Research Article Vol. 5, No. 5 / May 2018 / Optica 666

Deep learning reconstruction of ultrashort pulses

Tom Zahavy,[1,3,†] Alex Dikopoltsev,[2,*,†] Daniel Moss,[2] Gil Ilan Haham,[2] Oren Cohen,[2] Shie Mannor,[1] and Mordechai Segev[2]

[1]Department of Electrical Engineering, Technion, Haifa 32000, Israel
[2]Department of Physics and Solid State Institute, Technion, Haifa 32000, Israel
[3]e-mail: tomzahavy@gmail.com
*Corresponding author: alexdiko@technion.ac.il

Received 18 January 2018; revised 4 April 2018; accepted 4 April 2018 (Doc. ID 320074); published 18 May 2018

Ultrashort laser pulses with femtosecond to attosecond pulse duration are the shortest systematic events humans can currently create. Characterization (amplitude and phase) of these pulses is a crucial ingredient in ultrafast science, e.g., exploring chemical reactions and electronic phase transitions. Here, we propose and demonstrate, numerically and experimentally, what is to the best of our knowledge, the first deep neural network technique to reconstruct ultrashort optical pulses. Employing deep neural networks for reconstruction of ultrashort pulses enables diagnostics of very weak pulses and offers new possibilities, e.g., reconstruction of pulses using measurement devices without knowing in advance the relations between the pulses and the measured signals. Finally, we demonstrate the ability to reconstruct ultrashort pulses from their experimentally measured frequency-resolved optical gating traces via deep networks that have been trained on simulated data.

OCIS codes: (320.0320) Ultrafast optics; (320.7100) Ultrafast measurements.

https://doi.org/10.1364/OPTICA.5.000666

1. INTRODUCTION

Ultrashort laser pulses [1] are among the shortest systematic events that can currently be created. These pulses are currently used in numerous applications in ultrafast science, including femtochemistry [2], coherent control [3], high-harmonic spectroscopy [4] and ultrafast imaging [5]. The term "ultrashort pulse" typically refers to a light pulse whose duration is below picoseconds (10^{-12} s). In this regime, pulses are too short to be measured directly by photodiodes. Naturally, it is highly important to characterize these pulses, but this is possible only through indirect measurements. A popular technique for characterizing ultrashort pulses is frequency-resolved optical gating (FROG) [6]. It is an established [7] and recently mathematically proven [8] approach for full characterization of the amplitude and phase of ultrashort optical pulses. It involves creating a 2D intensity diagram, called the FROG trace, by spectrally measuring a nonlinear product field of a pulse with time-shifted replicas of itself. The data acquired by FROG (or its variations such as (GRating-Eliminated Nonsense Observation of Ultrafast Incident Laser Light E-fields) GRENOUILLE [9] and frequency-resolved optical gating for complete reconstruction of attosecond bursts (FROG-CRAB) [10]) incorporate enough information to resolve the ultrashort pulse reconstruction up to "trivial" ambiguities: time shift, constant phase, and in some schemes, time flip with complex conjugation [7]. The reconstruction of the pulse from the FROG measurement requires a recovery algorithm. Among such retrieval algorithms, the principal component general projections algorithm (PCGPA) [11] is one of the typically deployed algorithms, although it requires a full spectrogram that fulfills the sampling Fourier relations in the frequency-time domain, and therefore has to satisfy constraints on the number of measurements. Another recent approach, ptychographic FROG [12], offers improvement by being able to handle any Fourier relations and partial measurements. Both the PCGPA and the ptychographic FROG can reconstruct ultrashort pulses at fairly low signal-to-noise ratios (SNRs) after manual pre-processing [13]. However, without the filtering stage, their performance deteriorates at low SNR to the extent that FROG traces of too weak ultrashort pulses are non-invertible.

Here, we propose and demonstrate, theoretically and experimentally, the reconstruction of ultrashort optical pulses by employing deep neural networks (DNNs), and show (on simulated data) that our trained network outperforms other state-of-the-art techniques for low SNR measurements. We further develop our methodology by modifying the network training stage to combine both supervised and unsupervised learning, and show that this new network is able to reconstruct ultrashort pulses from low SNR experimental data, while being trained on simulated data.

More specifically, we train a convolutional neural network (CNN) to learn the inverse mapping of the FROG measurement function using supervised learning. In the training procedure, we use a set of pairs of known pulses and their associated FROG traces, where the traces serve as input (to the procedure), and the recovered pulses are compared with the (known) original pulses. In the field of deep learning, this procedure is called

supervised learning using labeled data, and it allows us to blindly learn the structure of the problem without knowing the mathematical relation in advance. As we show in simulations below, this methodology enables the reconstruction of pulses, even from low SNR data. However, transferring this technique to experiments would require training the DNN on tens of thousands of measured pulses, which is clearly a drawback. To rectify this problem, we modify the learning procedure and combine supervised together with unsupervised (unlabeled) training. The procedure facilitates the reconstruction of pulses based on a network that is trained on simulated data. In this modified network, the supervised procedure uses the previous computer simulated data set, which helps the network learn a good basis to represent pulses and improves its ability to filter noise. The unsupervised procedure makes the network more specific in reconstructing the given experimental measurement. The joint training procedure makes the whole concept of DNN pulse reconstruction experimentally viable, by removing the need for measuring thousands of pulses for the network training stage. We also note that the unsupervised procedure could not be carried on its own (without the supervised procedure), as it is highly sensitive to ambiguities. Finally, we use our network and methodology on experimental data and demonstrate high-quality pulse reconstruction with networks trained strictly on simulated data. Our reconstruction outperforms PCGPA and ptychographic FROG in the regime of low SNR, while matching their performance for high SNR data.

Recovering the structure of ultrashort pulses through neural network techniques was pioneered more than two decades ago [14] when a parametric model was trained to reconstruct the pulse from its polarization gated (PG) FROG measurements. However, those attempts struggled to perform well empirically and were not widely adopted, since they used old neural network implementations, small training sets, and a shallow two-layer neural network. Even more importantly, those methods were used after manual engineering of features in the data, instead of being fed with raw measurements, e.g., [14] assumed that the phase follows polynomial dependence and aimed at reconstructing the first polynomial coefficients.

In contrast to those early neural networks, the past decade has witnessed overwhelming progress on deeper and more sophisticated network architectures, those known today as DNN, that are now being extensively used to extract expressive features from data automatically [15]. The massive progress with DNN research has recently led to superhuman performance, meaning that the DNN extracts features better than a human, in a plethora of applications ranging from image classification [16], natural language processing, [17] and Go [18] to visual diagnosis of skin cancer [19], microscopy [20], and estimating the parameters of strong gravitational lensing systems [21], to name a few. We note that deep learning techniques extract the structure of the data by learning from many examples, whereas classical algorithms reconstruct a single pulse from a single measurement. This fact endows DeepFROG with the advantage of filtering noise better [22] than commonly used algorithms.

We focus on second-harmonic generation (SHG) FROG, the widely used FROG system, where spectrograms of autocorrelations of an unknown complex pulse $E(t)$ are measured in order to reconstruct the amplitude and phase structure of the pulse. The measured trace, after being discretized, is given by

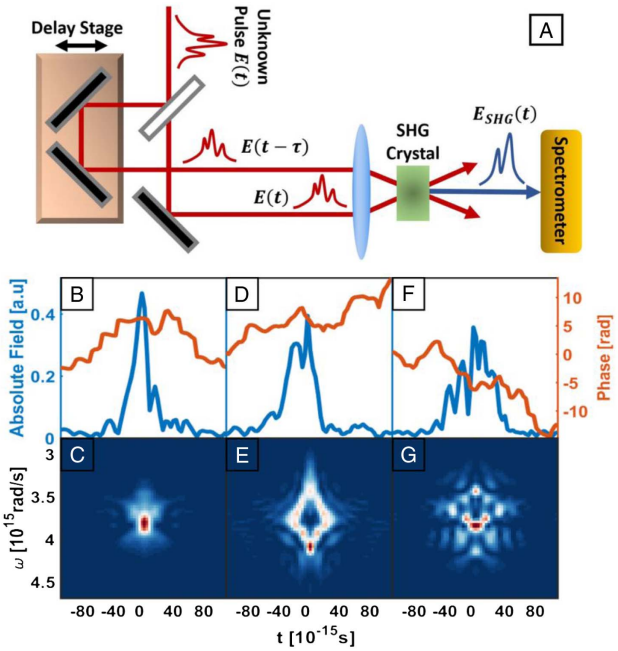

Fig. 1. (A) Experimental SHG FROG setup. (B)–(G) Simulated examples from the data set. (B), (D), (F) are pulse amplitudes (blue) and phases (red). (C), (E), (G) are the corresponding FROG traces.

$$I_{\text{measured}}(\omega_i, \tau_j) = |\mathcal{F}\{E(t)E(t-\tau_j)\}|^2, \tag{1}$$

where $\mathcal{F}$ is the Fourier transform operator, ω_i are the discrete frequencies of the spectrometer, and τ_j are time delays between the pulse $E(t)$ and its replica. The reconstruction problem is defined by mapping the FROG trace to the pulse that created it, $I_{\text{measured}}(\omega_i, \tau_j) \to \tilde{E}(t)$. Figure 1(A) depicts the experimental scheme for generating FROG traces. Figures 1(B)–1(G) show three simulated examples of short pulses and their corresponding FROG traces.

If a SHG FROG trace is invertible, then the reconstructed pulse, $\tilde{E}(t)$, is unique up to trivial ambiguities: constant phase shift, inversion with conjugation, and translation (the proof [8] is based on the fundamental theorem of algebra). This observation is essential, as it assures us that pulses with similar FROG traces are themselves similar (Fig. 2). However, while these ambiguities are considered trivial and often have virtually no impact on the use of ultrashort pulses, training a DNN to overcome one of these ambiguities suffers from instabilities, e.g., when an ambiguity appears in the data set, i.e., two different pulses correspond to the same (or a similar) FROG trace. This problem is highly related to multi-label classification [23], a fundamental problem in machine learning where a data point may be associated with multiple labels instead of a single class. One approach to deal with it is to transform the problem into many single-label problems and solve each one separately. We avoid introducing ambiguities to the training set by generating pulses with the same constant phase shift, inversion with conjugation, and translation.

2. RECONSTRUCTION METHOD

Our algorithm, which we call DeepFROG, consists of two functions that are implemented using differential programming software and represented as DNNs. The first neural network,

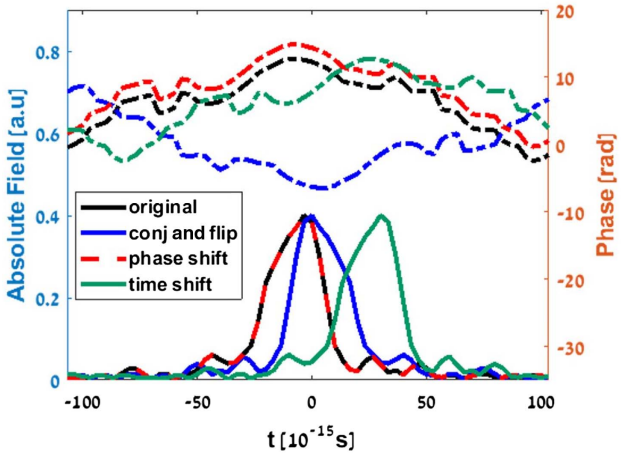

Fig. 2. Four different pulses (amplitude and phase) that correspond to the same FROG trace. Each pulse corresponds to a different ambiguity in SHG FROG measurements.

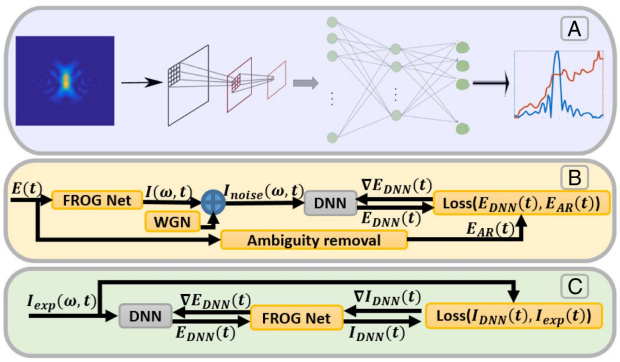

Fig. 3. (A) DNN architecture for mapping FROG traces into complex pulses. A FROG trace, which serves as input data, is convoluted by learned filters in three subsequent layers. The result serves as an input to three fully connected layers with ReLU activation, after which the real and imaginary parts of the pulse are fully reconstructed. (B) Supervised training procedure. The generated pulses are used (after ambiguity removal) as labels for the supervised DNN training step. We create a FROG trace from these pulses with the FROGNet, add white Gaussian noise (WGN), and then forward propagate the pulses in a DNN. The label comparison gradient (loss) back propagates and updates the weights. The gradient is back propagated through the DNN and is added to the weights through a stochastic gradient descent (SGD) update. (C) Unsupervised training procedure. Similar to B, but now, the reconstructed pulse (the output of the DNN) is also forward propagated through the FROGNet, such that the reconstructed FROG trace is compared with the measured one. The gradient is then computed and is back propagated through the FROGNet to update the weights of the DNN.

denoted by FROGNet, represents the measurement system governed by Eq. (1). It uses differential building blocks, and provides efficient evaluations of the measurement system as well as its gradients. For example, Fourier transform is a linear transformation and can be represented using a fully connected layer whose weights are the Fourier matrix. In this vein, time shifts, multiplication, and Euclidean norms can also be represented by differential building blocks, and they all have implementations in Torch [24], the deep learning library we use here, as well as in other libraries. We construct the FROGNet by using these building blocks and adding them sequentially one after the other (e.g., norm after Fourier), in parallel (e.g., applying many parallel time shifts to the pulse), and combining them (e.g., summing the imaginary and real parts of the complex Fourier transform). We emphasize that this function is a-parametric and remains constant, i.e., it is not learned or changed at any time, and it represents the measurement system exactly through the entire learning process. The second function is a parametric CNN, which receives a FROG trace and outputs a vector of the real and imaginary parts of the reconstructed pulse. The weights of the CNN function are optimized such that the function is close to the inverse mapping of the FROGNet function. This is achieved using Adam [25], a variant of stochastic gradient descent (SGD), by minimizing the l_1 loss between the measurement and the reconstruction:

$$w^* = \arg\min_w \{\text{loss}(I, E)\}$$
$$= \arg\min_w \left\{ \begin{array}{l} \|\text{CNN}(I; w) - E\|_1 \\ +\lambda \|\text{FROGNet}(\text{CNN}(I; w)) - I\|_1 \end{array} \right\}, \quad (2)$$

where $\text{CNN}(I; w)$ denotes the output of the CNN function with weights w, given the input I and label E, and λ is the supervised to unsupervised learning rates ratio [26–28]. In principle, λ may vary between different datasets. In Section 3, we show how this λ parameter can be exploited to reconstruct experimental pulses by DNNs that were trained with computer simulated pulses, where $\lambda \neq 0$. For $\lambda = 0$, the network will learn only from supervised training.

Our DNN architecture is a CNN [29], which commonly consists of a few convolution layers followed by fully connected neuron layers with nonlinear activations in between (Fig. 3(A)). Given a layer y_{in} in the CNN, which consists of M matrices, also called M channels, of size $N_{\text{in},x} \times N_{\text{in},y}$, a convolution layer with $P \cdot M$ filters w of size $N_{w,x} \times N_{w,y}$ will output P matrices of size $N_{\text{out},x} \times N_{\text{out},y}$, denoted by y_{out}. These matrices are used as the next CNN layer. Mathematically, the convolution for each pixel in the pth matrix in the output layer $(n_{\text{out},x}, n_{\text{out},y})$ is described by summing all the multiplications between input pixels of the mth matrix $(n_{\text{in},x}, n_{\text{in},y})$ and the pth filter pixels $(n_{w,x}, n_{w,y})$ for the mth input matrix:

$$y_{\text{out}}(p; n_{\text{out},x}, n_{\text{out},y}) = \sum_{\substack{n_{w,x} \\ n_{w,y} \\ m}} y_{\text{in}}(m; n_{\text{out},x} - n_{w,x}, n_{\text{out},y} - n_{w,y}) w(p, m; n_{w,x}, n_{w,y}). \quad (3)$$

In addition, a convolutional layer may have a padding parameter p (additional zeros added to the input) and a stride parameter s. For stride s we move the filters by s pixels at a time. Given these parameters, the size of the output image is given by the size of the input image and the filter parameters so that (for one of the dimensions):

$$n_{\text{out}} = (n_{\text{in}} + 2p)/s + (n_w - 1). \quad (4)$$

During training, the convolutional layers automatically extract 2D shift invariant features from the data, by learning the weights of the filters w.

After the last convolutional layer is applied, the information in the different feature maps is reshaped into a single, one-dimensional vector, followed by fully connected (linear) layers. The output of this layer is also a vector, $y_{\text{out}} = W y_{\text{in}}$, where

W is a matrix (the weights of the layer), and y_{in} is the input vector. The parameters of the fully connected layers are updated during training, similar to the filters of the convolutional layers. First, the gradients are computed using backpropagation [30]. Then, the gradients are scaled and added to the weights via a SGD algorithm. SGD typically maintains a single learning rate for all weight updates, and the learning rate does not change during training. In practice, we used Adam [25], a variant of SGD that has individual adaptive learning rates for different parameters, calculated from estimates of first and second moments of the gradients (see Ref. [25] for more details).

We test three different architectures. The first type is a simple CNN with three convolutional layers and two fully connected layers with a basic neuron function, the rectified linear unit (ReLU, $\mathrm{ReLU}(x) = \mathrm{Max}(x, 0)$), in between. The convolution layers have 32 filters each, kernel sizes of 4,2,1, respectively, strides 2,2,1, and pad 1,1,1. The convolutions are followed by two fully connected layers, where the size of the last hidden layer is 512, and the size of the last layer is 128 (corresponding to the imaginary and real parts of the pulse).

The second type is inspired by Ref. [31] and uses convolutional blocks with filters at multiple resolutions (sizes) in each convolutional block, and is denoted by Multires. Explicitly, this network is composed of six convolutional layers and two fully connected layers, all followed by ReLU activations. The convolution layers are divided between Multires layers and regular convolution layers. The Multires layers contain filters with four different sizes, e.g., 11,7,5,3, where the stride and padding parameters are adjusted such that the output images from each filter are of the same length as the input image [using Eq. (4)]. Thus, the output of a Multires convolution layer is of the same size as the input, but has four times more channels (due to the multiple filters). Each Multires layer is followed by a standard convolution layer, adjusted such that the output image is half the size of the input image with twice as many filters. Finally, the fully connected layers size is adjusted such that the size of the last hidden layer is 512, and the size of the last layer is 128 (similar to the CNN).

The third type is the densely connected convolutional network (DenseNet) [32] architecture, a very deep network with high interconnectivity between nonadjacent layers. Overall, DenseNet and our Multires architecture display the best performance, while Multires takes a shorter time to reach good results. Here, we report only results for the Multires architecture and leave architecture comparison for a future comparative study.

3. RESULTS

A. Simulations

We create our simulated training data in the following manner. To design simulated pulses, we first generate a random spectral phase $\phi(\omega)$ by smoothing a randomly generated vector and add it to a Gaussian power spectrum $S(\omega)$, such that $\tilde{E}(\omega) = \sqrt{S(\omega)}e^{i\phi(\omega)}$, where $\tilde{E}(\omega) = \mathcal{F}[E(t)]$. We take a wide enough spectrum $S(\omega)$, and we smoothen the phase $\phi(\omega)$ by reducing the high frequency Fourier coefficients of the random vector to ensure that the pulse $E(t)$ is limited in time while still having small fast features. After generating $\tilde{E}(\omega)$, we inverse Fourier transform it into a time-domain electromagnetic field $E(t)$. The widths of our simulated pulses are of 100 fs, with 64 (N) delay points. Then, we forward propagate $E(t)$ in the FROGNet to yield a FROG trace. To deal with ambiguities whose removal is essential for DNNs, we choose a single pulse $E_{AR}(t)$ out of its "trivial" $E(t)$ counterparts, which correspond to the same FROG trace. Specifically, $E_{AR}(t)$ must have an intensity peak at a specific time; its phase at the peak is zero, and the integral of the left (to the peak) side of the real part of $E_{AR}(t)$ is larger than its right side integral. Examples of generated pulses and their FROG traces are given in Fig. 1. We collect 60 K training examples containing pairs of inputs (FROG traces, I) and labels (pulses, E_{AR}). These pairs are used to calculate the reconstruction error and s weights by back propagating the error gradient back to the DNN. The entire learning process is described in detail in Fig. 3(B). Once the training is complete, we use the trained network to test 10 K simulated examples that were not part of the training set.

We now present results showing the reconstruction of computer-simulated pulses using a DNN that was trained with simulated data. We train two variants of the DeepFROG algorithm, one without injected noise in the traces during training, and the second with injected noise (WGN), sampled uniformly to have SNR values of 0–30 dB. Every FROG trace in the training data is injected into the DNN, the output of the DNN is compared to the matching pulse, and the resulting error is back propagated to update the weights in the network. This process is repeated for every datum a couple of hundreds of times. The total training time of the network is between 15 and 45 min on a standard PC with a Titan X GPU. Figure 4 presents the final performance of the different methods, tested on simulated pulses (which were not part of the training set) with varying levels of noise. The DeepFROG variant that was trained with noise achieves lower reconstruction error, calculated by $\delta = \|E_r - E_i\|_1$, than classical methods for SNR values below 22 dB. Figure 5 presents an example from the simulated data set, along with its reconstructions. The DeepFROG algorithm produces less noisy pulses with lower reconstruction error compared to previous methods. Overall, injecting noise during training helps the network to learn how to filter noise from noisy unseen examples. Also, after training, it takes only 0.7 ms to recover a pulse from an SHG FROG trace.

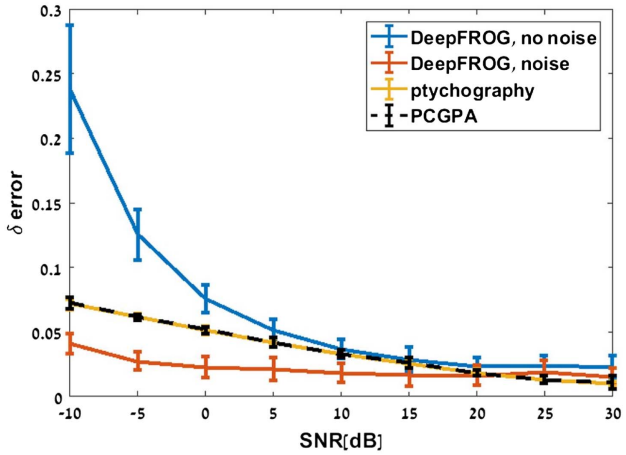

Fig. 4. Learning stage. Reconstruction error and STD as a function of SNR for different methods: DeepFROG DNNs trained with no noise (blue) or 0–30 dB SNR (red) training sets, ptychographic FROG (yellow), and PCGPA (dashed black). DeepFROG trained with noisy data reaches the lowest error at SNR values below 20 dB.

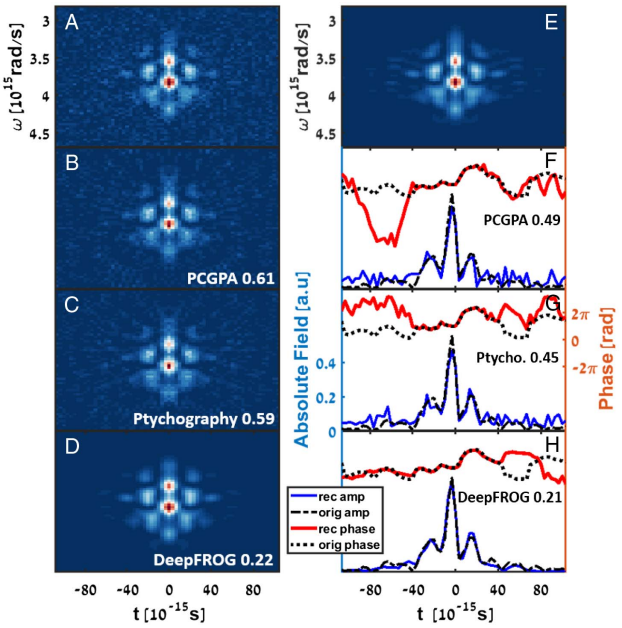

Fig. 5. Reconstruction of a simulated example pulse, which was not included in the training set, from its FROG trace with 10 dB SNR, using PCGPA, ptychography, and DeepFROG. (A) Noisy simulated FROG measurement at 10 dB SNR. (B), (C), (D) are the reconstructed FROG traces by PCGPA, ptychographic FROG, and DeepFROG, respectively. (E) FROG trace of the original simulated test pulse. (F), (G), (H) reconstructed pulses by PCGPA, ptychographic FROG, and DeepFROG, respectively, compared to the original. Pulse error and FROG trace error are denoted. DeepFROG reaches the lowest reconstruction error, almost three times lower than the error achieved by other algorithms.

Figure 4 shows, in simulations, that the DeepFROG DNN trained with some added noise outperforms both ptychographic FROG and PCGPA for SNR lower than 20 dB. Based on these simulations, we believe that training the DNN on experimentally measured data will facilitate similar results on experiments. However, training the DNN requires some 60,000 pulses, which clearly sets an experimental toll that would pose a clear disadvantage.

The challenge is therefore to create a DNN to be trained by simulated data, and would still be able to handle experimentally measured data, even at low SNR. To deal with this issue, we created the second unsupervised procedure (Fig. 3(C)), in which we impose the mathematical relation between the pulse and the FROG trace to train the network to reconstruct experimental pulses.

B. Reconstructing Ultrashort Pulses from Experimentally Measured Data with Networks Trained by Simulated Data

In this section, we demonstrate that a DNN with the same architecture, trained on simulated data, but, in parallel, using the unsupervised loss for the experimental pulse, is able to accurately reconstruct ultrashort pulses from their experimentally measured FROG traces, even at low SNR. In fact, as we show below, our network outperforms other state-of-the-art techniques for low SNR data. To test this, we turn to reconstruct ultrashort laser pulses measured in our lab. We use a mode-locked Ti-Sapphire laser to generate the pulse and shape it with a spatial

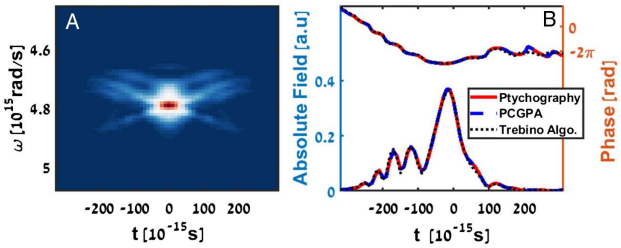

Fig. 6. Reconstruction of an experimentally generated reference pulse. (A) High SNR SHG FROG measurement of the reference pulse. (B) Reconstruction of the pulse from (A) with the commonly used algorithms: ptychographic FROG, PCGPA, and the algorithm provided within the commercial reconstruction FROG program by the Trebino group.

light modulator (SLM). We measure the pulse using an SHG FROG setup.

We evaluate the performance of our method by calculating the error between the FROG trace of a reference pulse and the FROG trace of the reconstructed pulse, and compare to the performance of other state-of-the-art methods. The reconstruction error is defined based on the one-to-one mapping between a pulse and its FROG trace (up to trivial ambiguities, see Ref. [8]). The error between two FROG traces is defined by $\delta_I = \|I_r - I_i\|_1 / \|I_i\|_1$, where I_i is the measured FROG trace intensity, and I_r is the reconstructed FROG trace intensity. As the reference pulse, we use a

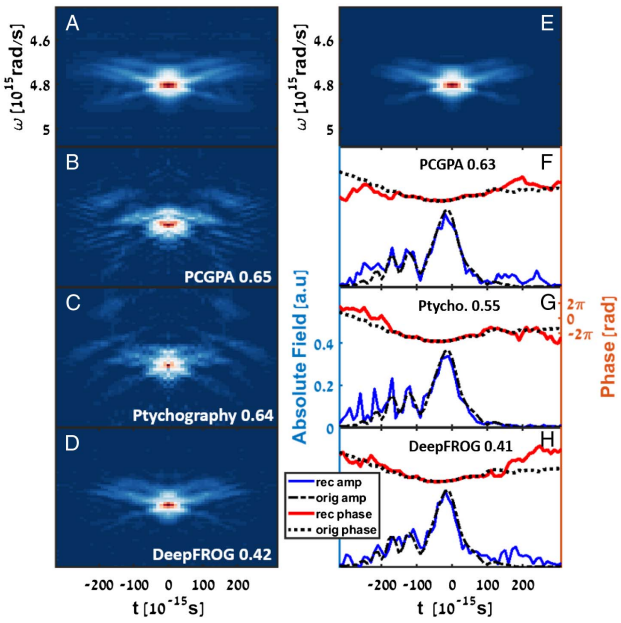

Fig. 7. Reconstruction of an experimental pulse from low SNR FROG measurements. (A) Measured FROG trace. (B), (C), (D) FROG traces constructed from the pulses reconstructed by PCGPA, ptychographic FROG, and DeepFROG, with their error compared to the FROG trace of the reference pulse listed in white. (E) FROG trace of the reference pulse. (F), (G), (H) Corresponding reconstructed pulses and their reconstruction errors. Additionally, the δ_I errors between the reconstructed FROG traces and the low SNR measured trace are 0.49, 0.44, and 0.32 for PCGPA, Ptychography, and DeepFROG, respectively (not shown in figure). The modified DeepFROG displays the lowest reconstruction error for both the pulse and the FROG trace.

pulse measured at high SNR, where all the methods yield virtually identical reconstructions. Then, we use the same pulse measured at low SNR as the test pulse, and as we show below, our combined training methodology outperforms the other techniques.

To make our method able to reconstruct pulses from experimentally measured data based on networks trained by simulated data, we make use of a combination of supervised learning on the computer-simulated data (Fig. 3(B)), along with unsupervised learning (Fig. 3(C)) on the measurement [λ in Eq. (2) was set to 0.1 for the simulated pulses and to 0.8 for the measured pulse]. That is, we are using the gradients from the FROGNet to update the trained DNN according to the experimentally obtained FROG trace. The unsupervised learning part allows us to use the structure of the problem, benefiting from the gradients of the FROGNet, to find the solution specific to the experiment, without any prior knowledge on the reconstructed pulse.

To compare our results to a known reference pulse, we measure two FROG traces from the same pulse, one measured at a higher SNR than the other, to be used as a reference for the noisier case. To reconstruct the reference pulse in the case of the high SNR, we use three methods: PCGPA, ptychographic FROG, and the Trebino group software [33] for FROG pulse reconstruction. Because of the high SNR, we can use the error between the measured and reconstructed FROG traces as a recovery quality criterion. The results are shown in Fig. 6. The reconstructions are similar, with δ_I-errors of 0.1. We chose (arbitrarily) to use the pulse recovered by the Trebino group software as a reference.

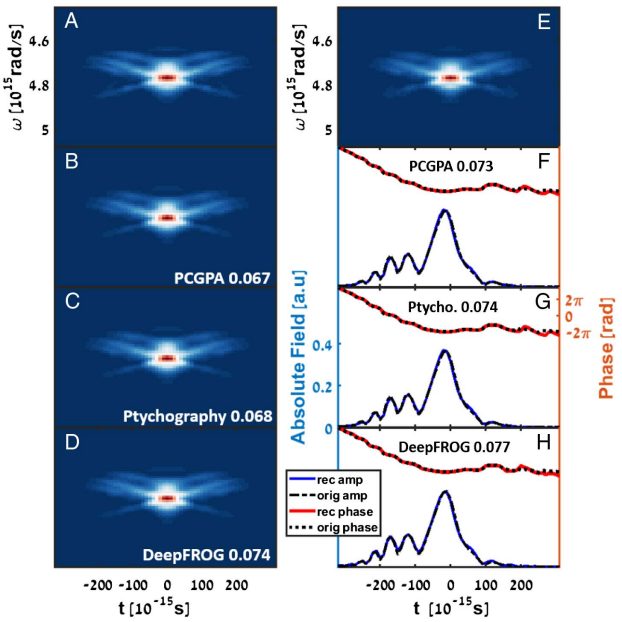

Fig. 8. Experimental pulse reconstruction from noisy but filtered FROG measurement. (A) Measured and filtered FROG trace. (B), (C), (D) are FROG traces constructed from the retrieved pulses reconstructed by PCGPA, ptychographic FROG, and DeepFROG and their errors from the reference FROG trace, respectively. (E) FROG trace constructed from the reference pulse. (F), (G), (H) are the corresponding pulses and pulse reconstruction errors δ_E, relative to the reference pulse. Additionally, the δ_I errors between the reconstructed FROG traces and the low SNR measured trace are 0.102, 0.103, and 0.092 for PCGPA, Ptychography, and DeepFROG, respectively (not shown in figure). In this filtered trace reconstruction, DeepFROG performs comparatively to PCGPA and ptychography—results we predicted in the simulations section.

Then, we turn to reconstruct pulses from their low SNR experimentally measured FROG traces. We compare among three reconstruction methods—PCGPA and ptychographic FROG, both state-of-the-art methods, with our DeepFROG reconstruction. Figure 7 presents the reconstructed experimental pulses and their FROG traces for the ptychographic FROG, PCGPA, and the deep learning methods, on top of the chosen reference pulse. It is possible to see, even with a naked eye, that all pulse reconstructions differ from the original pulse, but the deep learning method has reached the lowest δ_I-error when comparing the reconstructed FROG trace to the reference FROG trace. Clearly, the error ($\delta_E = \arccos(\langle E_r|E_i\rangle/\sqrt{\langle E_r|E_r\rangle\langle E_i|E_i\rangle})$, as in Ref. [12]) between the reference pulse E_i and the recovered one E_r is the lowest for the DeepFROG method. In Fig. 8, we show the pulse recovery from the same low SNR measured FROG trace, but after using noise filtering on the trace. Here, both PCGPA and ptychography achieve a slightly better reconstruction than the proposed DeepFROG algorithm, when comparing the reconstructed FROG traces to the high SNR measured FROG trace.

To summarize this section, we find that the DeepFROG is able to handle low SNR data and perform almost as the PCGPA and ptychography do at high SNR. These experimental results match the performance we expected for low and high SNR values in the training stage, thus confirming the high potential of using DNNs for pulse reconstruction.

4. DISCUSSION

The proposed DeepFROG method for reconstructing ultrashort pulses from FROG measurements consists of two training procedures. The first one involves training a DNN to reconstruct pulses from FROG measurements using supervised learning on a synthetic dataset, i.e., pairs of pulses and FROG measurements that are created for the purpose of training. In the second method, we modify the network to be able to reconstruct an experimental pulse after being trained by simulated data. To do that, we add an unsupervised component in the training stage, where the learning uses a large computer-generated data set and the specific experimentally measured FROG trace (with no additional knowledge on the pulse or on the experiment).

For the first method, which uses solely computer-generated pulses for training, the DNN outperforms the other methods for any SNR below a reasonably high SNR of 20 dB on test pulses (pulses that were not used by the network for raining). This implies that a network that was trained on experimentally measured data should be able to reconstruct pulses from measurements of their FROG traces. However, this would require tens of thousands of measured pulses in the training stage. This raises the question of whether a network trained solely with simulated data would be able to reconstruct pulses from real measurements. We tested this with an experimentally measured FROG trace via our DNN trained with simulated data (first method), and the network managed to reconstruct the pulse but was outperformed by other state-of-the-art methods. This challenge, known in the deep learning community as sim-to-real, is due to the fact that the simulated data are limited to our generation methods and do not represent perfectly all the possible pulses in the world, and in particular, those that we measure in the lab.

To solve this issue, we created the second method, which combines the supervised learning from simulated data with an unsupervised procedure, where we imposed the mathematical

relation between the pulse and the FROG trace. In doing that, we trained the network to reconstruct the experimental pulse without knowing anything on the actual pulse. As we show in Figs. 7 and 8, the network that was trained using a combination of supervised and unsupervised training procedures provided better reconstruction on experimental data than other algorithms, when the noise level was high, and was slightly below them at high SNR. The advantage of using both procedures together is that the supervised training procedure trains the network to filter the noise and deal with ambiguities, while the unsupervised procedure trains the network to reconstruct the measured pulse. We note that the unsupervised procedure did not perform well without the supervised one, because the ambiguities caused the DNN to diverge, a phenomenon we observed during training.

To further enhance the performance of our approach, we plan to investigate the sim-to-real challenges in future work. First is increasing the variety of the computer-generated dataset to include as many spectral amplitudes and phases as possible. The second is to significantly enlarge the number of measured pulses that train the network. Of course, this suggestion has obvious disadvantages, but in some experimental schemes, where the measurements are embedded in noise, or when extreme accuracies are crucial, this could be practical. The third is using generative models to generate more data by learning the data distribution of measured pulses. In particular, a recently developed network called the generative adversarial network (GAN) [34] can be used to create new data pulses on which the DNN tends to make mistakes (poorly reconstruct the pulses). These pulses will be new to the dataset on purpose, and will increase the variety of the pulses in the training dataset.

5. CONCLUSIONS

In this work, we presented a deep learning approach to reconstruct ultrashort laser pulses from their measured FROG traces. We constructed a supervised DNN with convolutional and fully connected layers and trained it by injecting labeled data examples and updating the network using the reconstruction error. By doing this, we use convolutions not only to construct the FROG trace, but also to reconstruct the pulse from it. We point out that in the supervised training procedure, our method does not require any advance knowledge of the relation between the pulse and the measurement (up to trivial ambiguities). That is, by blindly learning the structure of the problem, our network enables the reconstruction of pulses from measured data, even when the mathematical relation between the measurement and the pulse is unknown. We have shown that the DeepFROG network is able to recover the pulses from measurements even when the pulses are weak and embedded in noise, and that the DeepFROG is superior to other methods by reducing reconstruction error, in particular with high noise levels. The ability to compete with state-of-the-art methods and outperform them is a clear proof that there is a fundamental difference between our DNN-based DeepFROG and the machine learning method suggested two decades ago in Ref. [14]. Finally, we implemented an unsupervised network by using supervised DNN followed by the FROGNet to recover an ultrashort laser pulse from experimental data based on a training stage performed on simulated data. We believe that application of DNN can lead to improved performances in many techniques in diagnostics of ultrashort laser pulses, including d-scan, multiplexed FROG [35], and interferometric FROG [36]. In a broader context, what we have shown here is a method for phase retrieval that does not require any advance information: neither the support, nor sparsity, or a desirable mathematical basis to represent the data [37]. It does not even require any knowledge of the mathematical relation between the measured data and the information we wish to reconstruct. Clearly, this work indicates that DNNs can become a powerful approach in diagnostics of ultrashort laser pulses.

Funding. Israel Science Foundation (ISF); Wolfson Foundation; Azrieli Foundation; Israeli Centers for Research Excellence (I-CORE).

Acknowledgment. Tom Zahavy has received funding from the Azrieli Fellows Program. Daniel Moss has received funding from the Joan and Reginald Coleman-Cohen Foundation. This work is part of the Israeli Centers for Research Excellence (I-CORE), 'Circle-of-Light'.

[†]These authors contributed equally to this work.

REFERENCES

1. A. M. Weiner, *Ultrafast Optics* (Wiley, 2009).
2. M. F. Kling and M. J. J. Vrakking, "Attosecond electron dynamics," Annu. Rev. Phys. Chem. **59**, 463–492 (2008).
3. Y. Silberberg, "Quantum coherent control for nonlinear spectroscopy and microscopy," Ann. Rev. Phys. Chem. **60**, 277–292 (2009).
4. F. Lépine, M. Y. Ivanov, and M. J. J. Vrakking, "Attosecond molecular dynamics: fact or fiction?" Nat. Photonics **8**, 195–204 (2014).
5. H. Mikami, L. Gao, and K. Goda, "Ultrafast optical imaging technology: principles and applications of emerging methods," Nanophotonics **5**, 441–453 (2016).
6. R. Trebino, K. W. DeLong, D. N. Fittinghoff, J. N. Sweetser, B. A. Richman, D. J. Kane, and M. A. Krumbügel, "Measuring ultrashort laser pulses in the time-frequency domain using frequency-resolved optical gating," Rev. Sci. Instrum. **68**, 3277–3295 (1997).
7. R. Trebino, *Frequency-Resolved Optical Gating: The Measurement of Ultrashort Laser Pulses* (Springer, 2012).
8. T. Bendory, P. Sidorenko, and Y. C. Eldar, "On the uniqueness of FROG methods," IEEE Signal Process. Lett. **24**, 722–726 (2017).
9. P. O'shea, M. Kimmel, X. Gu, and R. Trebino, "Highly simplified device for ultrashort-pulse measurement," Opt. Lett. **26**, 932–934 (2001).
10. X. Gu, S. Akturk, A. Shreenath, Q. Cao, and R. Trebino, "The measurement of ultrashort light—simple devices, complex pulses," in *Femtosecond Laser Spectroscopy* (Springer, 2005), pp. 61–86.
11. D. J. Kane, "Principal components generalized projections: a review [Invited]," J. Opt. Soc. Am. B **25**, A120–A132 (2008).
12. 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).
13. D. N. Fittinghoff, K. W. DeLong, R. Trebino, and C. L. Ladera, "Noise sensitivity in frequency-resolved optical-gating measurements of ultrashort pulses," J. Opt. Soc. Am. B **12**, 1955–1967 (1995).
14. M. A. Krumbügel, C. L. Ladera, K. W. DeLong, D. N. Fittinghoff, J. N. Sweetser, and R. Trebino, "Direct ultrashort-pulse intensity and phase retrieval by frequency-resolved optical gating and a computational neural network," Opt. Lett. **21**, 143–145 (1996).
15. M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision - ECCV 2014*, Lecture Notes in Computer Science (Springer, 2014), Vol. **8689**, pp. 818–833.
16. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (NIPS, 2012), pp. 1–9.
17. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems* (NIPS, 2014), pp. 3104–3112.

18. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," Nature **529**, 484–489 (2016).
19. A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," Nature **542**, 115–118 (2017).
20. Y. Rivenson, Z. Göröcs, H. Günaydin, Y. Zhang, H. Wang, and A. Ozcan, "Deep learning microscopy," Optica **4**, 1437–1443 (2017).
21. Y. D. Hezaveh, L. P. Levasseur, and P. J. Marshall, "Fast automated analysis of strong gravitational lenses with convolutional neural networks," Nature **548**, 555–557 (2017).
22. P. Vincent and H. Larochelle, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion Pierre-Antoine Manzagol," J. Mach. Learn. Res. **11**, 3371–3408 (2010).
23. J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," Mach. Learn. **85**, 333–359 (2011).
24. R. Collobert, S. Bengio, and J. Mariethoz, "Torch: a modular machine learning software library," IDIAP-RR-46-2002 (2002).
25. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv:1412.6980 (2015).
26. C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer-Verlag, 2006), Vol. **4**.
27. D. Erhan, A. Courville, and P. Vincent, "Why does unsupervised pre-training help deep learning ?" J. Mach. Learn. Res. **11**, 625–660 (2010).
28. Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, "Building high-level features using large scale unsupervised learning," in *29th International Conference on Machine Learning* (2012), p. 38115.
29. L. D. Le Cun Jackel, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, B. Le Cun, J. Denker, and D. Henderson, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems* (NIPS, 1990), pp. 396–404.
30. Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**, 436–444 (2015).
31. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, C. Hill, and A. Arbor, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 1–9.
32. G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," arXiv:1608.06993 (2017).
33. "Code for retrieving a pulse intensity and phase from its FROG trace," 2013, http://frog.gatech.edu/code.html.
34. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (NIPS, 2014), Vol. **27**, pp. 2672–2680.
35. G. I. Haham, P. Sidorenko, O. Lahav, and O. Cohen, "Multiplexed FROG," Opt. Express **25**, 33007–33017 (2017).
36. G. Stibenz and G. Steinmeyer, "Interferometric frequency-resolved optical gating," Opt. Express **13**, 2617–2626 (2005).
37. Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, "Phase retrieval with application to optical imaging: a contemporary overview," IEEE Signal Process. Mag. **32**(3), 87–109 (2015).