CS230

Reconstructing the temporal electric-field of attosecond X-ray pulses from an angularly streaked 2D photoelectron momentum distribution

Final Report
Category: Computer Vision

Paris Franz and Rachel Margraf
Stanford University
June 8, 2020
franzpl@stanford.edu and rmargraf@stanford.edu

Abstract

This project reconstructed the electric-field vs time profiles of attosecond X-ray pulses from images taken by a coaxial velocity map imaging spectrometer (VMI) using a deep neural net (NN). We trained our 5-layer fully connected network on simulated VMI images, then tested it on experimental VMI images taken at SLAC. Here, we describe our results, and compare our method to the non-linear fitting algorithms used in previous analyses.

1 Introduction

The coaxial velocity map imaging spectrometer (VMI) at SLAC measures the time-profile of attosecond X-ray pulses in the Linac Coherent Light Source (LCLS). In the VMI, an X-ray pulse is overlapped with a circularly-polarized infrared (IR) laser pulse in the presence of a gas. The X-ray pulse ionizes the gas, producing photoelectrons. The ejected electrons are separated by momenta, producing a 2D image when they hit the detector. From these images, information about the electric field of the X-ray pulse that produced the photoelectrons can be reconstructed (for a nice visual representation, see Appendix A). Our project's goal is to take an experimental VMI image and retrieve the electric-field time profile of the X-ray pulse.

2 Related work

2.1 Non-linear Fitting Algorithms

Past work reconstructing X-ray pulses from VMI images has primarily relied on decomposing the set of possible simulated VMI spectra into a set of von Neumann basis functions, and using non-linear fitting algorithms to express a given image as a linear combination of these basis functions [2]. These methods are reliable but slow, and require a good initial guess of the basis coefficients. They also require preprocessing of experimental images to reduce noise.

2.2 Reconstructing VMI images with deep learning

A recent student project [3] worked with this data set, and tried to determine good von Neumann basis coefficients by feeding VMI images into a Convolutional Neural Network (CNN). The intention was that these coefficients could be fed into the non-linear fitting algorithm to refine the result. This achieved some success, but had difficulties reconstructing the phase of the electric field. The author of that project advised us against using a CNN, as it is insensitive to the absolute location of pixels in an image, which is tied to the phase of the IR field in our problem.

The group who developed the non-linear fitting algorithm also briefly tried a neural network to predict the electric-field time profile of the X-ray pulse from VMI images. While that project did not achieve much success, since the simulated data [4] which we received was formatted in the correct format for that algorithm, we used some functions from that project to develop our baseline model.

2.3 Related Experimental Techniques utilizing Machine Learning

To our knowledge, the three efforts described above are the only projects which have worked on reconstructing VMI images of infrared-streaked X-ray pulses. However, other experimental methods have been utilized to measure attosecond pulses, and some have been assisted by deep learning. For example, in frequency-resolved optical gating, (FROG), an XUV pulse is split into two identical pulses, then overlapped with a time delay, τ, in the presence of a gas. By scanning τ and measuring the energy of photoelectrons produced, the electric field of the XUV pulse can be reconstructed via a non-linear fitting method [5]. Previous papers have circumvented the slow non-linear fitting methods with a deep neural net. One group used a 3-layer CNN followed by a 3-layer fully-connected network, and was successful at reconstructing both the electric field magnitude and phase of XUV pulses [6]. In a related technique, Frequency-resolved optical gating for complete reconstruction of attosecond bursts (FROG-CRAB) [7], a 3 (or more) layer CNN followed by a 2-layer fully-connected network was implemented [8]. These groups found that this CNN to fully-connnected network architecture worked well for them, even with moderately sized training sets (60,000 and 80,000 training examples respectfully), suggesting this may be an architecture worth exploring in the future.

3 Dataset and Features

We were provided two scripts used to generate simulated training examples. The first produces the von Neumann basis functions of possible VMI streaking spectra. We calculated these bases on a 64 x 64 grid to reduce aliasing effects due to calculating a circular distribution on a rectangular grid.

The second script generates a training set by randomly sampling possible combinations of these bases for X-ray beams of a given central frequency, frequency spread, 1st order spectral phase (arrival of the X-ray pulse in time), 2nd order spectral phase (frequency chirp) and 3rd order spectral phase (which governs additional higher order effects). We also scanned over possible values of the streaking laser intensity, E0. The script produces examples and labels of 64 x 64 pixel simulated VMI images and electric-field time arrays with 64 real elements, 64 imaginary elements, and one element containing E0. We then binned the VMI images down to 32 x 32 to make our model manageable to run.

We generated a set of 1,070,920 simulated VMI images and electric-field time profiles from three different sets of the von Neumann basis functions. 998,334 of these training pairs went into the training set. Since 998,334 images could not be held in memory at once, we trained on each basis set sequentially, causing bumps on the training set cost curve in Fig. 3. Training pairs within each basis set were randomly shuffled before training. Our development set consisted of 62,500 training pairs.

We also generated 12,000 simulated images containing noise using a frequency-filtering method depicted in Fig. 1. We extracted noise from 100 experimental images, binned to 256 x 256 pixels, by fourier-transforming the images, filtering out low frequencies then fourier-transforming back. Six modifications of these extracted noise images were produced by reflecting across the x-axis, y-axis or both axes, or by rotating 90°, 180°, or 270°, producing 600 noise images. For 20 different streaking laser intensities, E0, these modified noise images were normalized to the input intensity of the simulated image, then added to a randomly sampled simulated image, producing 12,000 images.

4 Methods

We used a 5-layer fully-connected network with tanh activation functions and 4,886,657 trainable parameters, as depicted in Fig. 2.

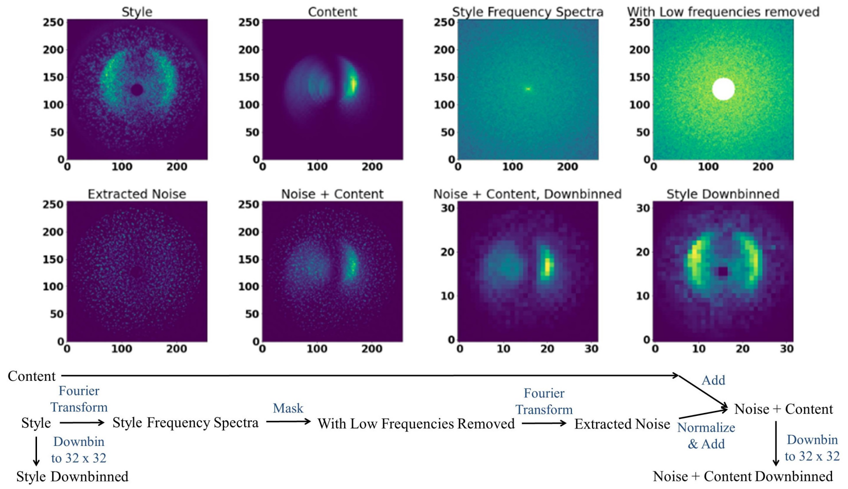

Figure 1: Noise addition by frequency filtering.

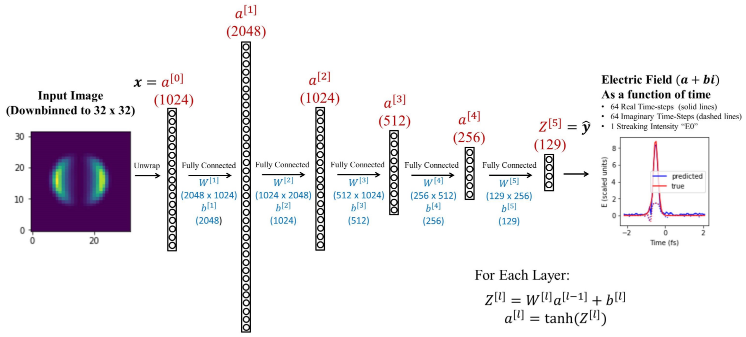
Input Image
(Downbinned to 32 x 32)
Unwrap
$x = a^{[0]}$
(1024)
$a^{[1]}$
(2048)
$a^{[2]}$
(1024)
$a^{[3]}$
(512)
$a^{[4]}$
(256)
$Z^{[5]} = \hat{y}$
(129)
Fully Connected
$W^{[1]}$ (2048 x 1024) $b^{[1]}$ (2048)
Fully Connected
$W^{[2]}$ (1024 x 2048) $b^{[2]}$ (1024)
Fully Connected
$W^{[3]}$ (512 x 1024) $b^{[3]}$ (512)
Fully Connected
$W^{[4]}$ (256 x 512) $b^{[4]}$ (256)
Fully Connected
$W^{[5]}$ (129 x 256) $b^{[5]}$ (129)
Electric Field $(a + bi)$
As a function of time
• 64 Real Time-steps (solid lines)
• 64 Imaginary Time-Steps (dashed lines)
• 1 Streaking Intensity "E0"
predicted
true
E (scaled units)
Time (fs)
For Each Layer:
$Z^{[l]} = W^{[l]}a^{[l-1]} + b^{[l]}$
$a^{[l]} = \tanh(Z^{[l]})$

Figure 2: Network architecture.

We modeled our network in Tensorflow 1 [9]. Back-propagation was performed using an Adam Optimizer [10], which modifies a gradient descent algorithm to correct for bias caused by initialization and implements a variable effective learning rate to train more quickly in the beginning of training, and more slowly later in training. We implemented a least-squares cost function, given in Fig. 3 where n_L is the number of outputs in the output layer ($n_L = 129$) and m is the number of training examples. During development, we scanned over possible learning rates and minibatch sizes, and found we needed a fairly small minibatch size and very small learning rate for the best performance. Based on our scan, we chose a minibatch size of 64 and a learning rate of 0.00001. We trained our model for 1100 epochs. We trained on three different training sets, resulting in bumps in the minibatch cost curve; however, the cost for a constant development set consistently decreased.

$$\mathbf{Cost} = \frac{\mathbf{1}}{\mathbf{m}}\frac{\mathbf{1}}{\mathbf{n_L}}\sum_{\mathbf{j=1}}^{\mathbf{m}}\sum_{\mathbf{i=1}}^{\mathbf{n_L}}(\mathbf{Z_i^{[5](j)}} - \mathbf{Y_i^j})^\mathbf{2}$$

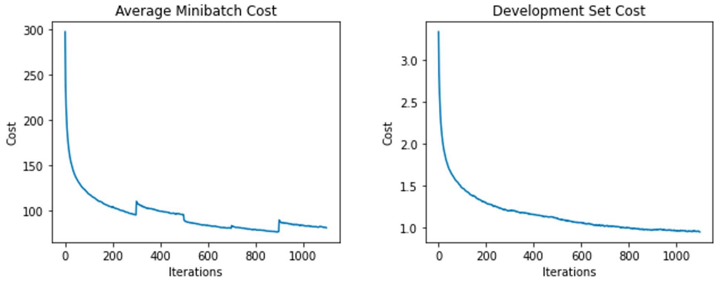

Figure 3: Cost functions.

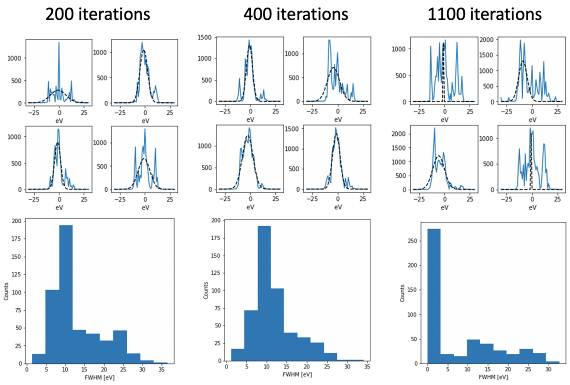

Figure 4: The parameters used to predict the electric-fields were trained for 200, 400, and 1100 iterations. (Top) Spectra for four examples with fitted gaussians in black. (Bottom) Histogram of bandwidth (in eV) of predicted pulses for 496 experimental examples.

We ended up early stopping training our parameters at 400 iterations, where the development set cost was 1.15. Although the cost was still decreasing, our other sanity checks were revealing concerning results. For a set of 496 experimental images we calculated the frequency spectra of the predicted electric fields by taking the Fourier transform, shifting the zero-frequency component to the center of the spectrum to remove the carrier frequency, and multiplying by the complex conjugate. Based on experimental measurements, we expect the electric field pulses to be gaussian-like in shape and have a central bandwidth of approx. 7.5 eV. However, as the training iterations increase, we see the spectra become more spiky and the spectral distribution broaden away from 7.5 eV, as shown in Fig. 4. Our training set was randomly generated, and so contains both realistic and unrealistic electric-fields. We suspect as the training iterations increase, the model learns to better fit the unrealistic pulses, at the cost of fitting the realistic (single, narrow peak) pulses worse.

5 Experiments/Results/Discussion

We trained and tested our model on simulated data. For a simulated data set of size 62,500, our mean squared error was 1.15.

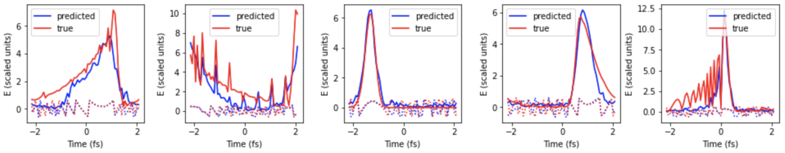

Figure 5: Predicted and true electric fields for example simulated data. The solid lines show the amplitude, and the dotted lines show the phase. The phase has been scaled for clarity.

The nonlinear fitting algorithm produces 20 possible electric-field profiles per shot, shown as transparent grey lines in Fig. 6. There is good agreement between our model's predictions for amplitude of the electric field and the verified fitting algorithm.

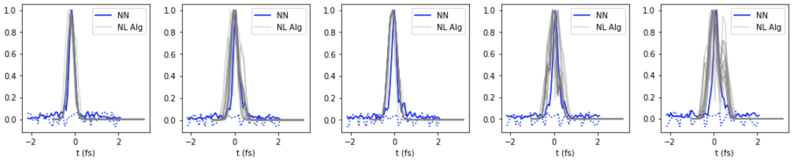

Figure 6: Comparison of nonlinear fitting algorithm and NN model. Normalized amplitudes are shown by solid lines. Predicted phase is shown by the dotted line, and is scaled for clarity.

We trained our model on simulated data without noise; however, the experimental data had noise present. From the model's predicted electric-field we can simulate a VMI image, and compare this predicted image to the model input image, as shown in Fig. 7. Comparing the two, it appears that our model has learned to ignore the center of the VMI image and the outer ring, where the signal is low and conveniently where noise is present.

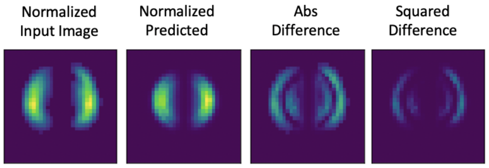

Figure 7: Comparison of an input VMI image with an image from a predicted electric field. The absolute value of the difference between the left two images, and the squared difference are shown. The input and predicted images have been normalized to their respective max values.

Figs. 6 and 7 show our NN's fitting to processed experimental images. Ultimately, we would like our model to perform well on unprocessed images. To help our model learn what noisy experimental images look like, we trained our model for an additional 100 epochs on 12,000 simulated images with added noise. A few sample shots for this method compared to our original algorithm are shown in Fig. 8. While we don't have a non-linear fit to compare these values to for this dataset, we see that the fit is fairly consistent for some, while very different for others. The fit seems to change the most when part of one lobe is fainter than the rest. This suggests the algorithm may be ignoring fainter pixels in the inner ring as noise.

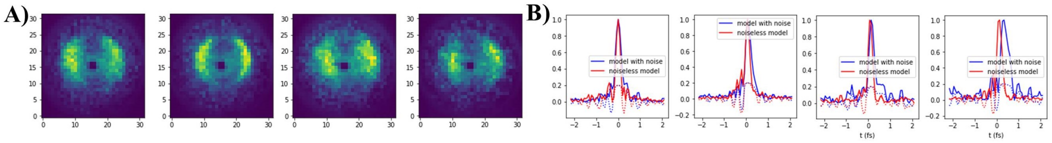

Figure 8: Comparison of NN model trained purely on simulated data to NN model trained for 100 additional epochs on 12,000 simultated images with added noise. A) Four sample unprocessed experimental images binned to 32 x 32. B) The NN algorithm predictions for the same four images, with solid lines for intensity and dashed lines for phase.

6 Conclusion/Future Work

We successfully demonstrated agreement between our model output and a previously verified algorithm. It is important to note that our model is both fast and also extracts phase information, which matches well for simulated images but has yet to be verified for experimental data.

Our model can be expanded on by exploring a deeper/broader model that may be better able to fit more interesting pulses. In particular, it may be of interest to train a similar model on larger 64 x 64 images, which may contain more information in the fine structure. To do this, we might also attempt other architectures, such as adding a few convolutional layers, to reduce the total number of trainable parameters in our system. We also may want to include other interesting pulse shapes in our dataset, such as two pulses close in time (see Appendix C).

7 Contributions

PF and RM discussed the algorithm architecture, hyperparameters, metrics and results together, and contributed equally to the production of reports. PF ran the algorithm training, imported and analyzed the results from the non-linear fitting method for comparison, and produced the fit interpretation plot in Fig. 7. RM investigated two methods of producing a dataset of simulated images with added noise, the first of which (using style transfer) was not used, but is shown in Appendix B, and the second, frequency filtering, is shown in Fig. 1.

8 Acknowledgements

The authors would like to thank James Cryan and Taran Driver for their help in obtaining datasets and guiding our project. We would also like to thank Siqi Li for providing the results of her non-linear fitting algorithm, and Zhaoheng Guo for discussing with us his previous work on this topic.

References

Our code (but not the data due to storage constraints) can be found at: https://github.com/parisfranz/cs230-project.git in the Final_code folder.

[1] J. Duris et al., “Tunable isolated attosecond X-ray pulses with gigawatt peak power from a free-electron laser,” Nat. Photonics, vol. 14, no. 1, pp. 30–36, Jan. 2020, doi: 10.1038/s41566-019-0549-5.

[2] S. Li et al., “Characterizing isolated attosecond pulses with angular streaking,” p. 17.

[3] Z. Guo, “Characterizing Attosecond X-Ray Free Electron Pulses with Convolution Neural Network,” Project for CS231, June 2019, Available: `https://github.com/zhgstanford/CS231n_Final_Project`.

[4] Taran Driver, private communication, 2020.

[5] R. Trebino et al., “Measuring ultrashort laser pulses in the time-frequency domain using frequency-resolved optical gating,” Review of Scientific Instruments, vol. 68, no. 9, pp. 3277–3295, Sep. 1997, doi: 10.1063/1.1148286.

[6] J. White and Z. Chang, “Attosecond streaking phase retrieval with neural network,” Opt. Express, vol. 27, no. 4, p. 4799, Feb. 2019, doi: 10.1364/OE.27.004799.

[7] Y. Mairesse and F. Quéré, “Frequency-resolved optical gating for complete reconstruction of attosecond bursts,” Phys. Rev. A, vol. 71, no. 1, p. 011401, Jan. 2005, doi: 10.1103/PhysRevA.71.011401.

[8] T. Zahavy et al., “Deep learning reconstruction of ultrashort pulses,” Optica, vol. 5, no. 5, p. 666, May 2018, doi: 10.1364/OPTICA.5.000666.

[9] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Version 1.15.2. Software available from tensorflow.org.

[10] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” arXiv:1412.6980 [cs], Jan. 2017, Accessed: Jun. 08, 2020. [Online]. Available: `http://arxiv.org/abs/1412.6980`.

[5] L. A. Gatys, A. S. Ecker, and M. Bethge, “A Neural Algorithm of Artistic Style,” arXiv:1508.06576 [cs, q-bio], Sep. 2015, Accessed: May 22, 2020. [Online]. Available: http://arxiv.org/abs/1508.06576. Pretrained dataset: imagenet-vgg-verydeep-19.mat from https://www.vlfeat.org/matconvnet/pretrained/

Appendix A

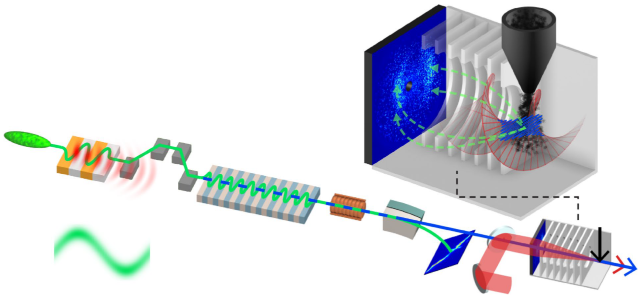

Figure 9: Production and measurement of attosecond X-ray pulses at LCLS. An accelerated electron bunch (green) passes through a series of undulator magnets to produce X-rays (blue). In the VMI, the X-ray pulse is overlapped with an infrared pulse in the presence of a gas jet. Photoelectrons are produced and imaged on a 2D screen to reconstruct the pulse's time-profile [1].

Appendix B

Fig. 10 shows our initial attempt at adding noise to simulated images. While a lot of effort went into this, ultimately adding noise by frequency filtering was computationally faster and produced results better resembling experimental images.

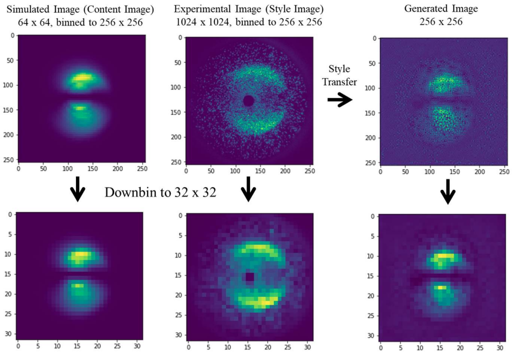

Figure 10: Initial noise addition method using style transfer algorithm.

Starting with homework C4M4 as a template, we performed style transfer using pretrained values from the 19-layer Convolutional Neural Net VGG-19 [11]. We first binned a simulated image and an experimental image to 256 x 256 pixels. Since the VGG network requires a three channel input, and our VMI images have one channel, we copied our single channel into all three channels, then averaged the result at the end to get a one channel output. We performed style transfer with 60% style weight from conv1_1, the first layer, and 40% style weight from conv5_1, the 13th layer. These two were chosen by visibly comparing style transfers from different layers after 200 epochs. Conv1_1 provided very small textures, and conv5_1 provided a fairly uniform background.

To prepare these images to feed into our neural net, we then binned them down to 32 x 32 pixels. We found it was necessary to perform the style transfer with larger images when using the VGG-19 dataset, however. The VGG-19 dataset is designed for images of 300 x 400 pixels, and thus we expect the trained filters are too large to stylize details in comparably tiny 32 x 32 images.

While style transfer may be a viable method for noise modeling if we had pre-trained parameters on similar images, other methods can achieve a similar or better effect without using a NN.

Appendix C

Fig. 11 shows a test of our algorithm on two-pulse data, which it was not exposed to in the training set.

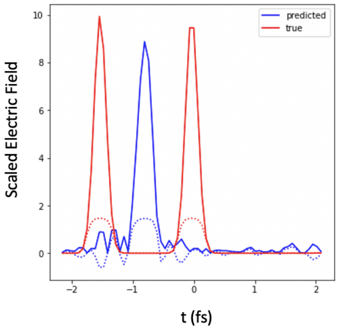

Figure 11: As expected, a model trained exclusively on single pulse data cannot reconstruct electric fields with two pulses. It is interesting that it does predict a single pulse in between the the two real pulses