aligned} \tag{S23}$$

TDP The expressions for time-domain ptychography are very similar to SHG-FROG, requiring only the replacement of A_{mk} by

$$A_{mk} = \mathrm{FT}^{-1}_{n\to k}[\tilde{B}(\omega_n)\exp(i\tau_m\omega_n)\tilde{E}_n]. \tag{S24}$$

$B(\omega)$ describes the amplitude transmission of the bandpass filter in one correlator arm. The gradient is then given by

$$\begin{aligned}\nabla_n Z_m = &-\frac{4\pi\Delta\omega}{\Delta t}\{\tilde{B}(\omega_n)\exp(-i\tau_m\omega_n)\mathrm{FT}_{k\to n}[\Delta S_{mk}\,E_k^*] \\ &+ \mathrm{FT}_{k\to n}[\Delta S_{mk}\,A_{mk}^*]\}.\end{aligned} \tag{S25}$$

B. Collinear methods

We define the pulse in the time domain after application of the parametrization operator

$$C_{mk} = \mathrm{FT}^{-1}_{n\to k}[\tilde{H}_{mn}\,\tilde{E}_n], \tag{S26}$$

where $\tilde{H}_{mn}$ is the discrete evaluation of the parametrization filter from Eq. S6.

SHG For methods based on SHG we have

$$S_{mk} = (C_{mk})^2, \tag{S27}$$

which leads to

$$\nabla_n Z_m = -\frac{8\pi\Delta\omega}{\Delta t}\,\tilde{H}^*_{mn}\,\mathrm{FT}_{k\to n}[\Delta S_{mk} C^*_{mk}]. \tag{S28}$$

THG For methods based on THG we have

$$S_{mk} = (C_{mk})^3, \tag{S29}$$

which leads to

$$\nabla_n Z_m = -\frac{12\pi\Delta\omega}{\Delta t}\,\tilde{H}^*_{mn}\,\mathrm{FT}_{k\to n}[\Delta S_{mk}(C^*_{mk})^2]. \tag{S30}$$

SD For collinear methods using inline SD we have

$$S_{mk} = |C_{mk}|^2\, C_{mk}, \tag{S31}$$

which leads to

$$\begin{aligned}\nabla_n Z_m = -\frac{4\pi\Delta\omega}{\Delta t}\,\tilde{H}^*_{mn}\mathrm{FT}_{k\to n}\Big[\Delta S^*_{mk}\, C^2_{mk} + \\ 2\,\Delta S_{mk}\,|C_{mk}|^2\Big].\end{aligned} \tag{S32}$$

C. Global iteration

The expression for the gradient $\nabla_{mk} r$ used in the global iteration was obtained using exactly the same method as above. The global gradient $\nabla_n Z$ is obtained by simply summing the above expressions over m

$$\nabla_n Z = \sum_m \nabla_n Z_m. \tag{S33}$$

D. Implementation

Nowadays, most programming languages offer a data type for complex numbers which can be used to implement the expressions above straightforwardly. This is what makes the Wirtinger formalism so attractive. The alternative of giving the expressions in derivatives of $\mathrm{Re}[\tilde{E}_n]$ and $\mathrm{Im}[\tilde{E}_n]$ would make the notation and implementation more cumbersome.

Because ΔS_{mk} has to be calculated in the algorithm before $\nabla_k Z_m$, the values ΔS_{mk}, A_{mk}, E_n, C_{mk} and $\tilde{H}_{mn}$ can be reused. As a result the calculation of the gradient is relatively cheap in terms of computing time. It requires only one (collinear methods) or two (non-collinear methods) additional fast Fourier transforms (FFT).

The scaling factors in front of the gradients are inconsequential and specific to the Fourier transform convention that was used. We provide them merely for completeness.

4. METHODS

A. Test pulses

We aimed to create test pulses that are localized in time and frequency while having a complex amplitude and phase structure in both domains. In this way they serve as a good benchmark on how algorithms pick up spectral *and* temporal amplitude features. One example is shown in Fig. S1.

The pulses were generated from complex, random noise. This is comparable to the first test set described in [7]. In doing so we did not model our test pulses after realistic measurements but rather tried to benchmark the worst-case scenario for a pulse retrieval algorithm. Other strategies to construct test pulses, e.g., a Gaussian spectrum with random spectral phase coefficients, should generally lead to higher retrieval ratios and faster convergence.

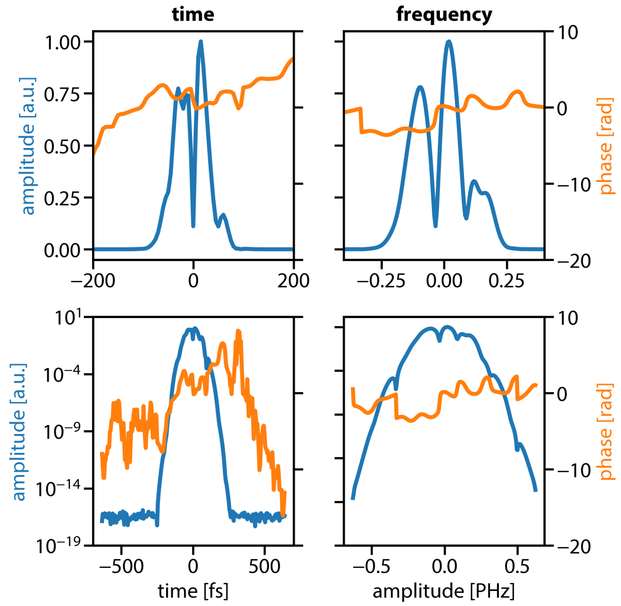


Fig. S1. An example for one of the test pulses with TBP 2. In the bottom row we show the logarithmic scaling to demonstrate the time-frequency localization.

We found the time-frequency localization to be especially important. If the pulse extends to the edges of the simulation grid in any domain, wrap-around will occur in the computation of the trace and the result will usually be unphysical. This problem is made worse when computing d-scan and MIIPS traces as there the pulse is spectrally chirped and, consequently, stretched in the time domain. Curiously, this can improve the conditioning of the retrieval problem and can lead to faster convergence of COPRA – however, the resulting solution is unphysical (see Sec. 7).

To avoid any issues like that we created pulses that dropped off to a relative amplitude of 1e−15 at the edges of the simulation grid in both domains. Additionally, we had to leave enough padding in the time domain so that even the strongly chirped pulses in d-scan could be calculated without wrap-around. This very strict requirement explains why we chose a comparably low TBP of 2 for our simulations. For higher TBPs we would have needed much larger grid sizes, e.g., $N = 1024$ for a TBP of 5. This would have made the simulation of the retrieval of a very large number of pulses infeasible at some point – even when using COPRA. By limiting the TBP we can make sure that the computed traces do not suffer from numerical artifacts and the reported retrieval probabilities are meaningful for the retrieval from experimental traces. An example for retrieval of a pulse with very high TBP is shown in Sec. 8.

The procedure to create the test pulses requires a target TBP and a simulation grid. First, a random array is created in the frequency domain (amplitude uniformly distributed on $[0, 1]$, phase uniformly distributed on $[0, 2\pi]$). This array is multiplied by a Gaussian function that just reaches the required edge values at ω_0 and ω_{N-1}. Then the Fourier transform is calculated and the result is multiplied by a Gaussian function in the time domain. The end result is the test pulse in the time domain.

The width of the second, temporal Gaussian function is roughly determined by the Fourier transform of the spectral one. However, its exact width has to be optimized to result in the specified TBP exactly. This is done with a scalar minimization algorithm.

B. General minimization algorithms

As a complementary approach to COPRA we suggested using general minimization algorithms. This was illustrated with a retrieval comparison from a simple FROG measurement using different minimization algorithms.

Specifically, we used the implementations from the SciPy package for Python in its version 1.1.0 [8]. The NM and BFGS algorithms were run from the scalar minimization wrapper `minimize` with default arguments. We used the DE implementation in the function `differential_evolution` and the parameters described in [9]. We used the sophisticated LM implementation that is provided as the default solver of the `least_squares` function with its default arguments. We found that other implementations of LM may show slightly worse convergence speed (e.g., the one in MINPACK).

The LM algorithm can be provided with an implementation of the Jacobian of T_{mn} for more accurate and efficient retrieval. The required expressions can be obtained using the methods from Sec. 3. However, we found that in this case there is no great advantage in doing so. The calculation of the full Jacobian is computationally expensive and the numerical finite-differences approximations worked very well for us.

On the other hand, the gradient of r required for BFGS can be calculated efficiently – either by symbolic or automatic differentiation. The resulting speed-up makes BFGS an attractive alternative to LM if the extra effort of implementing the gradient is invested. This approach was not investigated in this work.

C. COPRA retrieval tests

The parameters we used for our pulse retrieval simulations for the different methods are listed in Tab. S3. The parameters were chosen after careful consideration of several factors. For SHG-FROG we wanted to do a direct comparison to PCGPA which requires the delay sampling $\tau_m = t_m$. The same was also chosen for PG-FROG, SHG-iFROG and THG-iFROG. For SD-iFROG we found that higher delay sampling is necessary for reliable convergence. For SHG-TDP we used only $M = 128$ delay samples.

For d-scan the maximum glass insertion and the pre-chirp were adapted to the TBP of the pulse, because retrieval worked best when the largest induced GVD was on the order of the pulse chirp. The pre-chirp was simulated by using negative insertion distances. This leads to a slight difference from a true d-scan measurement where pre-chirping is usually done in a grating compressor. However, we found this difference to be inconsequential from the perspective of the retrieval algorithm. For MIIPS we also adapted the parameters α and γ to match the spectral chirp of the pulse.

In general, the parameters we selected are biased in the sense that retrieval for all methods should be possible within the iteration count chosen for the simulation. It is certainly possible to choose parameters that are harder to retrieve. For example, for SD-iFROG for $\tau_m = t_m$ over 1000 iterations would have been necessary to reach comparable trace errors. However, these cases touch the question of which minimal information is necessary for pulse retrieval – something that we did not look at in this work.

Table S3. Parameters for our retrieval simulations.

Methods	Parameters
all	$N = 256$ $\Delta t = 5\,\text{fs}$ $\Delta\omega = 2\pi/(N\Delta t) \approx 4.9\,\text{THz}$ $\lambda_0 = 800\,\text{nm}$
PG-FROG SHG-FROG SHG-iFROG THG-iFROG	$\tau_m = t_m$ $M = N$
SD-iFROG	$\tau_m = t_0 + m\Delta\tau$ $\Delta\tau = (t_{N-1} - t_0)/M$ $M = 4N$
SHG-TDP	$\tau_m = t_0 + m\Delta\tau$ $\Delta\tau = (t_{N-1} - t_0)/M$ $M = 128$ filter center 790 nm filter width 10.6 nm (FWHM)
SHG-d-scan THG-d-scan SD-d-scan	$z_m = (m - M/2 + 0.5)\Delta z$ $\Delta z = 25.0\,\text{mm}/M$ $M = 128$ glass: BK7
SHG-MIIPS THG-MIIPS SD-MIIPS	$\delta = m\Delta\delta$ $\Delta\delta = 2\pi/M$ $M = 128$ $\gamma = 22.5\,\text{fs}$ $\alpha = 1.5\pi$

While we tried to use parameters that are similar to the experiment this was clearly not possible for all PNPS schemes. Our actual values for bandwidth and center wavelength were taken from pulses in our lab and do not correspond to single-cycle pulses. Thus, e.g., the required glass insertion for d-scan is far higher than what has been used in an experiment (and probably also higher than what is feasible). But we stress again that the retrieval is agnostic to these specific numbers as the whole trace and the simulation can be scaled and shifted to different frequencies. Our results should be directly transferrable to the retrieval of single-cycle pulse measurements.

D. Ambiguity removal

After retrieval from synthetic PNPS traces we want to compare the retrieved pulse to the original that was used to create the trace. We used the procedure described in [10] with an additional scaling of the field and a different normalization. The former was necessary as the introduction of the scaling factor μ in our objective function made the absolute scale of the retrieved pulse ambiguous. The latter was chosen to match the convention of the trace error, which was supposed to coincide with the well-known FROG error G. Below we describe our numerical procedure in full.

In the following we will adopt a notation in which arrays are denoted by bold font (e.g., $\tilde{\mathbf{E}}$) and all operations are defined element-wise (e.g., $\rho\tilde{\mathbf{E}}$ and also $\tilde{\mathbf{E}}\tilde{\mathbf{E}}^0$). We assume that all arrays have N elements and define $\boldsymbol{\omega} = (\omega_0, \ldots, \omega_{N-1})$. This notation is imprecise but it should allow an easy implementation in modern programming languages. We start by defining the NRMSE as a function of two arrays

$$\delta(\boldsymbol{x},\boldsymbol{y}) = \Big[\sum_{n=0}^{N-1} |x_n - y_n|^2 / (N \max_n |y_n|^2)\Big]^{1/2}. \tag{S34}$$

The retrieval error ε can then be defined as

$$\varepsilon(\tilde{\mathbf{E}}) \equiv \min_{\rho,\varphi_0,\varphi_1} \delta\{\rho \exp[\mathrm{i}(\varphi_0 + \varphi_1 \boldsymbol{\omega})]\tilde{\mathbf{E}}, \tilde{\mathbf{E}}^0\}. \tag{S35}$$

To calculate it we need two analytical results. First the error

$$\delta(\rho|\tilde{\mathbf{E}}|, |\tilde{\mathbf{E}}^0|) \tag{S36}$$

is minimized by

$$\rho(\tilde{\mathbf{E}}, \tilde{\mathbf{E}}^0) = \big(\sum_n |\tilde{E}_n|\, |\tilde{E}^0_n|\big) / \big(\sum_n |\tilde{E}_n|^2\big). \tag{S37}$$

This calculates the best scaling to match the amplitudes of two complex-valued arrays in the least-squares sense. It can be obtained by straightforward differentiation. Additionally, the error

$$\delta(c_0\tilde{\mathbf{E}}, \tilde{\mathbf{E}}^0) \quad \text{with} \quad |c_0| = 1, c_0 = \exp(\mathrm{i}\varphi_0) \tag{S38}$$

is minimized by

$$c_0'(\tilde{\mathbf{E}}, \tilde{\mathbf{E}}^0) = \big(\sum_n \tilde{E}^0_n \tilde{E}^*_n\big) / \big|\sum_n \tilde{E}^0_n \tilde{E}^*_n\big|, \tag{S39}$$

$$c_0(\tilde{\mathbf{E}}, \tilde{\mathbf{E}}^0) = \begin{cases} c_0'(\tilde{\mathbf{E}}, \tilde{\mathbf{E}}^0) & \delta(c_0\tilde{\mathbf{E}}, \tilde{\mathbf{E}}^0) < \delta(-c_0\tilde{\mathbf{E}}, \tilde{\mathbf{E}}^0) \\ -c_0'(\tilde{\mathbf{E}}, \tilde{\mathbf{E}}^0) & \text{else} \end{cases} \tag{S40}$$

This calculates the optimal constant phase to match two complex-valued arrays in the least-squares sense. Using these results we can proceed to calculate ε. First $\tilde{\mathbf{E}}$ is scaled to match the amplitude of $\tilde{\mathbf{E}}^0$:

$$\tilde{\mathbf{E}} \to \rho(\tilde{\mathbf{E}}, \tilde{\mathbf{E}}^0)\tilde{\mathbf{E}}. \tag{S41}$$

Then we define the objective function

$$O(\varphi_1) = \delta\{c_0(\tilde{\mathbf{E}}', \tilde{\mathbf{E}}^0)\, \tilde{\mathbf{E}}',\ \tilde{\mathbf{E}}^0\} \tag{S42}$$

with

$$\tilde{\mathbf{E}}' \equiv \tilde{\mathbf{E}}'(\varphi_1) = \exp(\mathrm{i}\varphi_1 \boldsymbol{\omega})\tilde{\mathbf{E}}. \tag{S43}$$

This calculates the minimal retrieval error ε for a given linear spectral phase φ_1. It remains to determine the optimal φ_1. For that we sample $O(\varphi_1)$ using $2N$ regularly spaced points between $-\pi/\Delta\omega$ and $\pi/\Delta\omega$ to obtain a bracket that encloses the global minimum. The exact location of the minimum is found by a bisection method. The final retrieval error is O at the optimal φ_1. The direction-of-time ambiguity is taken into account by trying out both variants and selecting the one with the lower retrieval error, i.e., $\varepsilon = \min[\varepsilon(\tilde{\mathbf{E}}), \varepsilon(\tilde{\mathbf{E}}^*)]$.

5. COMMON PULSE RETRIEVAL ALGORITHM

In this section we want to discuss various aspects of COPRA in more detail. This includes design decisions and possible modifications.

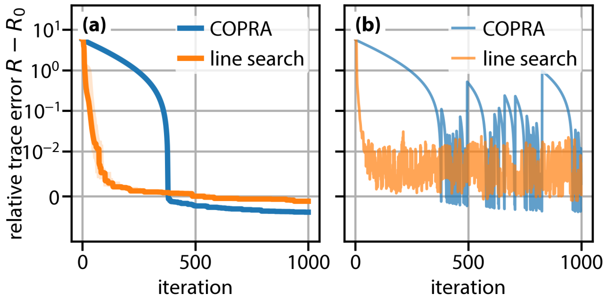


Fig. S2. Convergence behavior for a noisy SHG-FROG trace using the global iteration of COPRA and a modified version using a backtracking line search. (a) the running minimum of the trace error (b) the actual trace error over the iteration count.

Trace error One aspect that was silently disregarded in the main paper is that the trace error R cannot be calculated during the local iteration without extra effort. As $\tilde{\mathbf{E}}$ is changed after processing every spectrum, S_{mn} is calculated based on a different $\tilde{\mathbf{E}}$ for every m. Still, we recommend calculating an approximation to R by using the S_{mn} from the local iteration. As the changes in $\tilde{\mathbf{E}}$ between the iterations become smaller it will get closer to the real R. We observed that it works very well to use this approximation to terminate the local iteration and to select the best solution. All results from the main paper were obtained in this way.

However, one has to be very careful when reporting the error R during or after the local iteration of COPRA. The approximation will regularly underestimate R. Similarly, one has to take care to calculate the correct final trace. It cannot be based on the S_{mn} obtained during the local iteration. Rather, it has to be calculated in full for the retrieved pulse $\tilde{\mathbf{E}}$. In this work, we went to the extra effort and calculated the correct R in every local iteration for logging purposes (this data is shown in the convergence graphs).

We remark that these problems also exist in ptychographical algorithms.

Convergence behavior The convergence graphs in the main paper explicitly show the *running minimum* of the trace error R. Ignoring that, one may get the wrong impression that COPRA reduces the trace error in every iteration. This is not the case. Rather the trace error can fluctuate strongly between iterations. Still, the minimal trace error R will converge. An example of the actual behavior of the trace error in dependence of the iteration count is shown in Fig. S2 b).

Projection The projection on the measured intensities is denoted in the following way in the main paper:

$$S'_{mk} = \mu^{-1/2}\, \mathrm{FT}^{-1}_{n\to k}\left(\tilde{S}_{mn} / |\tilde{S}_{mn}|\, \sqrt{T^{\mathrm{meas}}_{mn}}\right). \tag{S44}$$

Here, the complex-valued square root has to be calculated, effectively mapping negative intensities to imaginary values. Such negative values appear naturally when measuring with CCD spectrometers and subtracting the dark count. It was already observed in the literature that this way of handling them is necessary for an unbiased projection, e.g., in [11].

Spectra iteration order We recommended to iterate randomly over the spectra in the local iteration stage without commenting on the effects. We found that it generally has a positive effect on the retrieval ratio. However, depending on the PNPS scheme other iteration orders can be more beneficial, e.g., iterating the spectra from highest to lowest intensity. However, the effect is small and any iteration order should lead to comparable results.

Step sizes The speed of COPRA benefits from not requiring a line search in every iteration. Rather, we give heuristically safe step sizes γ, η_z, and η_r that show good convergence in our tests. Our step sizes are all based on one specific choice, namely the function value divided by the L2-norm of its gradient. For example, in the global iteration we have

$$\tilde{E}'_n = \tilde{E}_n - \alpha\big(Z/\sum_k |\nabla_k Z|^2\big)\nabla_n Z. \tag{S45}$$

We motivated this step size by performing a Taylor expansion. For example, we expand $Z(\tilde{\mathbf{E}}')$ in $\tilde{\mathbf{E}}$ using η as the step size [6]:

$$Z(\tilde{\mathbf{E}}') \approx Z(\tilde{\mathbf{E}}) + 2\mathrm{Re}\big[\sum_n (\tilde{E}'_n - \tilde{E}_n)^* \, \nabla_n Z\big] \tag{S46}$$

$$Z(\tilde{\mathbf{E}}') \approx Z(\tilde{\mathbf{E}}) + 2\mathrm{Re}\big[\sum_n -\eta(\nabla_n Z)^* \, \nabla_n Z\big] \tag{S47}$$

$$\Rightarrow \quad \eta \approx \big(Z(\tilde{\mathbf{E}}) - Z(\tilde{\mathbf{E}}')\big)/\big(2\sum_n |\nabla_n Z|^2\big). \tag{S48}$$

Now we assume that every iteration substantially decreases Z, i.e., $Z(\tilde{\mathbf{E}}') \ll Z(\tilde{\mathbf{E}})$ to obtain

$$\eta \approx Z(\tilde{\mathbf{E}})/\big(\sum_n |\nabla_n Z|^2\big), \tag{S49}$$

which is the choice we used in COPRA, up to a constant scale. Though we have not found any explicit discussion of this choice in the literature, it is similar to what is recommended as the starting value for a line search in some algorithms [12].

For application in COPRA we scale this η by a factor α. In the local iteration we use $\alpha = 1$ (and name it γ to allow subscripts), in the global iteration by $\alpha = 1/4$ (to obtain η_z and η_r). We found that for many PNPS methods α can be safely increased for the local iteration. This increases the convergence speed but may lead to divergence in some cases. For the global iteration α can be decreased to obtain a more accurate solution. However, this obviously slows down the convergence.

We tested more sophisticated approaches where ν is decreased over the iteration count, e.g, $\alpha \propto (1.0 - \text{iter./max. iter.})$. Sometimes this leads to faster and more accurate convergence, but we did not find a generally applicable strategy.

We also compared our step sizes to those obtained by an inexact line search. Specifically, we modified the global iteration to use a backtracking line search that fulfills the Armijo condition (see [12] p. 37) with $c = 1\mathrm{e}{-4}$ and $\rho = 0.75$. It was used for both the minimization of r and Z. Then we ran only the global iteration of COPRA on a noisy SHG-FROG trace. The results are shown in Fig. S2. We see that the line search initially converges faster – in terms of iteration count, not function evaluations – but then stagnates above the accuracy reached by the original COPRA. However, in Fig. S2 b) we see that the trace error in COPRA fluctuates more strongly. An exact line search could provide additional accuracy but we conclude that the COPRA step sizes are generally an almost optimal choice for the global iteration.

For the local iteration we observed the same, almost optimal behavior only in the noiseless case. When noise is present adapting the stepsize to the local gradient norm results in poor convergence and even leads to divergence. The modification we provided, the usage of the maximum gradient norm in the denominator instead of the local gradient norm, is purely heuristic. We cannot provide any more insight except than that we observed it to work very well.

Ambiguities in pulse retrieval It seems that even though the pulse retrieval problem is ill-posed due to its many ambiguities solving it does not require to remove the ambiguities in retrieval. We observed no clear indication of stagnation or a cyclic behavior due to ambiguous solutions like it happens in classical phase retrieval. In practice, COPRA will simply converge randomly to one of the many ambiguous solutions. On the other hand, our attempts to remove the ambiguities, e.g., by fitting the second derivative of the phase, showed no improvement.

In fact, COPRA even seems to profit slightly when the ambiguities are used to effectively widen the search range. After every iteration we can update the current guess by

$$\tilde{E}'_n = \exp[\mathrm{i}(\varphi_0 + \varphi_1\omega_n)]\tilde{E}_n, \tag{S50}$$

where φ_0 and φ_1 are randomly chosen. We found that this can slightly increase the retrieval probability – but the increase was not significant enough to be included in the default algorithm.

This effect was used when we introduced an additional ambiguity to the algorithm by the scale factor μ in the expression for r. It leads to an effective ambiguity in the scale of $\tilde{\mathbf{E}}$. Originally, it should only serve to assure convergence even when an initial guess with the wrong magnitude was provided (by determining μ once in the first iteration). But we observed that convergence is more reliable when it is updated in every iteration.

Momentum The convergence speed of COPRA can be increased further by calculating an exponentially-decaying sum of the past gradients and using that as the descent step. This is usually likened to a momentum of the iteration and is well-known to increase the learning rate in neural networks [13]. This approach is also used in the extended ptychographical iterative engine [14]. However, as COPRA often converges within 100 iterations we found this additional increase in speed not worth the effort. Thus, it was excluded from the version presented in the paper for brevity. Still, it may be useful for very large retrieval problems where a single iteration is already computationally expensive.

Weighted least-squares In the main paper we studied the influence of constant additive Gaussian measurement noise on the pulse retrieval. This case is commonly assumed in pulse retrieval simulations [5, 15] and corresponds to low intensity measurements with CCD array spectrometers as they are often used for PNPS measurements [16]. In other situations the signal-dependent part of the noise in a measurement will dominate.

COPRA can be modified to solve the weighted least-squares problem to deal with this case. It requires knowledge of the standard deviation σ_{mn} of the Gaussian noise of the measurement T_{mn}, e.g., from repeated measurements. In this case a maximum-likelihood estimate can be obtained by solving the weighted least-squares problem:

$$r = \sum_{m,n} \big[\tilde{T}^{\mathrm{meas}}_{mn} - \mu\tilde{T}_{mn}(\tilde{\mathbf{E}})\big]^2 / \sigma^2_{mn}. \tag{S51}$$

Making COPRA work with this modified r requires two small changes. First, the expression for μ has to be modified to

$$\mu = \left[\sum_{mn} (\tilde{T}^{\mathrm{meas}}_{mn}\, \tilde{T}_{mn})/\sigma^2_{mn}\right] / \left[\sum_{mn} \tilde{T}^2_{mn}/\sigma^2_{mn}\right]. \tag{S52}$$

Second, in the global iteration the expression for the gradient of r has to be adapted to include the weighting

$$\nabla_{mk} r = -4\mu \frac{\Delta t}{2\pi\Delta\omega} \mathrm{FT}^{-1}_{n\to k}\left[\big(\tilde{T}^{\mathrm{meas}}_{mn} - \mu\tilde{T}_{mn}\big)\tilde{S}_{mn}/\sigma^2_{mn}\right]. \tag{S53}$$

After these modifications COPRA will solve the weighted least-squares problem.

Analyzing the retrieval in presence of more complicated noise models, e.g., the combination of additive and multiplicative noise or if the σ_{mn} are not known, is out of the scope of this work. We can only say that if the σ_{mn} are known or approximated well enough the solution obtained by COPRA will again be a maximum-likelihood estimate – and more accurate than the results obtained by other algorithms. We also point out that the impact of moderate levels of multiplicative Gaussian noise on PCGPA or PIE is much smaller than that of additive Gaussian noise. Consequently, the advantage of using COPRA in these situations is not as large.

Noncalibrated traces For single-cycle pulses the non-uniform spectral response of the measurement setup will usually severely impact the measurement. One approach to handle this problem is to retrieve the spectral response function simultaneously to the pulse. This was demonstrated for d-scan measurements and, in fact, exactly the same approach can be used in COPRA [11]. The scaling factor μ in Eq. (13) has to be replaced with a frequency-dependent scaling $\mu(\omega)$ that is calculated by

$$\mu(\omega_n) = \sum_m \left[\tilde{T}^{\text{meas}}_{mn}\, \tilde{T}_{mn}(\tilde{\mathbf{E}})\right] / \sum_m \tilde{T}_{mn}(\tilde{\mathbf{E}})^2.$$

All steps of COPRA stay the same with the exception of using $\mu(\omega)$ instead of μ.

6. PULSE RETRIEVAL ALGORITHMS

Here we describe other fast pulse retrieval algorithms, some of which were compared against COPRA. It mainly serves to provide information on how we implemented them and to discuss their properties. Furthermore, we present them using the notation of the PNPS formalism.

All the algorithms in this section work by projecting on the measured intensity $\tilde{T}^{\text{meas}}$, i.e., S'_{mk} is obtained by Eq. (S44) either for all m in parallel or subsequently for one m at a time. The scale factor μ was included by us as we found it improves convergence for all these algorithms. Its calculation is described in the main paper. For the treatment of negative intensities we refer to the discussion in Sec. 5.

A. Generalized projections algorithm

The generalized projections algorithm (GPA) for FROG [17] can be applied to all FROG variants. We will shortly sketch the SHG version here.

Every GPA iteration starts with calculating S_{mk} from the current solution $\mathbf{E}$ in the time domain. Then a better guess for the PNPS signal S'_{mk} is obtained by projecting on the measurement via Eq. (S44) (data constraint). In the next step $\mathbf{E}$ is updated by minimizing

$$Z = \sum_{mk} |S'_{mk} - S_{mk}|^2 \tag{S54}$$

in terms of $\mathbf{E}$ by a gradient descent step

$$E'_j = E_j - \gamma \nabla^t_j Z. \quad \text{mathematical-form constraint} \tag{S55}$$

Until now no assumption has been made about the delays τ_m or their sampling frequency $\Delta\tau$. An in fact, none is required. The general expression for $\nabla^t_j Z$ can be calculated with the methods described in Sec. 3 and is given by

$$\nabla^t_j Z = \sum_m 2\frac{\partial Z_m}{\partial E^*_j} \tag{S56}$$

$$= -2\sum_m \Big\{\Delta S_{mj} A^*_{mj} + \sum_k \Delta S_{mk}\Big[E_k \sum_n D^{-1}_{kn} D_{nj} \exp(\mathrm{i}\tau_m\omega_n)\Big]^*\Big\}, \tag{S57}$$

with the definitions from Sec. 3. We see that this general gradient expression is complicated and direct evaluation in this form would require a multiplication with a dense $N \times N$ matrix in the second term. To avoid this issue the gradient in COPRA is calculated with respect to $\tilde{\mathbf{E}}$. In GPA the solution is to require $\tau_m = t_m$, which is sometimes understood as a fundamental requirement for FROG measurements. With this specific choice of delays we can simplify the gradient to obtain

$$\nabla^t_j Z = -2\sum_m \Delta S_{mj} A^*_{mj} + \Delta S_{mj'} E^*_{j'} \tag{S58}$$

$$\text{with} \quad j' \equiv j'(k, l = \tau_m/\Delta t) = \begin{cases} j+l & j+l < N \\ j+l-N & j+l \geq N \end{cases},$$

where the second term is circularly shifted by $-\tau_m/\Delta t$. Furthermore, A_{mj} can be expressed as a circular shift of E_k by $\tau_m/\Delta t$:

$$A_{mj} = E_{j'} \tag{S59}$$

$$\text{with} \quad j' \equiv j'(k, l = \tau_m/\Delta t) = \begin{cases} j-l & j \geq l \\ N+j-l & j < l \end{cases}.$$

Using this and assuming a periodic continuation of the involved fields we can express the gradient as

$$\nabla^t_j Z = -2\sum_m S'(t_j, \tau_m)E^*(t_j - \tau_m) - E(t_j)|E(t_j - \tau_m)|^2 +$$

$$S'(t_j + \tau_m, \tau_m)E^*(t_j + \tau_m) - E(t_j)|E(t_j + \tau_m)|^2, \tag{S60}$$

which is a notation more common in the FROG literature (compare [18], pp. 172–173). The step size γ in classical GPA is determined by an exact line search. We note that a more efficient version of GPA can be implemented by using similar step sizes as the ones in COPRA, described in Sec. 5. We did not include GPA in our comparison as we found the algorithm described in the next section to provide the same solutions in terms of trace and pulse error - but after far fewer iterations.

B. Principal components GPA

The principal components generalized projections algorithm (PCGPA) [7] is inspired by GPA and also requires the specific delay sampling $\tau_m = t_m$. After obtaining S'_{mk} by Eq. (S44) it uses the algebraic structure of the FROG trace to update $\mathbf{E}$. Specifically, we can obtain the so-called outer product form from S_{mk} by reversing and performing a circular shift by k for every kth column:

$$S_{mk} \to \hat{S}_{mk} \quad \text{under } m \to \begin{cases} k-m & k \geq m \\ N+k-m & k < m \end{cases} \tag{S61}$$

which is

$$\hat{S}_{mk} = E_m E_k. \tag{S62}$$

This is a direct result from Eq. (S59). For the corrected PNPS signal S'_{mk} this outer product form will not exactly match to a field **E**, however, it can be decomposed in a least-squares sense by a singular value decomposition (SVD). As this approach scales badly with the size of the trace, PCGPA commonly uses a single iteration of the power method to obtain an estimate for the eigenvector associated with the largest eigenvalue. Specifically, this means to perform the following steps to update **E**

$$E_k \to \sum_k (\hat{S}'_{mk})^* E_k \tag{S63}$$

$$E_k \to E_k^* / \big(\sum_l |E_l|^2\big)^{1/2}. \tag{S64}$$

PCGPA can be modified to work with other FROG variants in which case the formulas above would have to be adapted. We found that using a full SVD gives more accuracy and slightly improved retrieval probability. So for small grid sizes it may be the preferable strategy.

C. Ptychographic iterative engine

Recently, several pulse retrieval algorithms based on ptychography have been proposed [15, 19–21]. All of them are based on the ptychographic iterative engine (PIE) [14, 22], which we chose as a common name for these slightly different algorithms. Specifically, in our comparisons we used the version for SHG-FROG from [15, 21] which we describe in the following.

In PIE for SHG-FROG the spectra are processed individually and in random order. For every m the PNPS signal S_{mk} is calculated from the current guess $\tilde{\mathbf{E}}$, followed by a projection on the measured data to obtain S'_{mk}. Then an updated guess for **E** is obtained by

$$E'_k = E_k + \beta\big(A^*_{mk}/\|\mathbf{E}\|^2_{\max}\big)\Delta S_{mk}, \tag{S65}$$

where $\beta \in [0.1, 0.5]$. Comparing this expression to Eq. (S57) we see that it is, in fact, a gradient descent step using only the first term of the GPA gradient with the specific step size

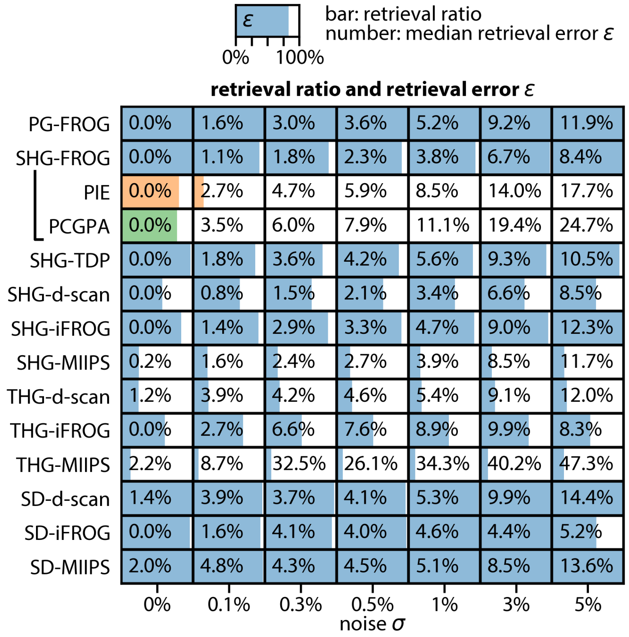
$$\gamma = \beta/\|\mathbf{E}\|^2_{\max}. \tag{S66}$$

The relation of the update step in ptychography to gradient descent has been discussed in [14]. There this specific step size is identified as the Lipschitz constant of the gradient.

The fundamental reason why only one part of the GPA gradient is used, is that in ptychography probe and object pulses, i.e., the delayed and undelayed pulses, are seen as independent variables. Now in PIE for XFROG and TDP both pulses are updated separately using the gradient step from above, which effectively means using both terms of the full gradient. This is very similar to GPA for blind FROG [18].

However, in PIE for SHG-FROG only a single gradient step is performed. For the specific choice $\tau_m = t_m$ this has almost no impact as then both parts of the GPA gradient are approximately the same under the transformation $\tau_m \to -\tau_m$. This can be seen best from Eq. (S60). Therefore, if τ_m and $-\tau_m$ are processed subsequently the algorithm still picks up the full gradient approximately. However, when the delay sampling is not symmetrical, e.g., $\tau_m = t_m + \Delta t/2$ or even $\tau > 0$, we found that the retrieval probability is reduced significantly.

The discussion here should make it clear that from a numerical perspective there is no large conceptual difference between GPA and PIE. The only major distinction is that PIE processes the spectra individually and provides a specific step size for the gradient. In general, the results obtained by both algorithms are also comparable – something that was confirmed in our work.



	0%	0.1%	0.3%	0.5%	1%	3%	5%
PG-FROG	0.0%	1.6%	3.0%	3.6%	5.2%	9.2%	11.9%
SHG-FROG	0.0%	1.1%	1.8%	2.3%	3.8%	6.7%	8.4%
PIE	0.0%	2.7%	4.7%	5.9%	8.5%	14.0%	17.7%
PCGPA	0.0%	3.5%	6.0%	7.9%	11.1%	19.4%	24.7%
SHG-TDP	0.0%	1.8%	3.6%	4.2%	5.6%	9.3%	10.5%
SHG-d-scan	0.0%	0.8%	1.5%	2.1%	3.4%	6.6%	8.5%
SHG-iFROG	0.0%	1.4%	2.9%	3.3%	4.7%	9.0%	12.3%
SHG-MIIPS	0.2%	1.6%	2.4%	2.7%	3.9%	8.5%	11.7%
THG-d-scan	1.2%	3.9%	4.2%	4.6%	5.4%	9.1%	12.0%
THG-iFROG	0.0%	2.7%	6.6%	7.6%	8.9%	9.9%	8.3%
THG-MIIPS	2.2%	8.7%	32.5%	26.1%	34.3%	40.2%	47.3%
SD-d-scan	1.4%	3.9%	3.7%	4.1%	5.3%	9.9%	14.4%
SD-iFROG	0.0%	1.6%	4.1%	4.0%	4.6%	4.4%	5.2%
SD-MIIPS	2.0%	4.8%	4.3%	4.5%	5.1%	8.5%	13.6%

Fig. S3. The retrieval ratio (blue bar) and median retrieval error (inset number) of COPRA in dependence of the PNPS method and the noise level *when starting with a random initial guess*. A comparison with PCGPA and PIE for SHG-FROG is included. Successful retrieval was assumed if $R < R_0 + 1\mathrm{e}{-4}$.

The slightly improved retrieval accuracy of PIE compared to GPA in some cases [21] stems from the individual processing of the spectra and, specifically, the order in which they are iterated. While in GPA all spectra contribute equally, in PIE the spectra that are processed last effectively influence the solution more strongly. So depending on the processing order the retrieval accuracy can be slightly better or slightly worse.

D. GPA for d-scan

Recently, a fast retrieval algorithm for d-scan based on GPA has been proposed [11]. It proceeds by projecting on the measured intensity and then updates $\tilde{\mathbf{E}}$ by a heuristic procedure:

$$S'_{mk} \to S'_{mk}\left(\mathrm{FT}^{-1}_{n\to k}[\tilde{H}_{mn}\tilde{E}_n]\right)^* \tag{S67}$$

$$S'_{mk} \to S'_{mk}/|S'_{mk}|^{2/3} \tag{S68}$$

$$\tilde{E}_n \to \sum_m \tilde{H}^*_{mn}\mathrm{FT}_{k\to n}[S'_{mk}] \tag{S69}$$

We found that when using it for simultaneous amplitude and phase retrieval this algorithm does not converge by the definition used in this work. For noiseless SHG-d-scan traces we found it never obtains $R < 5\mathrm{e}{-3}$ or $\varepsilon < 1\%$. If seeded with the original test pulse it quickly diverges from the numerical limit of $R \approx 1\mathrm{e}{-15}$ and settles at $R \approx 5\mathrm{e}{-3}$. This means that the solution of the pulse retrieval problem is not a fixed point of the algorithm. For these reasons it was not included in a comparison.

7. RANDOM INITIAL GUESS

As described in the main paper we performed a second retrieval simulation in which COPRA was initialized with a random guess (random amplitude and phase, uniformly distributed on $[0, 1]$ and $[0, 2\pi]$). Additionally, for d-scan and MIIPS the iteration count had to be increased from 300 to 1000 to obtain

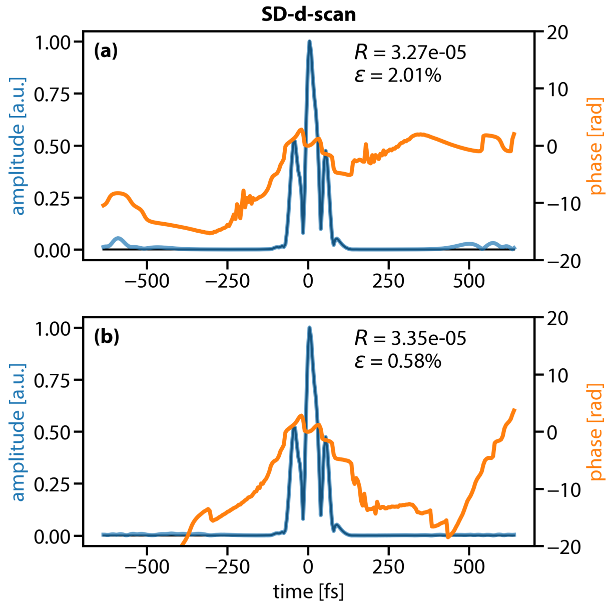


Fig. S4. An example of the conditioning problems in SD-d-scan. Even though the pulse in a) has a lower trace error R than the one in b), it contains spurious satellite pulses and has a much higher retrieval error ε. The black line indicates the amplitude of the original test pulse.

comparable retrieval errors. The retrieval ratios and the median retrieval error are shown in Fig. S3. We see that for many PNPS schemes COPRA performs almost equally well as with the Gaussian initial guess. Specifically, for the non-collinear methods (FROG and TDP) there is no difference at all and for iFROG we only see minor differences in retrieval ratio and retrieval accuracy. This demonstrates that COPRA often works without any knowledge about the measured pulse.

For the d-scan and MIIPS we observe a different behavior. First of all, the retrieval ratio is lower, especially for MIIPS. The reason are local minima of the pulse retrieval problem to which the algorithm converges. Particularly many exist for MIIPS as the applied phase patterns are periodic, admitting local solutions that are spectrally shifted with respect to the global solution. Consequently, they do not appear when starting with a Gaussian pulse that is localized in time and frequency as they do not overlap much with this initial guess.

Secondly, the retrieval accuracy for MIIPS and d-scan is lower than when starting from Gaussian initial guesses. Strikingly, this can be seen for SD-d-scan and SD-MIIPS where even in the noiseless case a high retrieval error remains. The reason is that the retrieval problem is ill-conditioned. Fields at large delays compared to the main pulse, i.e., satellite pulses, contribute only weakly to the PNPS trace for these schemes (see Fig. S4). The solutions obtained in this case have a small oscillating artifact in the frequency domain. They do not constitute non-trivial ambiguities in a strict sense as for noiseless measurements the correct solution can still be found when further reducing the trace error. However, COPRA stagnates when converging towards it. Curiously, this is not the case if the temporal grid is chosen too small and unphysical wrap-around happens in the calculation of the PNPS trace (see Sec. 4.A).

One way to avoid this convergence behavior is to regularize the problem by, e.g., incorporating knowledge about the pulse. This was done implicitly in the main paper by starting from a Gaussian pulse in the time domain which does not contain the problematic contributions at large times. In any other case we found it is sufficient to simply set the first and last 10% of the pulse field to zero after every iteration of COPRA. When testing for non-trivial ambiguities we excluded these artifacts by calculating a modified retrieval error ε' that takes only the central part of the pulse field $\tilde{\mathbf{E}}$ into account.

8. LARGE TIME-BANDWIDTH PRODUCTS

Although not explicitly shown in the main paper, COPRA can, in principle, deal with measurements of pulses of arbitrary TBPs. In fact, in non-comprehensive numerical tests we found no significant dependence of convergence speed and retrieval probability on the TBP. One example for retrieval of a very complex pulse with TBP 25 ($N = 16384$) from a synthetic d-scan trace ($M = 512$) is shown in Fig. S5. The retrieval with 150 iterations required roughly 5 min on an average notebook. Retrieval from measurements this large would probably be practically infeasible with state-of-the-art d-scan retrieval algorithms.

REFERENCES

1. W. L. Briggs and v. E. Henson, *The DFT: an owners' manual for the discrete Fourier transform* (SIAM, 1995).
2. E. W. Hansen, *Fourier transforms: principles and applications* (John Wiley & Sons, 2014).
3. V. Loriot, G. Gitzinger, and N. Forget, "Self-referenced characterization of femtosecond laser pulses by chirp scan," Opt. Express **21**, 24879–24893 (2013).
4. N. Forget, V. Crozatier, and T. Oksenhendler, "Pulse-measurement techniques using a single amplitude and phase spectral shaper," J. Opt. Soc. Am. B **27**, 742–756 (2010).
5. J. Hyyti, E. Escoto, G. Steinmeyer, and T. Witting, "Interferometric time-domain ptychography for ultrafast pulse characterization," Opt. Lett. **42**, 2185–2188 (2017).
6. K. Kreutz-Delgado, "The Complex Gradient Operator and the CR-Calculus," ArXiv e-prints (2009).
7. D. J. Kane, "Real-time measurement of ultrashort laser pulses using principal component generalized projections," IEEE J. Sel. Top. Quantum Electron. **4**, 278–284 (1998).
8. E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," (2001–). http://www.scipy.org/.
9. E. Escoto, A. Tajalli, T. Nagy, and G. Steinmeyer, "Advanced phase retrieval for dispersion scan: a comparative study," J. Opt. Soc. Am. B **35**, 8–19 (2018).
10. C. Dorrer and I. A. Walmsley, "Accuracy criterion for ultrashort pulse characterization techniques: application to spectral phase interferometry for direct electric field reconstruction," J. Opt. Soc. Am. B **19**, 1019–1029 (2002).
11. M. Miranda, J. Penedones, C. Guo, A. Harth, M. Louisy, L. Neoričić, A. L'Huillier, and C. L. Arnold, "Fast iterative retrieval algorithm for ultrashort pulse characterization using dispersion scans," J. Opt. Soc. Am. B **34**, 190–197 (2017).
12. J. Nocedal and S. J. Wright, *Numerical Optimization* (Springer, 2006), 2nd ed.
13. S. Ruder, "An overview of gradient descent optimization algorithms," CoRR **abs/1609.04747** (2016).
14. A. Maiden, D. Johnson, and P. Li, "Further improvements to the ptychographical iterative engine," Optica **4**, 736–745 (2017).
15. P. Sidorenko, O. Lahav, Z. Avnat, and O. Cohen, "Ptychographic reconstruction algorithm for frequency-resolved optical gating: super-resolution and supreme robustness," Optica **3**, 1320–1330 (2016).
16. J. J. Davenport, J. Hodgkinson, J. R. Saffell, and R. P. Tatam, "Noise analysis for CCD-based ultraviolet and visible spectrophotometry," Appl. Opt. **54**, 8135–8144 (2015).
17. K. W. DeLong, B. Kohler, K. Wilson, D. N. Fittinghoff, and R. Trebino, "Pulse retrieval in frequency-resolved optical gating based on the method of generalized projections," Opt. Lett. **19**, 2152–2154 (1994).
18. R. Trebino, *Frequency-Resolved Optical Gating: The Measurement of Ultrashort Laser Pulses* (Springer US, 2000).

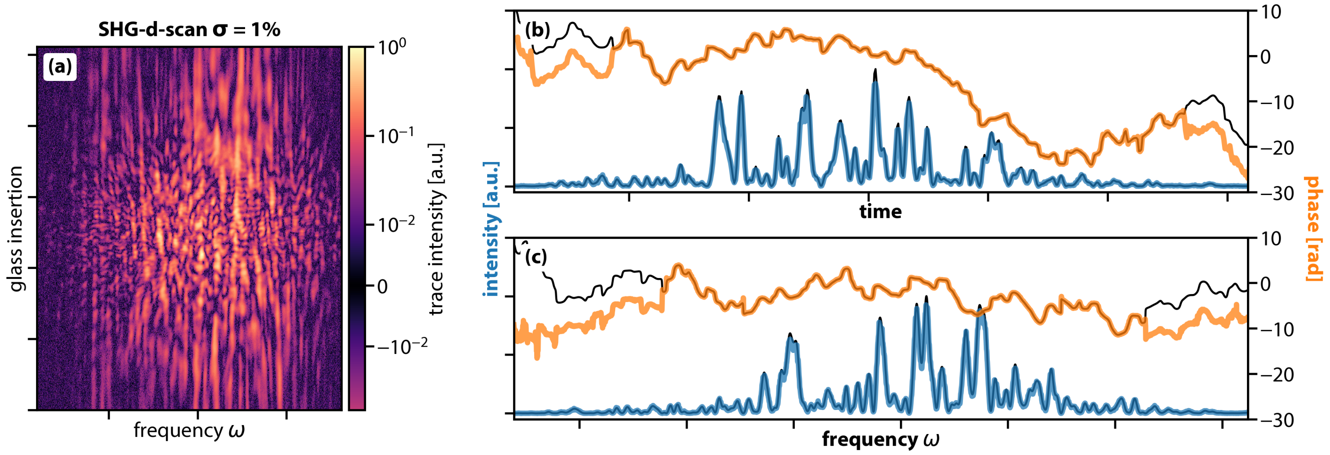


Fig. S5. Example for COPRA retrieval of a very complex pulse with TBP 25 from a large synthetic SHG-d-scan measurement ($N = 16384$, $M = 512$) with a moderate level of Gaussian noise ($\sigma = 1\%$). a) shows the noisy d-scan trace. b) and c) show the test pulse (black) and the retrieved solution (blue: intensity, orange: phase) in the time and frequency domain. The trace error of the solution is $R = 9.82\text{e}{-3} = R_0 + 1.03\text{e}{-7}$, the retrieval error is $\varepsilon = 3.7\%$.

19. D. Spangenberg, E. Rohwer, M. H. Brügmann, and T. Feurer, “Ptychographic ultrafast pulse reconstruction,” Opt. Lett. **40**, 1002–1005 (2015).

20. T. Witting, D. Greening, D. Walke, P. Matia-Hernando, T. Barillot, J. P. Marangos, and J. W. G. Tisch, “Time-domain ptychography of over-octave-spanning laser pulses in the single-cycle regime,” Opt. Lett. **41**, 4218–4221 (2016).

21. P. Sidorenko, O. Lahav, Z. Avnat, and O. Cohen, “Ptychographic reconstruction algorithm for frequency resolved optical gating: super-resolution and extreme robustness: erratum,” Optica **4**, 1388–1389 (2017).

22. A. M. Maiden and J. M. Rodenburg, “An improved ptychographical phase retrieval algorithm for diffractive imaging,” Ultramicroscopy **109**, 1256–1262 (2009).