The creation and reconstruction of ultrashort (femtosecond or attosecond) laser pulses is an important field of basic research. However, a large range of applications also use such pulses (femtochemistry or ultrafast imaging for example), but all are based on physical processes, which cannot be resolved in space or time using other means of investigation. Since these pulses cannot be measured by conventional electronics, their characterization involves special techniques, one of them being FROG [1] (frequency resolved optical gating), which uses the pulse and a delayed replica of itself to create a spectrogram. The reconstruction method means the recovery of the amplitude and phase of the pulse from the FROG spectrogram. Here, inspired by recent studies [2,3], we employed a CNN (convolutional neural network) to recover the amplitude and phase of ultrashort pulses.

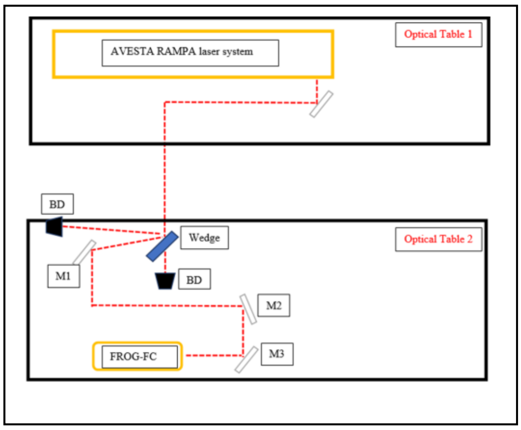


Fig. 1: The top view of the experimental layout used in the measurements. BD=beam dump, M1, M1, M2, M3, flat mirrors, AVESTA RAMPA is the laser system and the FROG-FC is the measurement device used.

Dataset generation

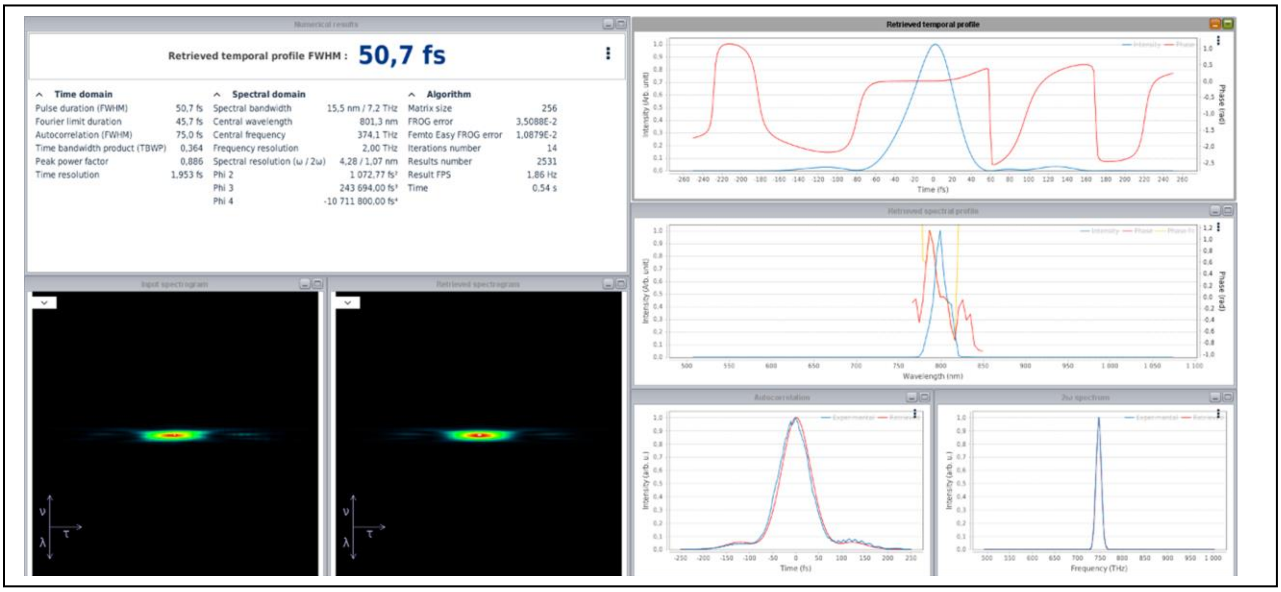
a. Experimental FROG Traces

FROG traces were acquired using the ultrashort laser pulses delivered by the AVESTA system delivering down to 40fs pulses duration with up to 1mJ energy at λ =800nm central wavelength [4]. The spectral phase and pulse duration of the laser pulses were measured by employing a FROG device. The experimental set-up is presented in Fig. 1.

The output pulses are attenuated in order to achieve an energy no higher than 100 µJ. The beam diameter has around 8 mm after the compressor. By adjusting several parameters of the temporal compressor, such as the distance and angle of the grating as well as cutting the 3rd pass spectrum in various positions and with different slit widths, the FROG traces are recorded and saved for post-analysis in order to train the artificial neural network.

The optical stretcher of the AVESTA laser system provides a stretching factor of 5.94 ps/nm. The inter-grating distance is 550.00 mm for an initial 40nm bandwidth pulse stretched to 245ps. The temporal compressor that compensates for the group delay dispersion generated by the stretcher and the optical materials along the amplification chain has similar distance between the gratings. As a consequence, 1 mm variation of the inter-gratings distance in the compressor corresponds to roughly 440fs additional pulse duration increase. The adjustment of the distance can be performed using a micrometer screw.

It can be shown that the second order Taylor series coefficient for the spectral phase decomposition in the case of a double pass compressor has the form φ_2=c L N^2 Sec(β)^{2}λ^{-3}, with the L the distance between the gratings along the normal direction, N the grooves density, c speed of light and β the diffraction angle at the central wavelength. In our case, this corresponds to: φ_2=3.54923*10^6 fs^2 Sec(β)^2 L/m, where L is expressed in meter (m) when N=1400lines/mm. So, 0.1% of the normal



Retrieved temporal profile FWHM : 50,7 fs

Time domain		Spectral domain		Algorithm	
Pulse duration (FWHM)	50,7 fs	Spectral bandwidth	15,5 nm / 7,2 THz	Matrix size	256
Fourier limit duration	45,7 fs	Central wavelength	801,3 nm	FROG error	3,5088E-2
Autocorrelation (FWHM)	75,0 fs	Central frequency	374,1 THz	Femto Easy FROG error	1,0879E-2
Time bandwidth product (TBWP)	0,364	Frequency resolution	2,00 THz	Iterations number	14
Peak power factor	0,886	Spectral resolution (ω / 2ω)	4,28 / 1,07 nm	Results number	2531
Time resolution	1,953 fs	Phi 2	1 072,77 fs²	Result FPS	1,86 Hz
		Phi 3	243 694,00 fs³	Time	0,54 s
		Phi 4	-10 711 800,00 fs⁴		

Fig. 2 Example of FROG trace measured and reconstructed using the FROG-FC device and associated Fast FROG Retrieval Software from Femto Easy.

compressor length variation corresponds to 3549 fs^2 $Sec(\beta)^2$ additional phase variation.

In this way the ranges of second phase in reach with the help of 0.1% compressor length tuning can be determined to be +/- 3500fs^2. The value is beyond the measurement range of the FROG device used in our case, which is limited to a maximum of 200fs pulse duration measurements. This corresponds to a second order spectral phase smaller than 2500fs^2 for a 40fs Fourier limited pulse duration. Example of experimental FROG trace and reconstructed pulse shape is shown in Fig. 2.

b. Simulated FROG traces

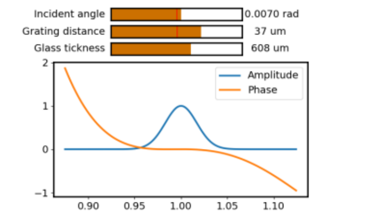


Fig. 3 Snapshot of the program used to analyze the parameter influence on the phase and amplitude.

Experimentally it is time and resource consuming to generate a large set of FROG traces (~10^4), therefore a computational model is used to generate simulated data sets. The code for the generation of datasets is written in Fortran90. We generate datasets by two methods, but we present here the results obtained with the first method only. The second method (which generates the phase randomly) was very recently developed and some preliminary results are under evaluation. The first method incorporates some parameters of the chirped pulse amplification system at the ELI-RO facility, which are chosen to be in an experimentally viable range and determine the phase of the laser pulse. Specifically, for different grating distances and incidence angles of the compressor, the higher order terms in the Taylor expansion of the spectral phase are determined and consequently the phase itself. We also take into account the variation of the width of a glass piece introduced in the laser beam path. The spectral phase and the spectral amplitude (the width for the spectral amplitude ranges between 20 and 50 nm) define the pulse in frequency domain, while its inverse Fourier transform gives the field in time domain, which is in turn used to generate the FROG trace. The wavelength associated with the central frequency is 800 nm. The Fortran code outputs the FROG trace and the spectral amplitude and phase, necessary to train the CNN model. The amplitudes or the phases are used as labels in the training of the CNN. The generated dataset is splitted into training, validation and test sets (70%, 15%, 15%, respectively). The test set contains data samples not used in the training or validation process.

In order to visually assess the behavior of the amplitude and the phase, we have written a Python program, which plots these quantities as a function of the parameters mentioned above. These parameters may be continuously varied through sliders in the program, as shown in Fig. 3. This program showed us that the phase varies on a wide range both in shape and value, being especially sensitive to the variation of the incidence angle. Changing the grating distance has moderate effect, while the thickness of the glass piece proved to have a minor effect on the phase.

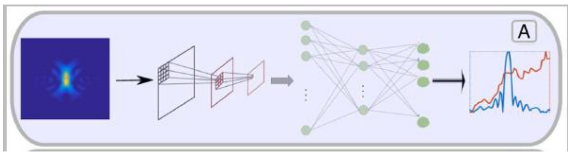


Fig. 4. Schematic representation of the recovery process of the amplitude and phase of a pulse. A FROG image is sent through the layers of the trained CNN and the output is the phase or the amplitude. Picture from [2]

Recovery process using CNN

The input to the CNN is a FROG trace along with the corresponding spectral amplitude or phase (the training data) and the output is the recovered amplitude or phase, generated by the CNN. A schematic representation of the process is in Fig. 4. The goal is to train the CNN (adjusting its parameters) in order to generate a correct amplitude or phase for pulses not used in the training process. These pairs of data (FROG traces and the corresponding amplitude or phase) are fed randomly to the CNN model and in mini-batches (we used 32 or 64 items per batch) to speed up the computations. The CNN is trained typically for a few hundreds or thousands of epochs (iterations) and the output of the CNN is compared to the input amplitude or phase through a loss function (MSE – mean square error or L1 loss) or training error (Fig. 5). The goal is to obtain a minimized loss function, therefore a good model (with adjusted parameters or weights) to reconstruct ultrashort laser pulses. The adjustment of the parameters is achieved through the back propagation method, where the

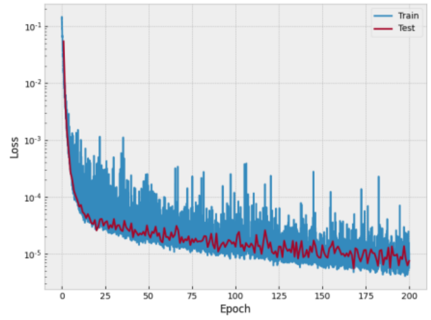


Fig. 5. Typical training and validation losses

gradient of the loss function is calculated with respect to the weights of the model and propagated backwards through the layers of the CNN model. The weights are adjusted according to the value of the gradient; practically the new weights are obtained by adding or subtracting the gradient from the old weight values, multiplied by the learning rate. In practice, we use a variant of the stochastic gradient descent algorithm, called Adam, which is an adaptive learning rate optimization algorithm to perform the back propagation and the weight adjustment. The learning rate is the step at which the adjustments of the weights are performed. We also calculate the validation loss, which is an error function obtained for data samples not used during the learning phase, in order to assess the validity of our model. This is helpful also to spot overfitting the data, which is often encountered during the training of neural networks.

CNN model architecture

Several CNN architectures were built during this phase of the project; however here we present only the most used one. This is a Densenet-BC [5] architecture, which was modified by us to meet the goal of the project. A Densenet architecture refers to a densely connected layer structure of a dense block as depicted in the Fig. 6, where each layer is connected to all subsequent layers in the block. This connection pattern improves the information flow between layers. A layer can output several feature maps, that is a version of the input image processed by some operation like convolution for example, where the original input is convoluted through a kernel, specialized to extract some features. In a single dense block the nth layer receives all the feature maps of the preceding layers by concatenating them into a single tensor. The number of feature maps entering the nth layer is g_0 + $g(n\text{-}1)$, and is controlled by the g growth rate, which is the number of feature maps outputted by a single layer. Here g_0 is the number of channels in the input layer. The number of input feature maps can be large for most of the layers, therefore a bottleneck layer (essentially a 1x1 convolutional layer) is introduced before the concatenation to reduce the number of input feature maps, which aids computational effectiveness. The B in the Densenet-BC refers to this bottleneck layer. The concatenation operation itself is usually a composite operation of convolution and batch normalization operations. However, the concatenation requires that the feature maps have the same size. Therefore in a single block all the feature maps have the same size, and the down-sampling is performed by transition layers, which connect several dense blocks. The transition layers are composed of several sublayers: batch normalization, convolutional and averaging. The averaging layer is responsible for down-sampling the feature maps. The number of feature maps can further be reduced at transition layers to improve model compactness. If a dense block has m feature maps the transition layer will generate $c{*}m$ feature maps, where $0 < c < 1$. This compression procedure is referred to in the model name by the letter C. If both bottleneck and compression layers are present, the model is Densenet-BC. Our version of the Densenet-BC architecture keeps the structure of the dense blocks and the transition layers, but changes the layers preceding and following the block structure. The dense blocks are preceded by two convolutional layers (the first one receives the unaltered FROG image) with kernel sizes of 11 and 7, respectively. The block structure is followed by a batch normalization layer and two fully connected linear layers. The last fully connected layer outputs the amplitude or phase array and has a size equal to k if the input image is of kxk size in pixels.

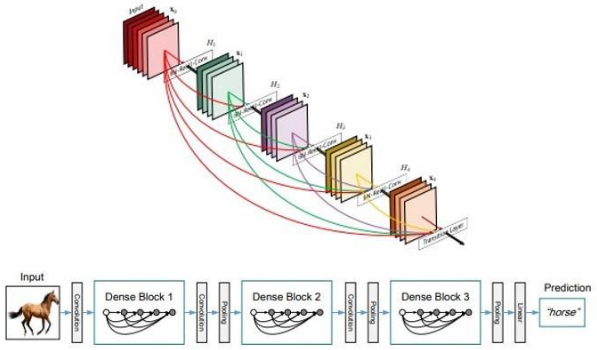


Fig. 6. Densenet architecture. Above is depicted a dense block and below is a typical arrangement of the blocks connected by transition layers. [5]

Reconstructed pulse shapes and model performance

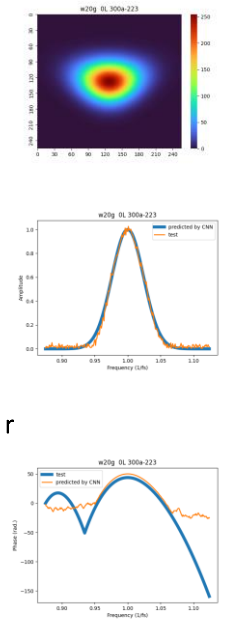


Fig. 7. Reconstructed amplitude and phase compared to the original ones.

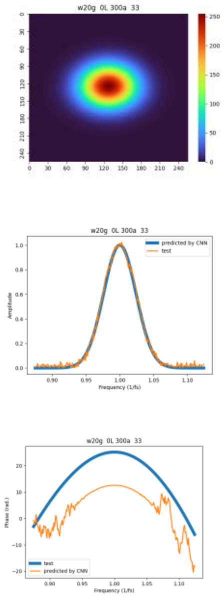


Fig. 8. Reconstructed amplitude and phase compared to the original ones.

Advised by the combined knowledge about experimental and simulated FROG trace properties, as well as the mechanism of CNN reconstruction process, we adopted the strategy to generate datasets gradually, that is from smaller sets to larger sets and from varying one parameter to two or more parameters varying at the same time. Below we show in Fig. 7 some results for FROG traces not seen by the model in the training or validation process. Since the angle most influenced the phase, we generated first a set of 600 data items varying only the angle, which further was split into training, validation and test sets. The angles (a) were set to have values in the range [-0.3, 0.3] radians. We set the width of the spectral amplitude $w = 20\ nm$, the glass thickness $g = 0\ \mu m$ and the grating distance $L = 300\ \mu m$ for this dataset. We show some results obtained using this dataset in Fig. 7, which shows a nice reconstruction of both the phase and the amplitude, especially for frequencies around the central frequency. However, given the reduced size of the dataset, the reconstruction is not so good for all data in the test set as we show such an example in Fig. 8. Since the amplitude shows a smaller variability than the phase, in both cases its reconstruction is correct. In Fig. 9 we show examples of results when both the grating distance [0, 300] μm and the glass thickness [0, 1000] μm were varied for an incident angle set at 0.1 radians or 100 mrad, and for three values of the spectral amplitude width [20, 35, 50] nm. The size of the dataset before splitting was of 4530 items. Below we show some results for the reconstruction of the spectral phase obtained for this dataset. Again, the reconstruction of the phase proved to be much more difficult than the amplitude: we have cases where the agreement with the original phase is very good, but also where some discrepancies may be observed. For all varying parameters the overall datasets can grow very large even for a relatively small number of values for the individual parameters, therefore we decided to generate datasets randomly, for which some preliminary results are under evaluation, but the model still needs some fine-tuning.

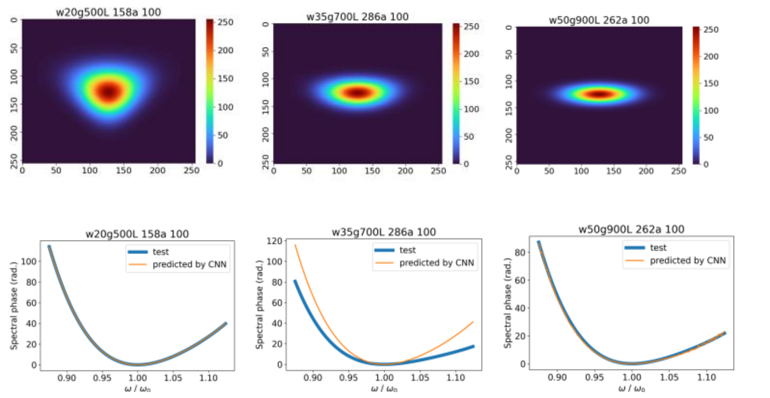


Fig. 9. Examples of FROG traces and corresponding reconstructed phases

References:

[1] R. Trebino et al, Review of Scientific Instruments 68, 3277 (1997)

[2] T. Zahavy et al, Optica 5, 5, 666-673 (2018)

[3] S. Kleinert et al, Optics Letters 44, 4, 979-982 (2019)

[4] V Aleksandrov, G Bleotu, L Caratas, R Dabu, I Dancus et al. *Upgrading design of a multi-TW femtosecond laser*, Romanian Reports in Physics, 2020, vol 72, p 413

[5] G. Huang et al, arXiv:1608.06993 (2017)