



# Deep learning in attosecond metrology

CHRISTIAN BRUNNER,<sup>1,3</sup> ANDREAS DUENSING,<sup>1,3</sup>  CHRISTIAN SCHRÖDER,<sup>1</sup> MICHAEL MITTERMAIR,<sup>1</sup> VLADIMIR GOLKOV,<sup>2</sup> MAXIMILIAN POLLANKA,<sup>1</sup> DANIEL CREMERS,<sup>2</sup> AND REINHARD KIENBERGER<sup>1,\*</sup>

<sup>1</sup>Department of Physics, Technical University of Munich, James-Frank-Str. 1, 85748 Garching, Germany

<sup>2</sup>Department of Informatics, Technical University of Munich, Boltzmannstr. 3, 85748 Garching, Germany

<sup>3</sup>These authors contributed equally

\*reinhard.kienberger@tum.de

**Abstract:** Time-resolved photoelectron spectroscopy provides a versatile tool for investigating electron dynamics in gaseous, liquid, and solid samples on sub-femtosecond time scales. The extraction of information from spectrograms recorded with the attosecond streak camera remains a difficult challenge. Common algorithms are highly specialized and typically computationally heavy. In this work, we apply deep neural networks to map from streaking traces to near-infrared pulses as well as electron wavepackets and extensively benchmark our results on simulated data. Additionally, we illustrate domain-shift to real-world data. We also attempt to quantify the model predictive uncertainty. Our deep neural networks display competitive retrieval quality and superior tolerance against noisy data conditions, while reducing the computational time by orders of magnitude.

© 2022 Optica Publishing Group under the terms of the [Optica Open Access Publishing Agreement](#)

## 1. Introduction

Ultrafast electron dynamics in atoms and molecules are of major scientific interest as they are pertinent to a vast array of cutting-edge technologies ranging from optoelectronics and information processing to molecular electronics and photovoltaics. Time-resolved photoelectron spectroscopy allows the study of these dynamics with unprecedented temporal resolution.

A key technique in the field is the attosecond streak camera [1–3], also called (attosecond) streaking for brevity. Over the past decade, streaking experiments have been realized to attain a number of remarkable results, including the discovery of relative photoemission delays in gas phase [4] and solid-state [5] samples, and ultimately the measurement of the absolute timing of the photoelectric effect [6].

In practice, the evaluation of streaking experiments remains challenging. All of the information that can be gained from a measurement is encoded in a streaking spectrogram. To tackle the problem of spectrogram inversion, sophisticated data analysis methods have been devised. On the downside, most common algorithms either lack precision as they rely on coarse physical approximations or come with prohibitively large computational cost, restricting their use with growing datasets. We present an alternative approach, exploring the application of contemporary deep learning based machine vision models for attosecond streaking evaluation, that can in principle break these limitations.

Owing to the rapid advances in terms of general purpose GPUs, the development of extensive software platforms like TensorFlow [7] and Torch [8] and the emerging availability of large-scale datasets, deep learning research achieved remarkable success in a wide spectrum of fields within recent years. Deep neural networks cut error rates in image classification and segmentation in unprecedented fashion [9–11], revolutionized applications like machine translation [12,13], reached expert human-level performances in abstract board games like Chess, Shogi and Go

[14,15] and ventured out to scientific applications like automated analysis of strong gravitational lensing systems [16] or online particle detection at CERN [17].

In ultrafast physics, deep learning has recently been applied to the problem of characterizing femtosecond pulses from FROG [18] and d-scan [19] traces and for basic phase retrieval from attosecond streaking spectrograms [20]. Building on these recent results, we demonstrate the application of popular neural network architectures for the complete reconstruction of single streaking traces. Furthermore, we extend the developed method to spectrograms with two streaking traces. This allows us to extract relative photoemission delays between electrons originating from different bound states in a sample.

## 2. Problem overview

Attosecond streaking experiments employ a pump-probe scheme to experimentally investigate photoemission dynamics in a sample. A single isolated attosecond pump pulse  $E_{\text{XUV}}(t)$  in the extreme-ultraviolet (XUV) region is used to generate photoelectrons. The presence of a few-cycle near-infrared (NIR) probe pulse  $E_{\text{NIR}}(t)$  modulates the final kinetic energy of the freed electrons, an interaction referred to as *streaking*. This modulation follows the vector potential of the NIR pulse (i.e. the time integral of the electric field) in case the XUV pulse is significantly shorter than one half-cycle of the NIR field. The result of an attosecond streaking experiment consists of a sequence of laser-dressed photoelectron spectra recorded in a range of NIR-XUV delays, which comprise a two-dimensional *spectrogram*.

It should be noted that in this work we consider the most common streaking geometry where XUV and NIR fields are both linearly polarized in the same direction, which also coincides with the observation direction of the photoelectron detector. Therefore, all physical quantities may be treated as scalars.

### 2.1. Attosecond streaking theory

According to quantum theory under the strong field approximation (SFA) [21] and the wavepacket approximation (WPA) [22], the probability of detecting a photoelectron at final energy  $E_f$  with NIR-XUV delay  $\tau$  using atomic units is

$$S_{\text{WPA}}(E_f, \tau) = \left| \int_{-\infty}^{\infty} dt \chi(t + \tau) e^{j(E_f + I_p)t} e^{j\Phi_V(E_f, t)} \right|^2, \quad (1)$$

where  $\chi(t)$  is the photoelectron wavepacket in the time domain,  $I_p$  the ionization potential of the target and  $j^2 = -1$ . The interaction of the liberated photoelectron with the NIR laser field causes a phase gating described in the Volkov term

$$\Phi_V(E, t) = - \int_t^{\infty} dt' \left( \sqrt{2E} A_{\text{NIR}}(t') + \frac{1}{2} A_{\text{NIR}}^2(t') \right), \quad (2)$$

where  $A_{\text{NIR}}(t)$  denotes the vector potential of the streaking laser field in Coulomb gauge.

Applying the WPA is a convenient starting point for our work, as it allows to apply our models directly for the retrieval of photoelectron wavepackets. In particular, we are not required to make specific assumptions about the physical properties of the target, e.g. the shape of the dipole transition matrix element. Analogously to a laser pulse, we choose a complex representation for the wavepacket in the energy domain

$$\tilde{\chi}(E) = \sqrt{I_{\text{WP}}(E)} e^{j\phi_{\text{WP}}(E)}, \quad (3)$$

decomposing the signal into power spectrum  $I_{\text{WP}}(E)$  and spectral phase  $\phi_{\text{WP}}(E)$ . Common classical methods, briefly described in the following, retrieve the photoelectron wavepacket either in the time or frequency domain.

## 2.2. CMA-based retrieval methods

In the following, we briefly review two widespread groups of algorithms commonly used for the evaluation of attosecond streaking spectrograms. The key to both presented methods is the central momentum approximation (CMA), which allows us to reformulate the streaking trace computation as an inverse Fourier transform

$$S_{\text{CMA}}(E_f, \tau) \propto |\mathcal{F}^{-1} [\chi(t + \tau) \tilde{G}(t)]|^2, \quad (4)$$

where  $\tilde{G}(t)$  is a complex gate function essentially containing the Volkov term, Eq. (2). As neither wavepacket nor gate can contain energy-dependent terms, impeding dependencies are approximated with the constant central-energy of the unstreaked wavepacket.

### 2.2.1. FROG-CRAB retrieval

Methods based on frequency resolved optical gating for complete reconstruction of attosecond bursts (FROG-CRAB) [23] are presently the most popular algorithms for spectrogram inversion. Powerful iterative algorithms exist for the simultaneous reconstruction of wavepacket and gate function from Eq. (4) without prior knowledge, such as the extended Ptychographic engine (ePIE) [24].

### 2.2.2. Restricted TDSE retrieval

The restricted time-dependent Schrödinger equation (rTDSE) method represents the classical route to formulating spectrogram inversion as an optimization problem. By explicitly parametrizing the wavepacket and NIR vector potential, one may use standard-least squares to fit streaking theory to recorded data. A strength of rTDSE is the possibility to incorporate physical insight into the fitting procedure through choice of parametrization and *restricting* certain parameter ranges. While not typically used for the retrieval of single wavepackets, rTDSE has successfully been applied for determination of photoemission delays from multi-trace spectrograms in a number of studies, e.g. [4,6,25,26].

## 3. Methods

In this work, we train feed-forward deep neural networks (DNNs) for the complete evaluation of attosecond streaking spectrograms. We divide the problem of single-trace retrieval into three sub-tasks. Individual networks are trained to predict the NIR vector potential, the wavepacket spectrum and the wavepacket spectral phase. In contrast to the rTDSE method, the DNN approach does not rely on parametrization of the signals. In the case of dual-trace spectrograms we add two additional networks, one for the second wavepacket spectrum and spectral phase, respectively.

Each sub-task is formulated in a regression setting. Attosecond streaking spectrograms are used as inputs. They are provided as two-dimensional arrays over energy and delay dimension and with a single data channel representing the relative electron yield. To realize a supervised learning scheme we use the WPA-based simulation (cf. Equation (1)) that maps the vector potential and wavepackets to a spectrogram. We train the neural networks to approximate the inverse mapping, i.e. to reconstruct the inputs of the simulation from its output.

The following sections account for important aspects in our methodology. In particular, we describe the streaking simulation, data pre-processing, DNN training as well as retrieval quality criteria and predictive uncertainty estimation.

### 3.1. Dataset generation

Our training, validation and test datasets are generated by numerical simulation of streaking traces. Since sufficiently large high-quality datasets of fully labeled real-world spectrograms

are not available, simulation offers the main way of providing the data required for supervised learning. Moreover, it allows to benchmark the expected performance of deep learning against established methods.

The following sections provide additional details on the two generated datasets. Prior to the subsequent processing, both datasets are divided in a 80/10/10 train, validation, test split.

### 3.1.1. Photoelectron wavepacket retrieval dataset

For the attosecond photoelectron wavepacket and vector potential retrieval, we simulate 275,000 single-trace streaking spectrograms. For each spectrogram, we generate one electron wavepacket and one NIR pulse. Wavepacket spectral phases have their zero- and first-order polynomial coefficients fixed to zero to circumvent label ambiguities, i.e. cases where different labels would explain the same spectrogram. This could occur as streaking is insensitive to the carrier-envelope-phase of the wavepacket manifesting as a zero-order spectral phase term [27]. Moreover, streaking is only sensitive to the relative NIR-XUV delay and not to simultaneous time shifts. Setting the first-order phase term to zero, we guarantee that the wavepacket serves as absolute time reference.

### 3.1.2. Photoemission Time delay retrieval dataset

For the photoemission delay retrieval experiments, we simulate 100,000 total spectrograms with two streaking traces each. This represents the case of electron emission from two electronic states with different binding energies. In contrast to the previous dataset, we now generate two photoelectron wavepackets and one NIR pulse per spectrogram. We separately compute one streaking trace per wavepacket using the single NIR vector potential and add up the two traces for the full observable spectrogram.

For this dataset we uniformly sample a random first-order phase term for one of the wavepackets, corresponding to a relative central-energy photoemission time delay  $d \in [-0.25, 0.25]$  fs. Thereby, we explicitly introduce a relative phase shift of the second trace with respect to the first. Recovering the correct photoemission time delay value from the spectrogram is the main challenge posed in dual-trace retrieval.

## 3.2. Dataset pre-processing

In order to make our networks more robust and start bridging the gap from pure simulated streaking retrieval to realistic real-world conditions, we develop a pre-processing pipeline for our datasets. The full pipeline is divided in two stages. In the first stage we attempt to incorporate common effects present in experiments, most notably background and noise. The second stage of the data processing pipeline implements the functionality that is identically applied to prepare real-world data for evaluation with our neural networks.

To evaluate the performance of our algorithms in dependence of input data quality, it is important to quantify the noise level in the simulated spectrograms. We quantify the amount of noise by estimating the signal-to-noise ratio (SNR) as

$$\text{SNR} = 10 \log_{10} \left[ \frac{\sum_i \sum_j S[i,j]}{\sum_i \sum_j |S[i,j] - S'[i,j]|} \right], \quad (5)$$

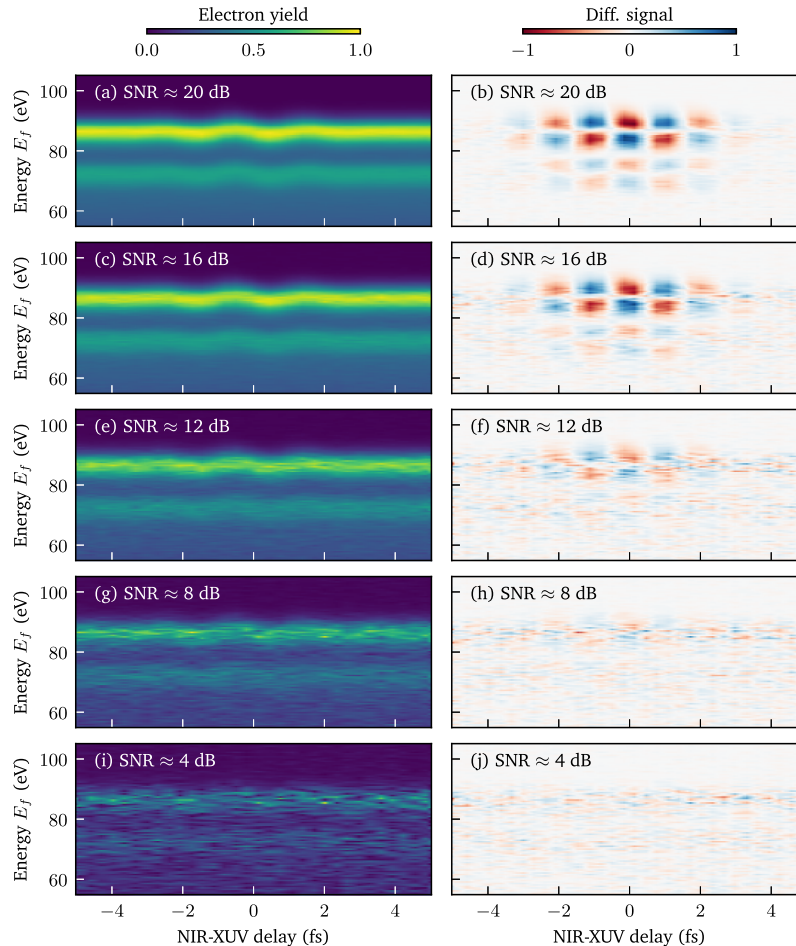
where  $S$  is the spectrogram with the photoelectron background and  $S'$  is the same spectrogram but including short-time laser fluctuations, multiplicative and additive noise. The sum over all pixels in the numerator approximates the total signal content in the spectrogram. Conversely, the sum over the absolute differences in the denominator approximates the total noise content. We express the SNR on the logarithmic scale in decibel.

As the simulation of large numbers of WPA streaking spectrograms without application of the CMA remains a computationally expensive task, we also make use of the pre-processing



pipeline to augment our training datasets. By pre-processing the base spectrograms with different realizations of background and noise, we can efficiently make our networks more robust to these adverse factors. We compute four realizations of the base training datasets, each with signal-to-noise ratio (SNR) uniformly distributed over the range 20 dB to 4 dB.

Figure 1 provides a visual impression for the quality of the simulated data after all processing steps. The left column shows a unique WPA-based spectrogram at five distinct noise levels. In the corresponding right column, we additionally applied the optional differential background subtraction as explained in the following.



**Fig. 1.** Visual impression of the range of simulated signal-to-noise ratios (SNR). Pseudo-color plots of a unique WPA-based spectrogram without background subtraction (left column) and with differential background subtraction (right column). Applying our pre-processing pipeline we generate data that resemble noisy real-world conditions.

### 3.3. Differential background subtraction

A challenging task in photoelectron spectroscopy is the separation of signal and background. One convenient method, originally developed for use in rTDSE streaking retrieval, is to differentiate the spectrogram along the NIR-XUV time delay dimension. This procedure, called differential background subtraction, removes the signal of non-streaked electrons that is constant along all

delays and emphasizes the zero-crossings of the NIR vector potential [4]. In comparison to most other techniques it does not assume a specific background shape and does not require user input that could affect the results [28]. We test our method both on data without background subtraction (DNN) and data after differential background subtraction (diffDNN), deemed *differential signal*.

### 3.4. Neural networks

In essence, a DNN is a graph (usually a sequence) of mappings referred to as layers, describing a mapping from an input to an output space. For image processing tasks it is common to use a special class called convolutional neural networks.

We train three popular neural network architectures, VGG11 [29], GoogLeNet (Inception-v2) [30,31] and ResNet50 [32]. To adapt these to our regression tasks, we cut off each network after its final pooling layer, attach a single fully-connected hidden layer with ReLU activations and a fully-connected layer with linear activations to compute an output vector.

### 3.5. Supervised learning

The most common form of machine learning is supervised learning. Let  $\mathcal{D} = \{X_{\text{train}}, Y_{\text{train}}\}$  be a dataset containing  $N$  independent and identically distributed (i.i.d.) input samples  $x^{(n)} \in \mathcal{X}$  and desired output labels  $y^{(n)} \in \mathcal{Y}$ . We use a DNN  $f : \mathcal{X} \rightarrow \mathcal{Y}$  controlled through parameters  $\vec{\theta}$  to find a mapping that generalizes to new unseen samples. To learn the parameters of the DNN, we minimize the cost function

$$\mathcal{J}(\vec{\theta}) = \frac{1}{N} \sum_{n=0}^{N-1} \mathcal{L}(y^{(n)}, f(x^{(n)}; \vec{\theta})), \quad (6)$$

where  $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$  is a scalar-valued loss function measuring the discrepancy between label and DNN prediction. The specific form of the loss function is typically derived from an underlying probabilistic model.

For training our networks, we set up one per-sample loss function for each sub-task. In our notation, quantities with a hat symbolize the neural network predictions. For the NIR vector potential  $\vec{A}_{\text{NIR}}$ , we compute the normalized  $L^2$  error

$$\mathcal{L}_A(\vec{A}_{\text{NIR}}, \hat{A}_{\text{NIR}}) = \frac{\|\vec{A}_{\text{NIR}} - \hat{A}_{\text{NIR}}\|_2}{\|\vec{A}_{\text{NIR}}\|_2}. \quad (7)$$

Analogously, our per-sample loss for the wavepacket spectrum  $\vec{I}_{\text{WP}}$  is computed as the normalized  $L^2$  error  $\mathcal{L}_I(\vec{I}_{\text{WP}}, \hat{I}_{\text{WP}})$ . In both cases, the normalization is used to cancel out effects of arbitrarily chosen units.

Regarding the wavepacket  $\vec{\phi}_{\text{WP}}$ , we need to be aware that its spectral phase becomes meaningless in regions where the power spectrum is small. To disregard the corresponding values, we introduce a weighted square error with weights  $w[i]$  assigned to each energy  $E_i$  ( $i = 0, \dots, I - 1$ ):

$$\mathcal{L}_\phi(\vec{\phi}_{\text{WP}}, \hat{\phi}_{\text{WP}}) = \sum_{i=0}^{I-1} \left[ w[i] (\phi_{\text{WP}}[i] - \hat{\phi}_{\text{WP}}[i])^2 + \eta (I_{\text{WP}}[i] - \hat{I}_{\text{WP}}[i])^2 \right]. \quad (8)$$

where  $\eta$  is a relative scaling factor. Ideally, the weights should be chosen according to the probabilistic model. However, in practice, values are not evident a priori. Therefore, we use a physics-motivated approach based on the wavepacket power spectrum labels by defining

$$w[i] = \mathbb{1}(I_{\text{WP}}[i] > t_h), \quad (9)$$

where  $t_h$  is a fixed threshold that is set to 0.01 in our experiments. As implied by Eq. (9), the particular networks learning the wavepacket spectral phase retrieval also implicitly require

knowledge of the wavepacket power spectrum labels. In our tests it has turned out beneficial to also use this knowledge explicitly by adding an auxiliary output to the networks that is tasked to predict the wavepacket spectrum. The additional output is thereafter included into the loss function Eq. (8) via the second term. For the presented experiments, we set the relative scaling factor  $\eta$  to 1. We observe that this simplifies network training by making convergence more likely and improving convergence rates. Note that the additional output is only used in network training and discarded later-on in prediction, as networks which are specifically trained for wavepacket spectrum prediction perform superior in this task.

In order to track the progress during training and report results, we average per-sample quality metrics. Regarding the NIR vector potentials and wavepacket spectra, we directly use the normalized  $L^2$  errors described in Eq. (7) as quality metrics. When considering the wavepacket spectral phase, we additionally define the quality criterion

$$Q_\phi(\vec{\phi}_{\text{WP}}, \hat{\phi}_{\text{WP}}) = \left[ \frac{\sum_i w[i] (\phi_{\text{WP}}[i] - \hat{\phi}_{\text{WP}}[i])^2}{\sum_i w[i]} \right]^{\frac{1}{2}}. \quad (10)$$

When plugging in the weights given in Eq. (9), the term inside the brackets boils down to the average square error over the regarded phase samples given the selected threshold. Note that the physical units of this metric are radians and we forgo defining a relative measure because absolute phase values are physically meaningful.

By combining the obtained NIR vector potential and wavepacket retrievals, we can compute a spectrogram reconstruction  $\hat{S}$ . To compare the latter to the original input spectrogram, we define the quality criterion

$$Q_S(S, \hat{S}) = \left[ \frac{\sum_i \sum_j (S[i, j] - \hat{S}[i, j])^2}{\sum_i \sum_j (S[i, j])^2} \right]^{\frac{1}{2}}. \quad (11)$$

Note that we use the raw spectrogram before noise and background is added as reference  $S$  here. The bare spectrogram, which is only available for simulated data, facilitates the most direct comparison to the computed reconstruction. The training duration depends on the extent of the training dataset and is typically below two hours for a complete model, including XUV spectrum and phase as well as NIR vector potential.

### 3.6. Predictive uncertainty estimation

Despite their tremendous success on a variety of tasks, DNN models typically lack the ability to quantify the inherent uncertainty in their predictions. In many practical settings it is crucial to establish meaningful confidence intervals in addition to accurate point predictions. In our work, this issue is aggravated by the fact that we exclusively train on simulated data, but want to apply the models to infer on real-world spectrograms.

For evaluating real-world data, we follow a common deep ensemble approach. We train  $M = 10$  models starting from different random parameter initialization and random sample shuffling during optimization. By computing the empirical variance of the individual network point predictions, we derive an approximate model uncertainty measure.

## 4. Results and discussion

In the following, we show and discuss results for single- and dual-trace retrieval. In both applications, we follow the same procedure. We start off by training and validating results for different DNN architectures on the large simulated datasets. To illustrate the performance of our best network instances, we discuss one representative test set retrieval. Then we use a smaller

simulated dataset with variable input noise level to benchmark against the state-of-the-art retrieval method for the respective task. In the case of photoelectron wavepacket retrieval, we compare our method to ePIE. For photoemission delay retrieval we compare our results to rTDSE as it is typically superior at this task. Finally, we demonstrate the applicability of our networks to real-world data, using deep ensembles to estimate the predictive uncertainty.

#### 4.1. Photoelectron wavepacket retrieval

The present section is concerned with obtaining a full retrieval of the electron wavepacket and NIR vector potential from a single-trace attosecond streaking spectrogram. For the electron wavepacket this encompasses spectrum, as well as spectral phase. Assuming the dipole transition matrix element of the target is well-known this retrieval allows to fully characterize NIR and XUV laser pulses using a streak camera setup.

##### 4.1.1. Retrieval example

A showcase for the retrieval results of our neural networks in comparison to the ePIE implementation on simulated data is provided in Fig. 2.

Panel (a) depicts the pseudo-color plot of a simulated test set spectrogram, which was pre-processed with our pipeline to exhibit a SNR  $\approx 12$  dB. It directly serves as input for the DNN and classical ePIE retrieval. In the diffDNN approach, differential background subtraction (not shown here) is applied prior to the network.

Panels (b), (d) and (f) display the NIR vector potential retrievals by DNN, diffDNN and ePIE respectively. Each result is plotted together with the label for comparison. In summary, the NIR reconstructions are excellent across all methods. The differential background subtraction is observed to slightly impair the retrieval quality in comparison to the non-background subtracted case. Judging from our experiments, the ePIE implementation shows a tendency to slightly underestimate the amplitudes of the signal maxima.

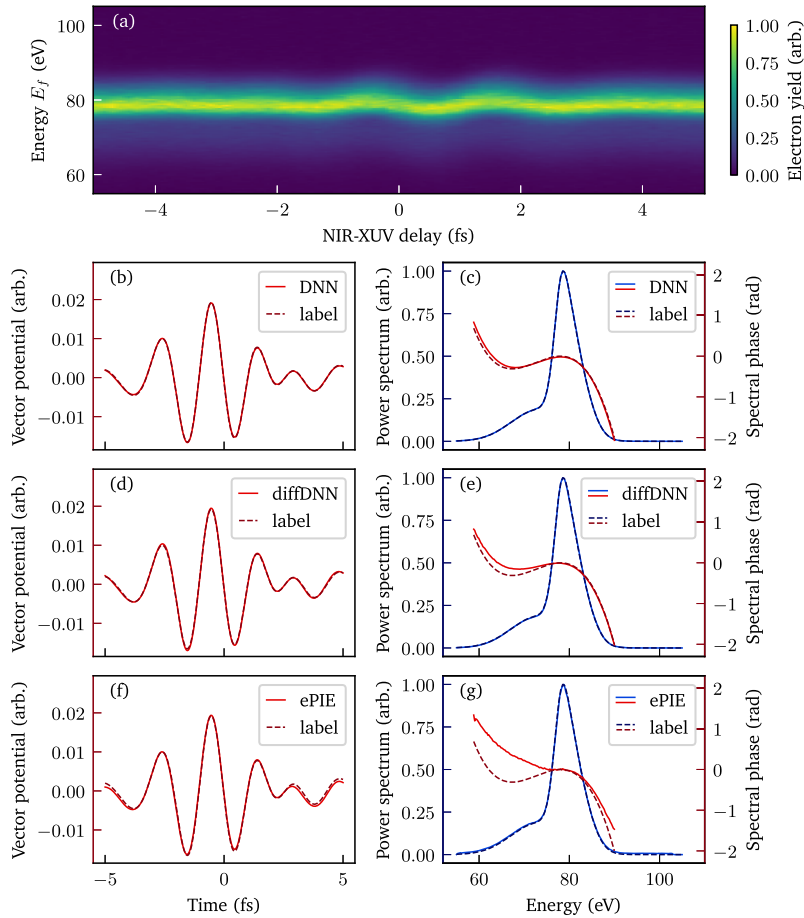
Panels (c), (e) and (g) display the retrieved wavepacket spectra (blue) and spectral phases (red) compared to their label. Regarding the spectra, all considered methods show great accordance to the label. Considering the spectral phase retrieval, both DNN and diffDNN approaches exhibit excellent agreement with the label shape. In this task the ePIE implementation is lagging behind, often yielding more noisy results.

##### 4.1.2. Quality benchmark

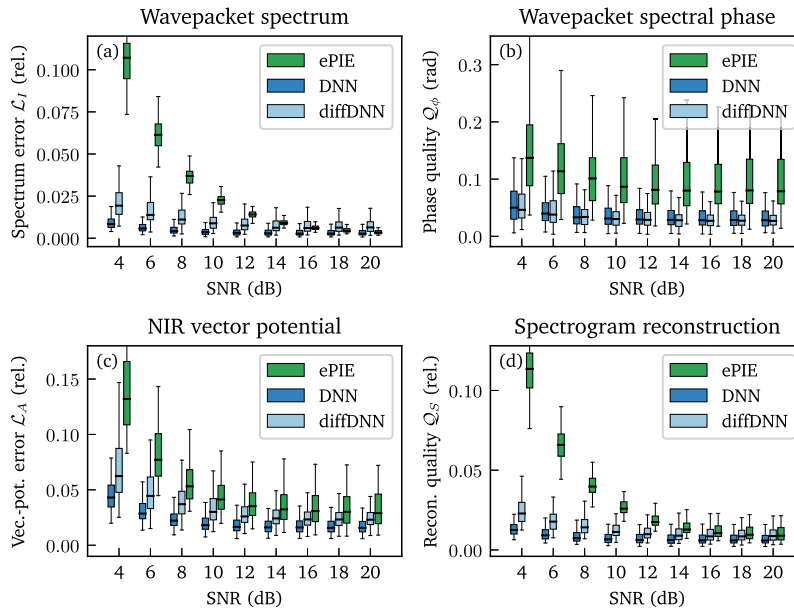
To systematically test the performance of our networks against ePIE we generate a benchmark dataset. For this experiment we draw 128 pairs of NIR laser pulses and electron wavepackets and simulate respective single-trace streaking spectrograms. To track the performance of our retrieval methods dependent on input noise level, we subject the base spectrograms to 9 evenly spaced SNR values over the range 20 dB to 4 dB.

The results of the ensuing benchmark experiment are summarized in Fig. 3. Both deep learning approaches are observed to outperform the ePIE implementation at high noise levels. The predominant advantage of supervised learning seems to be that the neural networks directly learn to associate noisy spectrograms with ground-truth labels, thereby learning to filter out noise. In comparison, ePIE has no mechanism to directly discriminate between streaking signal and noise.

For a direct comparison please also be aware that the inference from our deep networks only takes up to 100 ms per sample whereas the ePIE implementation requires up to minutes to converge. As ePIE can only unambiguously be applied to background-free spectrograms we generated the benchmark dataset without background.



**Fig. 2.** Comparison of different methods for single-trace retrieval. (a) Pseudo-color plot of the simulated test set spectrogram with  $\text{SNR} \approx 12$  dB. (b), (d), (f) NIR vector potential retrievals compared to the label (simulation input). (c), (e), (g) Electron wavepacket retrievals compared to label. Our deep learning approaches better match the ground-truth labels, especially regarding the wavepacket spectral phase.



**Fig. 3.** Single-trace retrieval quality as function of input noise. Box plots depicting the distribution of the respective quality criterion over the evaluated samples concerning: (a) Wavepacket spectrum, (b) Wavepacket spectral phase, (c) NIR vector potential and (d) Spectrogram reconstruction. Lower values are better, showing that our deep learning approaches significantly outperform ePIE at high noise levels.

#### 4.1.3. Application to real-world data

We use a deep ensemble of  $M = 10$  neural networks to reconstruct a streaking trace recorded in an experiment with Helium as a target. The real-world data were acquired at our state-of-the-art attosecond streaking beamline located at the Technical University of Munich. Intense sub 5 fs NIR laser pulses generate isolated XUV pump pulses with central-energy of 105 eV in an amplitude gated high harmonic generation (HHG) scheme. The XUV bandwidth of 5 eV allows for pulse duration below 400 as. The NIR and the XUV pulses are subsequently used in the attosecond streaking experiment.

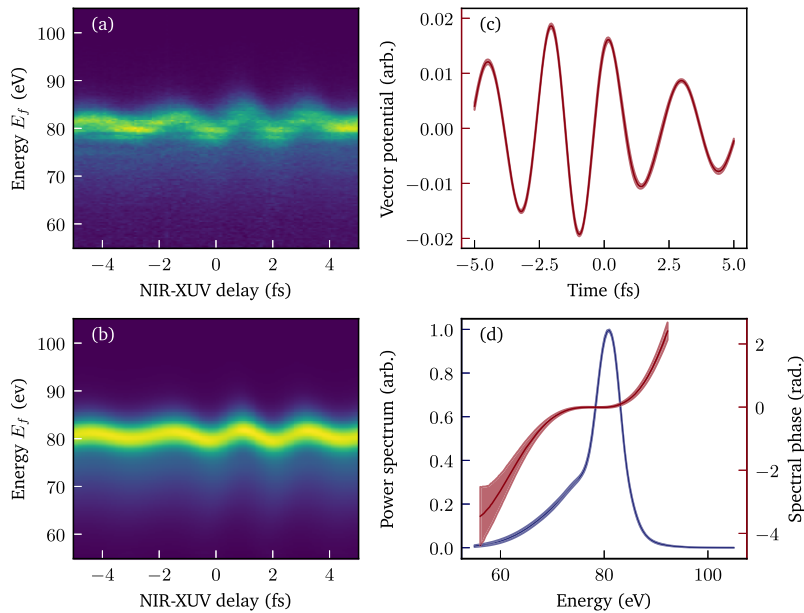
The resulting spectrogram, as well as the deep ensemble retrievals of the NIR vector potential and the wavepacket, are displayed in Fig. 4. We find excellent agreement between the recorded streaking trace and its reconstruction, calculated from the retrieved NIR pulse and electron wavepacket using Eq. (1). Since all networks within the ensemble return well-matching retrievals, the resulting uncertainty estimation is small.

#### 4.2. Photoemission time delay retrieval

Beyond the characterization of electron wavepackets from a single streaking trace, attosecond streaking can be used to determine phase differences between multiple wavepackets streaked simultaneously in the same measurement. By observing photoemission from two distinct energy levels of a species the phase imprint of the XUV pulse cancels out when examining their difference.

Frequently, experiments are primarily aimed at extracting the single scalar value  $d$  describing the central-energy photoemission time delay between two wavepackets. Due to the special interest in this quantity we extract it from our full reconstructions and benchmark its retrieval against respective rTDSE results.





**Fig. 4.** Real-world single-trace retrieval using a deep ensemble. (a) Pseudo-color plot of the experimentally recorded spectrogram with Helium as a target. (b) Pseudo-color plot of the attained spectrogram reconstruction. (c) NIR vector potential retrieval with uncertainty estimate. (d) Electron wavepacket retrieval with uncertainty estimate. We find excellent agreement between recorded and reconstructed streaking trace.

#### 4.2.1. Retrieval example

A showcase for the full dual-trace retrieval using our neural networks is provided in Fig. 5. Panel (a) depicts the pseudo-color plot of a simulated test set spectrogram without background subtraction and  $\text{SNR} \approx 12$  dB. Panel (b) displays a plot of the same spectrogram after application of the differential background subtraction.

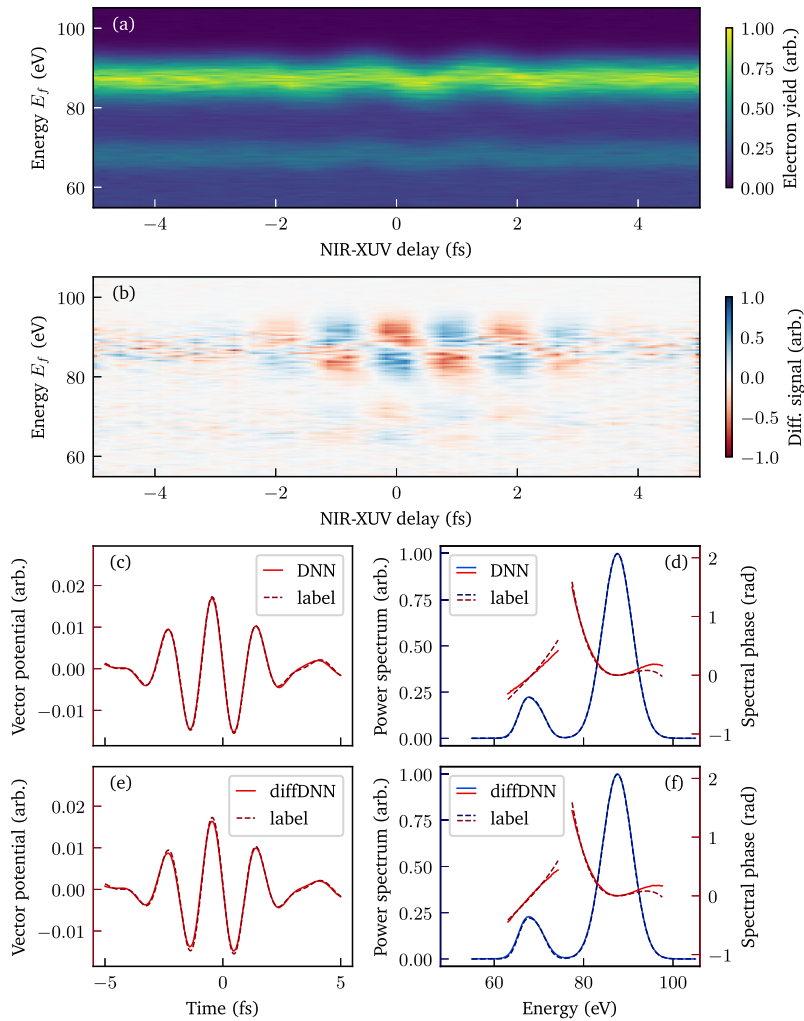
Panels (c) and (e) display the NIR vector potential retrievals of our neural networks compared to the original label. In both settings the retrieved curves exhibit excellent match with the label. In direct comparison the diffDNN using differential background subtraction performs slightly worse.

Panels (d) and (f) display the retrieved wavepacket spectra (blue) and spectral phases (red) compared to their labels. Regarding the spectra, both methods manage close to perfect agreement with the labels. Note that for wavepacket located at lower energies this involves predicting the correct relative amplitude. In contrast the spectrum of the higher energetic wavepacket is always normalized.

Regarding the spectral phase signals we find nice agreement of retrieval and original. Note that the lower energetic wavepacket is allowed a clearly visible first-order phase term conveying the relative photoemission time delay. In contrast the wavepacket at higher energies is set as absolute zero-time reference.

#### 4.2.2. Quality benchmark

To systematically test the performance of our models against rTDSE we generate a benchmark dataset. Therefore, we simulate  $N = 128$  spectrograms with two streaking traces each. All spectrograms are generated with the set relative central-energy time delay  $d = 30$  as between the wavepackets. As we are particularly interested in assessing the retrieval performance dependent



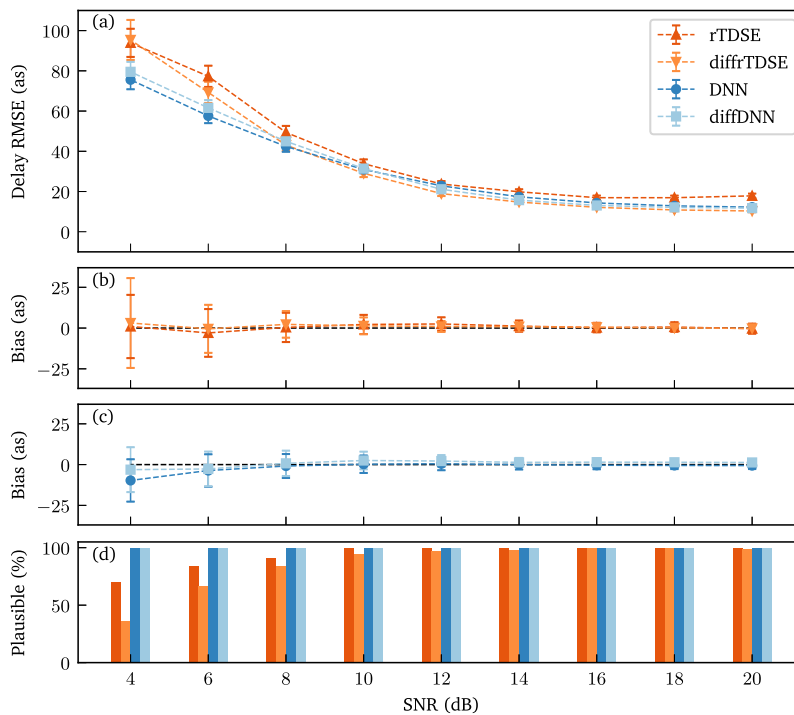
**Fig. 5.** Deep learning approaches for dual-trace retrieval. (a) Pseudo-color plot of the simulated test set spectrogram with  $\text{SNR} \approx 12$  dB. (b) Pseudo-color plot of the spectrogram after differential background subtraction. (c), (e) NIR vector potential retrievals compared to label. (d), (f) Electron wavepacket retrievals compared to labels. Our deep learning approaches nicely match the ground-truth labels, managing to reconstruct a first-order phase term for the lower energetic wavepacket.

on input noise level we subject the base spectrograms to 9 evenly spaced SNR values over the range 20 dB to 4 dB.

With this dataset we mimic the typical experimental approach in attosecond streaking. Frequently one records a set of spectrograms regarding a particular target over multiple measurement days. As laser settings typically have to be readjusted over this time frame, the observed electron wavepackets will vary as well. However, we presume the relative photoemission delay to be a constant population parameter directly tied to the physical properties of the target.

To analyze the accuracy of the retrieved delay predictions we report the root-mean-squared error (RMSE). Additionally it is especially important to us that the predictions cluster around the true delay value. Therefore, we also measure the bias given as the expected difference of prediction and target.

The results of our benchmark are displayed in Fig. 6. Panel (a) summarizes the RMSE of rTDSE and deep learning approaches. Comparing the approaches, our DNN and diffDNN methods behave competitively with respect to both rTDSE methods, and outperform them for SNR values below 8 dB.



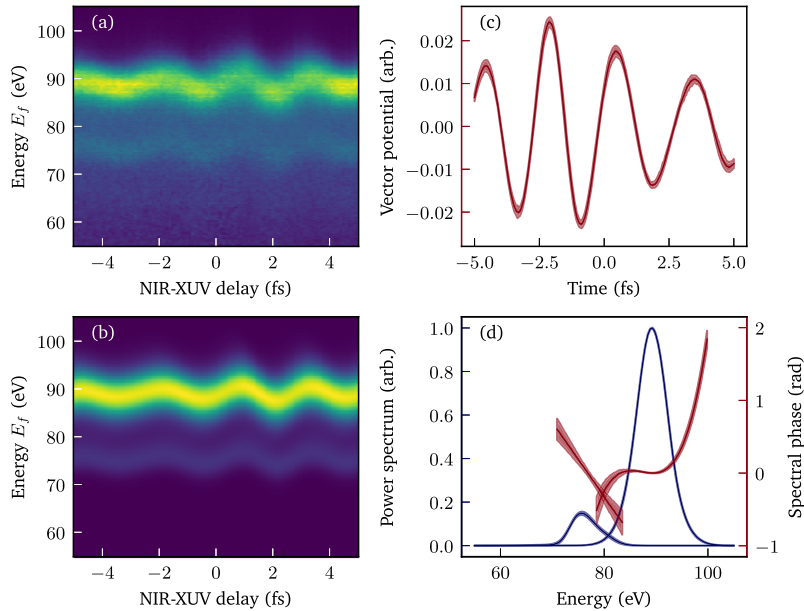
**Fig. 6.** Central-energy photoemission time delay retrieval quality as function of input noise. (a) Time delay retrieval RMSE. (b) Estimated bias of rTDSE methods. (c) Estimated bias of deep learning methods. (d) Percentage of plausible predictions for the respective methods on the benchmark dataset. Our deep learning approaches yield unbiased predictions and competitive RMSE with respect to rTDSE methods.

Panels (b) and (c) show the estimated bias of the rTDSE and deep learning methods, respectively. Neither of the discussed methods shows a significant bias on the benchmark dataset. We note that both rTDSE approaches do not converge on all samples. Additionally we sort out implausible predictions, i.e. delays that are outside the range  $[-0.25, 0.25]$  fs for all methods. The remaining percentage of plausible retrievals for the dataset is displayed in panel (d). Regarding our neural networks all spectrograms yield a plausible photoemission time delay retrieval.

#### 4.2.3. Application to real-world data

An attosecond streaking experiment with Argon as a target was performed with the same pulse parameters as in section 4.1.3. The resulting spectrogram is fed into a deep ensemble of  $M = 10$  neural networks specifically trained for the retrieval of spectrograms with two streaking traces.

As displayed in Fig. 7 it is possible to retrieve the associated spectral phases for the Ar3s and the Ar3p level. From these phases, the photoemission delay for this particular spectrogram can be extracted. However, to gain a statistically significant photoemission time delay value for Argon it will be necessary to collect a larger set of spectrograms.



**Fig. 7.** Real-world dual trace retrieval using a deep ensemble. (a) Pseudo-color plot of the experimentally recorded spectrogram with Argon as a target. (b) Pseudo-color plot of the attained spectrogram reconstruction. (c) NIR vector potential retrieval with uncertainty estimate. (d) Electron wavepacket retrievals with uncertainty estimate. We observe good agreement between recorded and reconstructed spectrogram. More data will be required to extract a statistically significant photoemission time delay value.

## 5. Conclusions and outlook

In this work, we demonstrated the application of deep neural networks for the complete retrieval of near-infrared laser pulses and electron wavepackets from attosecond streak camera experiments. Our deep models are trained in a supervised scheme on large labeled datasets. For the generation of the training data, we numerically simulated attosecond streaking without requiring the central momentum approximation. We successfully applied our networks to real-world data and used deep ensembles to approximate the predictive uncertainty.

Our results show that deep neural networks can be used to fully reconstruct the near-infrared vector potential and electron wavepackets from one or multiple traces in an attosecond streaking spectrogram. In the latter case this allows to extract relative photoemission time delays.

Our networks prove to be competitive or superior in comparison to established algorithms like ePIE and rTDSE, especially when facing noisy data conditions. In combination with the inference time of just several hundred milliseconds per spectrogram, this could pave the way for the evaluation of considerably larger datasets in the future.

The presented implementations are not limited to the attosecond streak camera setup. The RABBIT technique, which is based on the same physical phenomena as the attosecond streak camera, yields data of high complexity, the analysis of which could benefit significantly from the application of neural networks. With the presented methods and results, we hope to motivate further studies towards applying deep learning in other challenging physics applications.

**Funding.** Deutsche Forschungsgemeinschaft (Grant No. 01IS18036B, Munich Center for Machine Learning, SPP 1840 QUTIF); Bundesministerium für Bildung und Forschung (MLwin); European Research Council (ERC-2014-CoG AEDMOS).

**Acknowledgements.** This research was supported by grants from NVIDIA and utilized two NVIDIA Titan V and two Titan RTX GPUs.

**Disclosures.** The authors declare no conflicts of interest.

**Data availability.** Data underlying the results presented in this paper as well as the source code and trained neural networks may be obtained from the authors upon reasonable request.

**Supplemental document.** See [Supplement 1](#) for supporting content.

## References

1. M. Hentschel, R. Kienberger, C. Spielmann, G. A. Reider, N. Milosevic, T. Brabec, P. Corkum, U. Heinzmann, M. Drescher, and F. Krausz, "Attosecond metrology," *Nature* **414**(6863), 509–513 (2001).
2. R. Kienberger, E. Goulielmakis, M. Uiberacker, A. Baltuska, V. Yakovlev, F. Bammer, A. Scrinzi, T. Westerwalbesloh, U. Kleineberg, U. Heinzmann, M. Drescher, and F. Krausz, "Atomic transient recorder," *Nature* **427**(6977), 817–821 (2004).
3. E. Goulielmakis, M. Uiberacker, R. Kienberger, A. Baltuska, V. Yakovlev, A. Scrinzi, T. Westerwalbesloh, U. Kleineberg, U. Heinzmann, M. Drescher, and F. Krausz, "Direct measurement of light waves," *Science* **305**(5688), 1267–1269 (2004).
4. M. Schultze, M. Fieß, N. Karpowicz, J. Gagnon, M. Korbman, M. Hofstetter, S. Neppl, A. L. Cavalieri, Y. Komninos, T. Mercouris, C. A. Nicolaides, R. Pazourek, S. Nagele, J. Feist, J. Burgdörfer, A. M. Azzeer, R. Ernstorfer, R. Kienberger, K. U. E. Goulielmakis, F. Krausz, and V. S. Yakovlev, "Delay in photoemission," *Science* **328**(5986), 1658–1662 (2010).
5. A. L. Cavalieri, N. Müller, T. Uphues, V. S. Yakovlev, A. Baltuska, B. Horvath, B. Schmidt, L. Blümel, R. Holzwarth, S. Hendel, M. Drescher, U. Kleineberg, P. M. Echenique, R. Kienberger, F. Krausz, and U. Heinzmann, "Attosecond spectroscopy in condensed matter," *Nature* **449**(7165), 1029–1032 (2007).
6. M. Ossiander, J. Riemensberger, S. Neppl, M. Mittermair, M. Schäffer, A. Duensing, M. S. Wagner, R. Heider, M. Wurzer, M. Gerl, M. Schnitzenbaumer, J. V. Bart, F. Libisch, C. Lemell, J. Burgdörfer, P. Feulner, and R. Kienberger, "Absolute timing of the photoelectric effect," *Nature* **561**(7723), 374–377 (2018).
7. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, P. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," (2016).
8. R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS workshop*, (2011), CONF.
9. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, (Curran Associates Inc., 2012), p. 1097–1105.
10. C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8), 1915–1929 (2013).
11. C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," (2013).
12. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, (2014), pp. 3104–3112.
13. D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," (2016).
14. 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**(7587), 484–489 (2016).
15. D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of Go without human knowledge," *Nature* **550**(7676), 354–359 (2017).
16. Y. D. Hezaveh, L. P. LeVasseur, and P. J. Marshall, "Fast automated analysis of strong gravitational lenses with convolutional neural networks," *Nature* **548**(7669), 555–557 (2017).

17. T. Ciodaro, D. Deva, J. M. De Seixas, and D. Damazio, "Online particle detection with neural networks based on topological calorimetry information," *J. Phys.: Conf. Ser.* **368**, 012030 (2012).
18. T. Zahavy, A. Dikopoltsev, D. Moss, G. I. Haham, O. Cohen, S. Mannor, and M. Segev, "Deep learning reconstruction of ultrashort pulses," *Optica* **5**(5), 666–673 (2018).
19. S. Kleinert, A. Tajalli, T. Nagy, and U. Morgner, "Rapid phase retrieval of ultrashort pulses from dispersion scan traces using deep neural networks," *Opt. Lett.* **44**(4), 979–982 (2019).
20. J. White and Z. Chang, "Attosecond streaking phase retrieval with neural network," *Opt. Express* **27**(4), 4799–4807 (2019).
21. M. Kitzler, N. Milosevic, A. Scrinzi, F. Krausz, and T. Brabec, "Quantum theory of attosecond xuv pulse measurement by laser dressed photoionization," *Phys. Rev. Lett.* **88**(17), 173904 (2002).
22. V. S. Yakovlev, J. Gagnon, N. Karpowicz, and F. Krausz, "Attosecond streaking enables the measurement of quantum phase," *Phys. Rev. Lett.* **105**(7), 073001 (2010).
23. Y. Mairesse and F. Quéré, "Frequency-resolved optical gating for complete reconstruction of attosecond bursts," *Phys. Rev. A* **71**(1), 011401 (2005).
24. M. Lucchini, M. H. Brüggemann, A. Ludwig, L. Gallmann, U. Keller, and T. Feurer, "Ptychographic reconstruction of attosecond pulses," *Opt. Express* **23**(23), 29502–29513 (2015).
25. S. Neppl, R. Ernstorfer, E. M. Bothschafter, A. L. Cavalieri, D. Menzel, J. V. Barth, F. Krausz, R. Kienberger, and P. Feulner, "Attosecond time-resolved photoemission from core and valence states of magnesium," *Phys. Rev. Lett.* **109**(8), 087401 (2012).
26. J. Riemensberger, S. Neppl, D. Potamianos, M. Schäffer, M. Schnitzenbaumer, M. Ossiander, C. Schröder, A. Guggenmos, U. Kleineberg, D. Menzel, F. Allegretti, J. V. Barth, R. Kienberger, P. Feulner, A. G. Borisov, P. M. Echenique, and A. K. Kazansky, "Attosecond dynamics of *sp*-band photoexcitation," *Phys. Rev. Lett.* **123**(17), 176801 (2019).
27. R. Trebino, *Frequency-Resolved Optical Gating: The Measurement of Ultrashort Laser Pulses: The Measurement of Ultrashort Laser Pulses* (Springer Science & Business Media, 2000).
28. M. Ossiander, "Absolute photoemission timing," Phd thesis, Technische Universität München (2018).
29. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," (2015).
30. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," (2014).
31. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," (2015).
32. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," (2015).