

Artificial neural network as an effective tool to calculate parameters of positron annihilation lifetime spectra

Cite as: J. Appl. Phys. 134, 114902 (2023); doi: 10.1063/5.0155987

Submitted: 25 April 2023 · Accepted: 28 August 2023 ·

Published Online: 15 September 2023



M. Pietrow^{1,a)} and A. Miaskowski²

AFFILIATIONS

¹Institute of Physics, M. Curie-Skłodowska University, Pl. M. Curie-Skłodowskiej 1, 20-031 Lublin, Poland

²Faculty of Production Engineering, University of Life Sciences, Akademicka 13, 20-950 Lublin, Poland

^{a)}Author to whom correspondence should be addressed: marek.pietrow@umcs.pl

ABSTRACT

The paper presents the application of the multi-layer perceptron regressor model for predicting the parameters of positron annihilation lifetime spectra using the example of alkanes in the solid phase. Good agreement of calculation results was found when the approach is compared with the commonly used methods, e.g., LT. The presented method can be used as an alternative quick and accurate tool for the decomposition of positron annihilation lifetime spectroscopy (PALS) spectra in general. The advantages and disadvantages of this new method are discussed. We show the preliminary results where the trained network can give better outcomes than the results yielded by programs based on an analysis of a single PALS spectrum.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0155987>

I. INTRODUCTION

Positron annihilation lifetime spectroscopy (PALS) is one of the useful experimental methods using positrons for studying structural details in a wide spectrum of materials, in particular, in the solid state.^{1,2} This method is based on the annihilation of positrons where their lifetime and annihilation intensity in the sample is dependent on some properties of the material in the nano scale, including local electron density, bound electron energy, and the density and size of free volumes in the sample.

Depending on the material, besides the process of direct annihilation, a positron can form a metastable atomic state with an electron, called a positronium (Ps), which can exist in two spin states referred to as *para*- and *ortho*-Ps, which have different properties (especially, their lifetimes differ in vacuum by three orders of magnitude).³ The theory of Ps formation in solid matter is described by the *blob* model.⁴ A number of conditions must be met for the Ps to be formed in matter. As for the localized Ps in the molecular crystals of alkanes discussed here, one of the conditions is the presence of free volumes of a sufficiently large size.⁵ A Ps is extremely useful in materials science since its lifetime can be related to the size of

free volumes in the material. Depending on the structure of the sample, there is possibly a variety of Ps components which annihilate with characteristic lifetimes. All these populations give their own account to the positron annihilation spectrum measured experimentally. PALS spectra require decomposition in the post-measuring procedure, resulting in both the calculation of lifetimes for particular fractions of positrons and their relative account (intensity) in the PALS spectrum (so-called spectrum inversion problem).⁶

Many algorithms used for data processing require assuming an exponential character of positron decay. They also require fixing the number of components used during the decomposition. For example, the method used by adequate software, e.g., the LT program⁷ or PALSfit,⁸ consists in fitting the PALS experimental spectrum to a sum of a given number of exponential functions usually convoluted with the (multi) gaussian apparatus resolution curve.

The PALS spectra used here were measured for normal alkanes (*n*-alkanes), i.e., the simplest organic molecules where carbon atoms form a straight chain of the molecule and are saturated by hydrogen atoms. The *n*-alkanes with a different number *n*

17 September 2023 05:24:06

of carbon atoms in the molecule form a homologous series described by the general chemical formula C_nH_{2n+2} (C_n is used as an abbreviation). Alkanes in the solid phase form molecular crystals where the trains of elongated molecules are separated by gaps called inter-lamellar gaps—Fig. 1. Besides the straight chains of molecules (so-called *all-trans* form), other conformations, such as *end-gauche*, *double-gauche*, and *kink* are observed at higher temperatures approaching the melting temperature. All these spaces are the sites where the localized Ps is formed.⁵ Using the PALS technique, the size of these free volumes can be determined from the lifetime and the relation between both given by the Tao–Eldrup formula or its modification.⁹ Due to the large polarizability of molecules, the quasi-free electrons excited by energetic positrons can induce local shallow traps and locate there.^{1,10} These electrons take a part in Ps formation and they are responsible for the Ps annihilation intensity increasing in time observed during PALS experiments with alkanes.¹¹ As demonstrated by our previous analysis of alkanes carried out with the use of the PALS technique, the best results of the spectrum decomposition are achieved assuming only one population of *ortho*- and *para*-Ps, whereas the ratio of the *ortho* to *para* intensity is fixed at 3/1.

Tools of machine learning, e.g., genetic algorithms or artificial neural networks, have been used to perform numerical calculations in a variety of aspects in positron science.^{12–16} They have also been used for unfolding the lifetimes and intensities from PALS spectra.^{17–20} Possibly due to the low computing power of hardware and the low time resolution of PALS spectrometers at the time when the neural network algorithms for decomposition of PALS spectra were proposed, most of the spectra used in these calculations are simulated by software but not measured directly. For the same

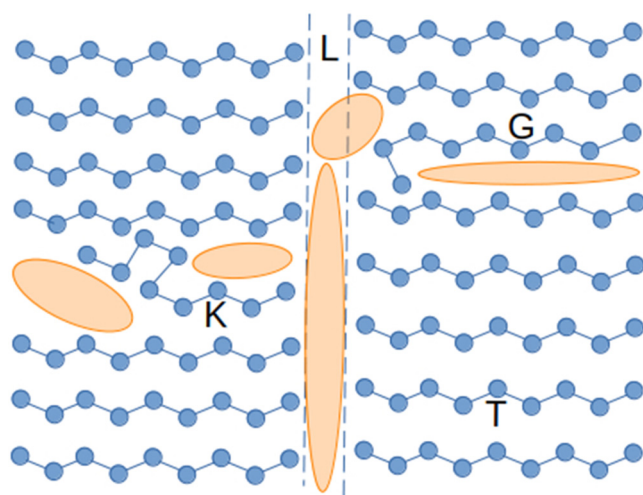


FIG. 1. Schematic view of the typical structure of a solid alkane. Round circles denote carbon atoms forming the zig-zag backbone of molecules. The rows of these chain molecules are separated by a so-called inter-lamellar gap (L). Most molecules form straight chains (*all-trans*) – T, but there are other possible conformations, i.e., *end-gauche* (G) or *kink* (K). The arrangement of the molecules generates spaces where Ps can be formed (marked as ellipses).

reason, the neural network architecture used there does not allow changing parameters to an extent allowed by algorithms developed today. Furthermore, no procedure allowing application for the same calculations of spectra registered for different time constants per channel has been presented since then. Thus, the preferred software used for spectrum decomposition is still based on non-linear fitting algorithms which do not include a possibility of establishing the result based on a multi-spectra set at the same time.

Here, we present an approach to analysis of PALS spectra based on the multi-layer perceptron (MLP) model, which is one of the tools of machine learning.²¹ The model assumes a network of inter-connected neurons grouped in the input layer (In_i), the hidden neurons layers (h_i^k) and the output neurons layer (Out_i), where i goes over the neurons in a given layer and k numbers the hidden layers. A graphical diagram of the network used is shown in Fig. 2. The numbers of In and Out neurons are determined by the amount of the independent input data introduced to the network and the data yielded by calculation in a given problem, respectively. The number of hidden layers and the number of neurons within these layers are set experimentally to optimize the network to give required results. To each layer (excluding the output layer), one bias neuron is attached for technical reasons.²² The tool assumes the learning process first, where the In neurons are fed with the data for which the result of the Out neurons is known in advance. During this process, the weight coefficients for pairs of inter-connected neurons are adjusted by an algorithm, so that the MLP output can give results with the greatest similarity to the expected ones. The MLP is trained after a number of iterations of training. Once the MLP results of learning are satisfactory, the MLP can be used to calculate the output for input data that have never been used in the training process.

The MLP type of network can be applied to solve the problem of both classification and regression. For the first group of problems, it is required from the MLP to ascribe the values of the output parameters in the form of well separated categories. These so-called *labels* can always be parameterized by a discrete set of numbers. The problem described in this paper is classified as rather a regression problem (MLPR) where the values of the output at each Out neuron are characterized by a continuous set of values. Consequently, the output may contain values approaching these appearing during the learning process but may not necessarily be exactly the same. The internal algorithms of the MLPR allow regarding the learning process as a way of finding the quasi-continuous output function of input parameters. In our case, based on the data from the PALS spectra applied as the input values of the perceptron, the MLPR is used for solving the regression problem of finding the values of key PALS parameters on the output.

The main goal of this work is to show that the trained neural network applied to PALS spectra can give compatible results to those produced by LT, which is used here as a learning tool. As a consequence, the trained network can be used as an autonomous tool returning PALS parameters for new spectra of compounds met in the training process. The quality of prediction of the network increases with the number of cases used during training. Since the network calculates the results based on a large amount of collected information, its ability to give more adequate results may overcome the quality of programs which calculate the PALS parameters based on the analytical method

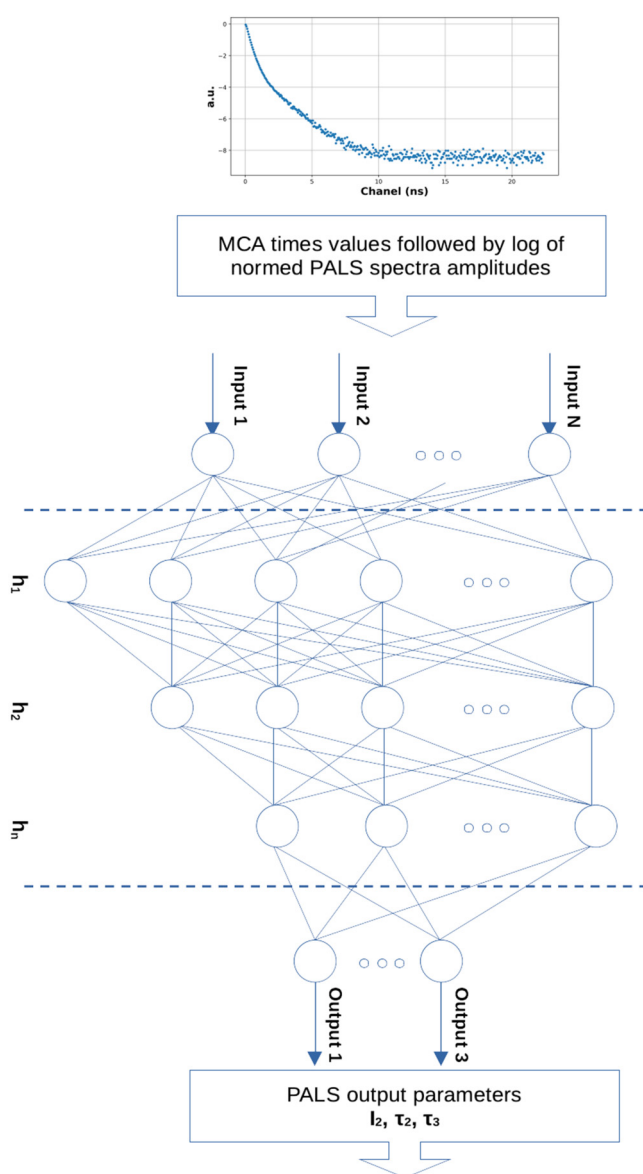


FIG. 2. Schematic view of the MLP applied. The PALS data from consecutive channels of the MCA are transferred as the amplitudes of consecutive input neurons $Input_i$. h_i denote neurons in the i -th hidden layer, whereas $Output_i$ denote output neurons returning chosen PALS decomposition parameters.

applied to one particular spectrum. Although an example is shown here that the presented network can give better results than those provided by LT, a comparable meticulous study of the results from both approaches in a wide range of possibilities deserves separate work.

II. METHOD

The *scikit-learn* library was used to estimate the PALS parameters for alkanes.²³ In our case, the MLP regressor class (called

MLPRegressor), which belongs to one of the supervised neural network models, was implemented. In this class, the output is a set of continuous values. It uses the square error as the loss function. This model optimizes the squared error using the Broyden–Fletcher–Goldfarb–Shanno algorithm (LBFGS),²⁴ which is one of the quasi-Newton methods. Some *MLPRegressor* parameters playing a key role are mentioned below. Their values require to be tuned, especially the α hyper-parameter, which helps in avoiding over-fitting by penalizing weights with large magnitudes. A full list of the *MLPRegressor* parameters is defined in Ref. 25.

In the learning process, we used spectra of several alkanes (in the range of C_6 – C_{40}) measured at several temperatures (-142 – 100°C) and collected for years from an analog spectrometer. Irrespective of both the goal of the particular experiment and the length of the alkane chain used as a sample, the initial assumptions made for starting the analysis of the spectra made by the LT program⁷ were the same. Each measurement resulting in spectra to be used was performed using a sample prepared in a similar way, i.e., the sample was degassed, and the rate of cooling or heating was the same. In each case, the measurement was made at constant temperature for at least 1 h, which gave some hundreds of thousands of annihilations (the strength of the radioactive source was similar in each case). The temperature during some experiments was changed step-wise but each spectrum was collected at constant temperature. The most important issue here is that the post-experimental analysis of the spectra was always conducted under the same general assumptions. Especially, for the decomposition of these spectra, we used LT, supposing that the time resolution curve can be approximated by one-gaussian curve. Each time it was assumed that the annihilation process in the Kapton envelope accounted for 10% (so-called *source correction*). Additionally, only one component was always assumed for *para*- and *ortho*-Ps, whereas their intensity ratio was fixed at the value of 3/1 (see Ref. 26 for details of the experimental procedure).

Considering these assumptions, each spectrum was decomposed into three exponential curves, for which the intensities (I) and lifetimes (τ) were calculated for the following sub-populations of positrons: free positron annihilation (I_2 , τ_2), *para* (I_1 , τ_1), and *ortho*-Ps (I_3 , τ_3) (numbering of the indices is related to the length of τ . The increasing values of the indices correspond to the rising length of lifetime.). The database collected in this way contained 7973 PALS spectra, wherein about 75% were used in the neural network training process and the rest were used as a testing set for checking the accuracy of the results given by the learned network.

The number of input neurons is determined by the number of channels of the multi-channel analyser (MCA) module of the PALS spectrometer recording PALS spectra. Furthermore, the number of the output neurons in this model is related to the number of PALS parameters, which are supposed to be predicted for further studies of physical processes occurring in the sample. The decomposition of the PALS spectrum made by commonly used programs, like LT, allows determining (I, τ) pairs for all assumed components of a given spectrum. However, not all these parameters are often needed for further analysis. Furthermore, some of these parameters are inter-dependent. For example, in the case of PALS spectra for the alkanes discussed here, the spectrum is assumed to be built up by events from the three populations of

17 September 2023 05:24:06

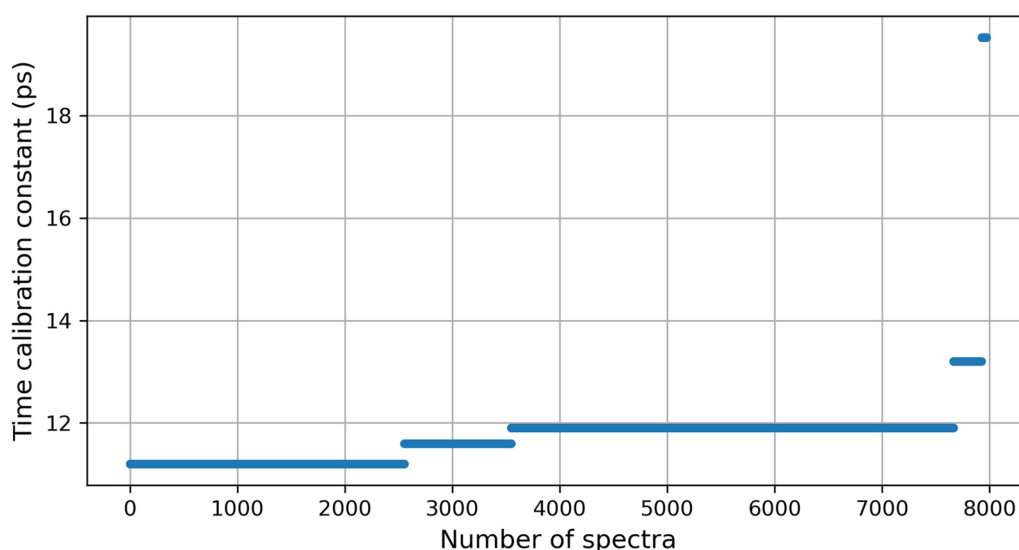


FIG. 3. Number of spectra (horizontal axis) with a given value of the time constant per channel Δ (vertical axis) used as a data set in the presented calculations.

positrons mentioned above ($\tau_1 - \tau_3$, $I_1 - I_3$ parameters). However, from the practical viewpoint, only τ_2 , I_2 , τ_3 , and I_3 are then used for studying physical processes and the structure of the sample. Furthermore, in this case, I_i are inter-dependent and fulfill the relations $I_1 + I_2 + I_3 = 100\%$ (Annihilation in Kapton was subtracted in advance.) and $I_3/I_1 = 3$. Thus, in fact, only I_2 , τ_2 , and τ_3 are considered as the Out parameters of MLPR. Hence, in our modeling, we declared only three output neurons for receiving values of these three parameters.

III. PREPARATION OF INPUT AND OUTPUT DATA

During the PALS measurements, the time constant per channel (Δ) varied, depending on the internal properties and settings of the spectrometer. Most of the data used here were collected with $\Delta = 11.9$ ps; however, some spectra were measured with $\Delta = 11.2$ ps, 13.2 ps, 11.6 ps, and 19.5 ps (Fig. 3). Therefore, it is important for the In neurons to code the PALS amplitude samples not in relation to the channel numbers but in the scale of time. Hence, in addition to the spectrum amplitudes, the regressor has to learn the times associated with these amplitudes. Thus, one half of the In neurons is fed with time values for consecutive channels of a spectrum, whereas the second half is fed with the values of their amplitude. The advantage of the regression approach applied here is the ability to test spectra measured even for a time sequence that has never appeared in an extreme case in the training process.

This method requires setting a common zero-time for each spectrum correctly. To achieve this, the original data from the left slope of the spectrum peak (and only a few points to its right) were used to interpolate the resolution curve, which is assumed to be in the gaussian form. The position of this peak defines a zero-time for a spectrum. One-gaussian interpolation is compatible with previous LT analysis assumptions. Based on the common starting position

for all spectra established in this way, the values of time for each channel on the right to the peak were re-calibrated for each spectrum depending on Δ for which the spectrum was measured. The definition of zero-time assumed here is, to some extent, independent of the zero-time definition in analytical programs such as LT. Here, the zero-time is understood in the context of choosing by definition and consistently the first channel taken into further consideration. Finally, for further analysis, we took the same N number of consecutive channels for each spectrum on the right to its peak (points p_i in Fig. 4). The δ parameter shown in Fig. 4 denotes the distance (in time units) between the first point on the right to the peak and the calculated time position of the peak. The N number taken for further analysis was established experimentally. Finally, the spectrum data for the MLPR input are the N points p_i with their two values: the re-calibrated number of counts in a given channel (see below) and their re-calibrated times of annihilation.

Then, to minimize errors, the original input data [Fig. 5(a)] were transformed before application. Each original spectrum was stored in 8192 channels of MCA. First, starting from the first channel on the right to the spectrum maximum (p_1 in Fig. 4), 2 k channels were taken from the original spectrum. This means that the spectra were truncated at about 25 ns of the registration time (varying to some extent, depending on the Δ for a given spectrum). Second, to smooth random fluctuations, the data were smoothed in most cases. One of the examples of smoothing is averaging over five consecutive channels. In this case, the number of samples in each spectrum shrank from the original 2k channels to the amount of 400. Since the In neurons transfer information about the pair of values—times (t -part) and amplitudes (A -part), 800 input neurons that fed the MLPR with the data in this case were declared. Third, to standardize the range of the input data values, the set of the PALS amplitudes was normalized to the maximum value of the amplitude and then logarithmized. According to these

17 September 2023 05:24:06

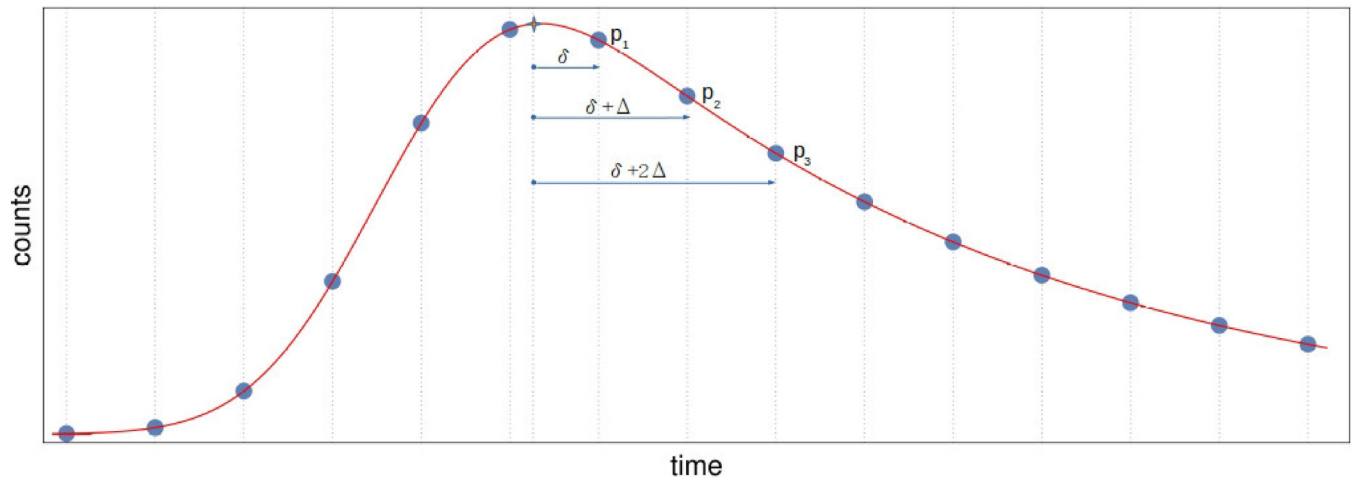


FIG. 4. Schematic view of a peak region of the PALS spectrum. The bullets indicate $(t, \log(A))$ pairs saved in the MCA channels, whereas the star indicates the position of a peak calculated assuming a gaussian shape of the apparatus distribution function. Only the points to the right of the star (p_1, p_2, \dots) are taken as data introduced to MLPR. The δ parameter denotes the time distance between the calculated peak and the first point, whereas Δ is the time distance between two points.

transformations, the A -part data covered the numerical range $[-9, 0]$ —Fig. 5(b). Furthermore, to adjust the range of the values in the t -sector, the values of time were divided by -2.5 . As a result, all data transferred to the In neurons were in the range of approximately $[-10, 0]$.

Additionally, we applied some transformation of the original values for the Out neurons in order to have their values at each neuron scaled to the same range. The first output neuron is related to I_2 , while the value range of I_2 is typically tens (in % units). The second neuron transfers the information related to τ_2 whose

original values are of the order of 0.1 (of ns), whereas the order of τ_3 related to the third neuron is originally 1 (of ns). In order to have a uniform order of numerical values on all Out neurons, the finally fed data are $[I_2/10, \tau_2 \times 10, \tau_3]$.

A criterion of acceptance of training the network was the best value of the score validation function defined for this regressor as

$$S = 1 - \frac{\sum_N (\mathcal{O}_{\text{true}} - \mathcal{O}_{\text{pred}})^2}{\sum_N (\mathcal{O}_{\text{true}} - \langle \mathcal{O}_{\text{true}} \rangle)^2}, \quad (1)$$

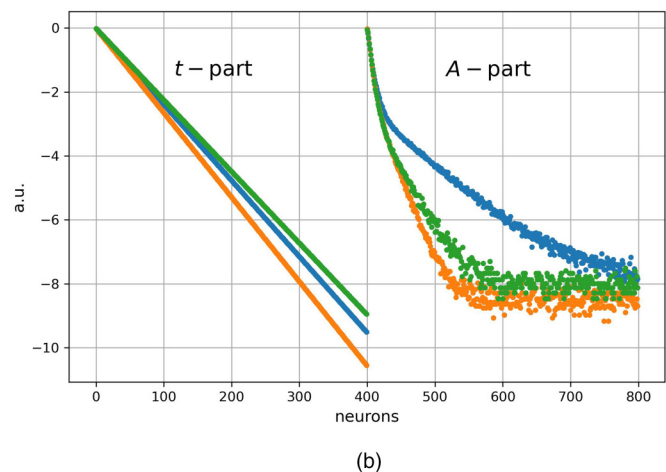
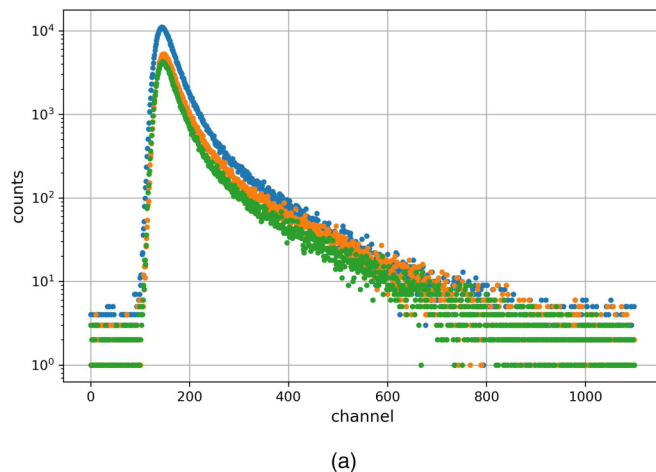


FIG. 5. Randomly chosen three “raw” PALS spectra truncated to the region of the peak—(a), and their transformed form—(b), according to the procedure described in Sec. III producing data directed to In neurons. The t -part denotes a set of time values for points p_1, p_2, \dots (see Fig. 4), while the A -part codes the log function of their normalized amplitudes. In special cases, these data are smoothed or compressed before use in MLPR (see Sec. IV).

TABLE I. Values of the MLPRegressor parameters applied for producing the final MLPR results.

Parameter	Value
hidden_layer_sizes =	7×150
activation =	<i>relu</i>
solver =	<i>lbfgs</i>
alpha =	0.01
learning_rate =	<i>invscaling</i>
power_t =	0.5
max_iter =	5×10^9
random_state =	None
tol =	0.0001
warm_start =	True
max_fun =	15 000

where $\mathcal{O}_{\text{true}}$ and $\mathcal{O}_{\text{pred}}$ denote expected (known) and calculated (predicted) values of the result, respectively, whereas $\langle \dots \rangle$ denotes a mean value.^{23,25} S is calculated for both the learning and testing sets separately. N denotes the number of spectra in the trained or tested set. The optimum value of S is ≈ 1 .

IV. RESULTS

The MLPRegressor used in these calculations requires establishing some key parameters²⁵ that influence the ability to learn and the speed of the learning process. We performed some tests trying to optimize these parameters. The best results were obtained by the settings shown in Table I. Both the names and the meaning of the technical parameters shown in the table are identical to those defined in the routine description.²⁵ Once the key parameters of the MLPR were established (especially the solver), we performed tests of the network credibility by changing the number of hidden layers, the number of neurons within the layers (hidden_layer_sizes parameter), and the alpha parameter. The results in Table II show examples of the results. For these networks, we specified the mean validation score parameter for both the training $\langle S_{tr} \rangle$ and testing $\langle S_{te} \rangle$ sets separately with their variation δS . Averaging was made over the results of ten runs of the training process for identical networks differing by initially random weights. We did not notice any

rule giving a ratio of the numbers of neurons that should be declared in the consecutive hidden layers (especially as the number of neurons should decrease proportionally in the consecutive layers). A few initial examples shown here suggest that the accuracy of results increases when both the number of hidden layers and the number of neurons inside increase. However, the last two rows of the table show that a further increase in these parameters does not give better results. Finally, the network that gave a nearly best result was chosen (marked in bold $\langle S_{tr} \rangle$ in the table). It was checked for this network that an increase in the iterations of training (max_iter parameter) beyond about 5×10^9 did not improve $\langle S \rangle$.

For several finally tested networks, the spectrum of the magnitude of inter-neuron weights was checked. It is expected that weights that differ significantly from the average range of values may affect the stability of the results. In this case, the range of weight values seems to be quite narrow. As shown in Fig. 6, the weight magnitude order (exponent of weights) for the chosen network ranges from 10^{-5} to 10^0 , while the relative number of cases in these subsets changes exponentially. The lack of values outside the narrow set of values suggests that self-cleaning of the resultant weights is performed by the MLPRegressor algorithm itself.

The number of all PALS spectra used as a database for the network was 7973, and 6500 were used to learn the output values (training set) by the network, while the rest were used for checking the results of learning (testing set). Table III shows a few examples of randomly taken results given by one of the networks used finally. The results given by the trained network were compared to the expected values known from the LT analysis. Although S for both the trained and tested sets in this case is not the highest of those obtained in our tests, the result of the use of this network is satisfactory in a practical sense because the deviation of the predicted and expected result is in the range of the deviation given by LT itself. Noteworthy, the standard deviation calculated for our model (given in the caption of Table III) should be understood in relation to LT values (expected values) which are regarded as referential.

The problem of pre-preparation of spectra for calculations by MLPR is worth mentioning. The main problems are the questions of where the spectrum should be cut and to what extent it is acceptable to smooth the spectra by averaging their consecutive

TABLE II. Valuation score for chosen values of some MLPR parameters. S values are averages over 10 runs with random initial neuron weights. A nearly optimum case of parameters is placed in a row with S marked in bold.

hidden_layer_sizes	max_iter	alpha	$\langle S_{tr} \rangle$	δS_{tr}	$\langle S_{te} \rangle$	δS_{te}
$30 \times 25 \times 15$	10^6	0.7	0.950	0.003	0.942	0.004
3×100	10^8	0.7	0.969	0.005	0.965	0.008
3×100	10^8	0.1	0.974	0.005	0.968	0.008
3×100	5×10^8	0.1	0.975	0.003	0.976	0.003
4×100	10^8	0.1	0.978	0.004	0.975	0.006
$500 \times 400 \times 300 \times 200$	5×10^9	0.01	0.977	0.003	0.974	0.007
7×150	5×10^8	0.01	0.985	0.002	0.975	0.013
$500 \times 500 \times 400 \times 400 \times 300 \times 300 \times 200 \times 200$	5×10^9	0.01	0.978	0.005	0.977	0.008
8×500	5×10^9	0.01	0.982	0.004	0.980	0.005

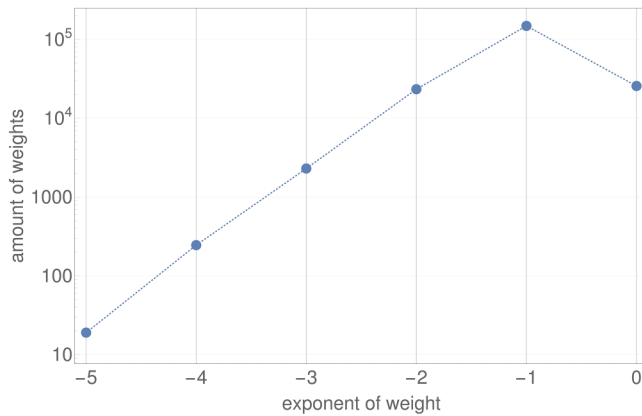


FIG. 6. Number of cases (log scale) of exponents of weights for a network with 7×100 hidden layers. For this network, the key parameters are: solver=*lbfgs*, max_iter= 5×10^{11} , alpha=0.005, learning_rate=*invscaling*, and activation=*relu*.

values. As for the first problem, using a series of runs for which the spectra were cut at another point than the aforementioned limit of 2 k channels, it was determined that this number of channels was almost the best choice. $\langle S \rangle$ was found to worsen in the case of a shorter cut (e.g., 1.5 k channels) and did not improve significantly in the case of the longer ones while it took longer to compute the result because of an increase in the number of In neurons (e.g., 3 k channels).

The accuracy of the prediction increases when the learning and testing processes are limited to one only Δ with all parameters of the network kept constant. In this case, the set of values in the t -part for every spectrum varies in a much narrower range (only δ changes). In this case, the training process is more effective even if the size of the training set is reduced. To show this, we separated the set of spectra measured for only one $\Delta = 11.9$ ps.

Consequently, the whole set of samples under consideration shrank to 4116, and 3000 of them were used for training after the transformations described above. The score S obtained in this case was much greater than S for an identical network applied to spectra with all possible Δ . The comparison of these two cases is shown in Table IV (the last row of the table). Here, to have a reliable result for networks with different (random) initial weights, the score was averaged over 30 runs.

The statistical changes in the value from point to point of the input spectrum do add to the error of the result. Thus, although data smoothing reduces to some extent the information given by the PALS spectrum, the reasonable scaling should not diminish the quality of approximation while the size of the network could be reduced. The validation score S is sensitive to smoothing the spectrum. In Table IV, two cases are compared where each 3- and 5-tuples of points of the spectrum (forming non-overlapping windows) were taken to calculate their average amplitude. For example, $N = 3500$ points of the initial spectrum are reduced to 700 points when averaging over $k = 5$ points; when the remainder of the division of N by k is not zero, an integer quotient is taken. $\langle S \rangle$ calculated for these two cases shows that both of them give the same results statistically. However, the further shrinking of the spectrum by setting $k = 6$ or more produces worse S . To give an example of translation of the differences in the score into the differences in the expected and calculated values, for the testing score of 0.96 and 0.97, the mean square error for all output data is 1.6% and 1.2%, respectively.

Table IV also shows the S parameter when the moving average is applied during preparation of spectra. The sampling window applied here is 10. The comparison of this result to the result of calculation with unsmoothed data shows that the application of the moving average does improve predictions for the testing data set.

The quality of prediction of the network decreases if the region of the values of training parameters differs from the region of expected values. However, the regression algorithm used here gives the opportunity to use the trained network for a wider range of spectra than that used for training. Accordingly, the network

TABLE III. Examples of a few randomly taken results of calculations (prediction) of the I_2 , τ_2 , and τ_3 parameters compared to the expected values calculated by LT. Here, hidden_layer_size = 7×150 , $S = 0.985$ for both training and testing sets. Standard deviations calculated by LT for the expected values are $\sigma_{I_2}^{\text{exp}} = 0.3$, $\sigma_{\tau_2}^{\text{exp}} = 0.04$, and $\sigma_{\tau_3}^{\text{exp}} = 0.1$ for I_2 , τ_2 , and τ_3 , respectively. SD calculated for the values predicted by this network are 0.02, 0.04, and 0.04, respectively.

Example	I_2 (%)		τ_2 (ns)		τ_3 (ns)	
	Expected	Predicted	Expected	Predicted	Expected	Predicted
1	68.0	68.8	0.27	0.28	1.21	1.20
2	47.2	47.6	0.38	0.39	3.23	3.18
3	65.4	65.5	0.30	0.29	1.35	1.38
4	78.3	78.3	0.23	0.24	1.11	1.12
5	52.4	52.4	0.35	0.34	2.93	2.93
6	60.5	60.4	0.31	0.30	1.25	1.20
7	59.3	58.8	0.29	0.29	1.19	1.22
8	61.0	62.3	0.30	0.31	1.91	1.91
9	70.2	69.5	0.21	0.21	1.06	1.10
10	39.9	38.8	0.23	0.22	1.15	1.13

17 September 2023 05:24:06

TABLE IV. Comparison of MLPR validation score S for different formats of the input data. Comparison of the results for “raw” data (log of normalized and adjusted data according to the procedure described in Sec. III) and data on which the moving average and compressing average (by each 3 and 5 separate spectrum points) are applied. The result for the network fed with the data collected for one chosen Δ is added in the last row of the table. $\langle S_{tr} \rangle$ and $\langle S_{te} \rangle$ denote the mean score for training and testing, respectively.

MLPR and spectra parameters			$\langle S_{tr} \rangle$	$\langle S_{te} \rangle$
solver= <i>lbfgs</i> activation= <i>relu</i> learning_rate= <i>invscaling</i> alpha = 0.01, In = 800 hidden_layer_sizes = 7×150 max_iter = 5×10^9 Averaged over 30 trials	Several Δ s	Unsmoothed data	0.984 ± 0.005	0.963 ± 0.006
	$N_{tr} = 6500$ (~80%)	Moving average	0.984 ± 0.005	0.977 ± 0.007
	$N_{te} = 1473$	k = 3	0.981 ± 0.005	0.976 ± 0.007
		k=5	0.981 ± 0.006	0.974 ± 0.010
	Fixed $\Delta = 11.9$ ps			
	$N_{tr} = 3000$ (~73%)	k=5	0.993 ± 0.002	0.989 ± 0.003
	$N_{te} = 1116$			

trained on alkanes can be applied for some derivatives of alkanes, e.g., alcohols, or complex mixtures containing alkanes, such as waxes. Our experience with these compounds shows that although the physical and chemical properties of the compounds differ substantially in this case, the number of PALS components which give the best fitting remains the same even though the trapped electrons mentioned in the *Introduction* are absent in alcohols, while the presence of polar groups changes the energetic conditions for the formation of Ps. In the case of the network described in Table IV ($k = 3$) with the training score of 0.990 (based on alkanes), a score 0.959 was obtained for testing a set of 20 randomly taken PALS spectra for alcohols.

One of the advantages of the use of the artificial neural network is its ability to estimate the output values for an unknown spectrum based on the larger amount of information taught during the training process. It is expected that the well trained network should exceed the quality of results compared to the program based on a single spectrum only. It would be especially useful in situations where spectra are distorted in some way (e.g., due to technical problems of the apparatus). Although the detailed comparison of the results from the presented model and from LT on a wide set of spectra modifications require a separate study, to simulate certain spectral disturbances, we took randomly ten spectra from the set of all those used for training and testing and modified them manually by changing the number of counts in randomly chosen channels or groups of channels. The region of modification was either the left or right slope of the peak in its vicinity. The number of counts in a given 10–20 channel was increased or decreased by about one order of magnitude. For these spectra, we calculated the PALS parameters both in the LT program and with the use of the trained MLPR regressor. To compare the results, we calculated the mean relative error in relation to the expected values for undisturbed spectra. The results are as follows:

$$\begin{aligned}
 \eta_{I_2}^{(LT)} &= 0.061, & \eta_{I_2}^{(MLPR)} &= 0.033, \\
 \eta_{\tau_2}^{(LT)} &= 0.055, & \eta_{\tau_2}^{(MLPR)} &= 0.047, \\
 \eta_{\tau_3}^{(LT)} &= 0.042, & \eta_{\tau_3}^{(MLPR)} &= 0.019,
 \end{aligned} \quad (2)$$

for the network with the score of 0.97. This result shows that the approximation made by the MLPR regressor is better than that made with the use of LT for such disturbed spectra.

V. CONCLUSIONS

We have shown in this paper that the easy-to-reach machine learning MLPR regressor tool enhanced with some programming in *Python* making some preparation of data can be used as an alternative method of solving the problem of inversion of PALS spectra. The main disadvantage of the presented method is the need for the decomposition of training spectra by other softwares to attain the Out values for training. Once the training set is collected and the network is trained, the algorithm works very quickly, giving the result for tested spectra that have never been used in the training process. The need for using another program, e.g., LT, may be regarded as a disadvantage in the use of the neural network for studies of PALS spectra. However, a support of such a program is necessary in the training phase only when training output data are needed. The trained network functions as an autonomous software that can be re-trained in the course of new data collection. The training process used here is based on results given by LT, i.e., a method producing results with some uncertainty itself. The uncertainty associated with the LT approach is caused by the use of numerical methods to compute the fit, in particular, cases. On the other hand, since the MLPR prediction is based on information from a large set of spectra, this approach seems to be less sensitive to the specific shape of a given spectrum and may be more accurate in predicting parameters. Furthermore, the presented method seems to be faster than the referenced ones, since calculations made by a trained network are reduced to simple transformations of matrices and vectors, which are not demanding computationally and less sensitive to numerical problems.

Although the model presented here is similar to that described in Ref. 18 (and repeated in Ref. 19), there are significant differences indicated in Table V. Our experimental data are collected by spectrometers differing in functional properties, especially differing in their time resolution. Even for one spectrometer, this parameter should be re-calibrated periodically due to changes in experimental conditions, especially temperature. In the algorithm presented in Ref. 18, the same resolution curve for all spectra is assumed. In our data preparation procedure, the parameters of the resolution curve are interpolated for each case. Based on this, the δ parameter is calculated and the value of the shift in time is established for consecutive channels. Although the one-gaussian resolution curve was assumed here, it is possible to extend this algorithm for much

TABLE V. Comparison of key parameters and results of the MLPR modeling applied in this study (*skLearn*) and a three-component spectrum analysis published previously (presented in Refs. 18 and 19).

	Pázsit ¹⁸	An ¹⁹	skLearn
Type of training spectra	Simulated	Simulated	Real (alkanes)
No. of training spectra	575	920	7973
Type of spectra tested	Simulated	Simulated, silicon	Alkanes
No. of test spectra	50	100 (30)	1473
Type of network	One-layer perc.	One-layer perc.	Multi-layer perc.
Channel width (ps)	23.2	24.5	Some (11.2–19.5)
No. of MCA/taken channels	1500/1500	1024/1024	8192/3500
Approx. no. of counts in spec.	10 M	10 M	~400 k
Solver	Backward error prop.	Backward error prop.	Some
No. of hidden layers	1	1	Some
I_2 , τ_3 average error (%) on tested simulated spectra	7.3, 1.0	1.07–3.52, 0.55–1.21	—, —
I_2 , τ_3 average error (%) on tested real spectra	—, —	—, —	1.03, 1.70

more complicated cases where the distribution curve consists in a sum of gaussians, for example. As already mentioned in,¹⁸ in that case, a possibility of recognizing the distribution function would give compatibility to MELT.²⁷ Such an extension requires extending the calculations by applying another neural network, working in advance, which returns the parameters of the resolution curve in a given case. This problem has been solved by application of a Hopfield neural network.²⁰ Taking into account our collection of spectra, it was checked with the use of LT and (occasionally) with MELT that the apparatus resolution curve is one-gaussian for our spectra. Hence, they do not allow testing such an extended model.

Furthermore, the MCA module of spectrometers may differ in the time constant per channel Δ . Thus, spectra used as a training data set may be collected for different channel widths. Taking into account the method presented in Ref. 18 for fixed Δ , the training result is of little use for spectra collected with another Δ . On the contrary, we have shown the possibility of application of an improved algorithm to data collected for different Δ s. The data collected from many spectrometers may contribute to a large training data set, which allows solving the inverse problem for any PALS spectrum and, thus, may be a universal tool that can be used in different laboratories. Although the set of Δ used here is small, the accuracy of the results is quite good. To use this tool to determine the real-world spectrum parameters, the training process should be extended by adding the spectra measured for a wider range of Δ .

For greater generalization, it is possible to attach spectra collected for other compounds to the training data set. For consistency, it suffices for a training database to keep the same number of components (three here) in spectrum decomposition. However, in practice, some incompatibilities of the spectra for different compounds may arise because decomposition into a few exponential processes is probably always a simplification of a real case where some distribution of the size and shape of free volumes should be taken into account as well as other Ps formation details.

Although the approach presented here is reduced to the analysis of alkanes solely, the algorithm can be applied in calculation of PALS parameters of other types of samples as well. The architecture of the network needs to be adjusted according to the properties

of the samples. This simplest version of the neural network shown here requires keeping the declared number of components in the spectrum. Once this number is chosen for given architecture and in the training process, the testing proceeds with respect to the same criteria. The training data can be extended to include spectra for other compounds, provided that the interpretative meaning of input and output neurons is respected.

In the case of alkanes, the ortho/para ratio is fixed, based on our previous experience about alkanes. If this ratio is changing, the intensity of the para-Ps component should be regarded as an additional parameter to be taught.

The application of such a network for other types of materials, e.g., metals or porous materials, requires delivering the training data for these compounds (due to the differences in the range of lifetime values) or rebuilding the network architecture (if there is a change in the number of components). Furthermore, a more generalized network is not hard to imagine where the number of components is a parameter for testing by the network itself, whereas the optimal decomposition is finally chosen based on the score parameter. Such a network trained on spectra from different compounds could be distributed as an autonomous tool for studying a variety of materials, including cases of spectra coming from a completely new sample.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

M. Pietrow: Conceptualization (lead); Data curation (lead); Formal analysis (equal); Investigation (equal); Methodology (equal); Project administration (equal); Software (supporting); Visualization (supporting); Writing – original draft (lead). **A. Miaskowski:** Conceptualization (supporting); Data curation (supporting); Formal analysis (equal); Investigation (equal); Methodology (equal); Project administration (equal); Software (lead); Visualization (lead); Writing – original draft (supporting).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

- ¹T. Hirade, "Positron annihilation in radiation chemistry," in *Charged Particle and Photon Interactions with Matter. Recent Advances, Applications, and Interfaces*, edited by Y. Hatano, Y. Katsumura, and A. Mozumder (CRC Press Taylor & Francis Group, Boca Raton, 2011), Chap. 7, pp. 137–167.
- ²D. Manara, A. Seibert *et al.*, "Positron annihilation spectroscopy," in *Advances in Nuclear Fuel Chemistry*, edited by M. H. A. Piro (Elsevier Ltd., Woodhead Publishing, 2020), Chap. 2.3.5, pp. 126–131.
- ³W. Greiner and J. Reinhardt, *Field Quantization* (Springer, 2013).
- ⁴S. V. Stepanov and V. M. Byakov, "Physical and radiation chemistry of the positron and positronium," in *Principles and Applications of Positrons and Positronium Chemistry*, edited by Y. C. Jean, P. E. Mallone, and D. M. Schrader (World Scientific, 2003), pp. 117–149.
- ⁵T. Goworek, "Positronium as a probe of small free volumes in crystals, polymers and porous media," *Ann. Univ. Mariae Curie Skłodowska, sectio AA—Chemia* **LXIX**, 1–110 (2014).
- ⁶Y. C. Jean, J. D. Van Horn, W.-S. Hung, and K.-R. Lee, "Perspective of positron annihilation spectroscopy in polymers," *Macromolecules* **46**, 7133–7145 (2013).
- ⁷J. Kansy, "Microcomputer program for analysis of positron annihilation lifetime spectra," *Nucl. Instrum. Meth. A* **374**, 235–244 (1996).
- ⁸J. V. Olsen, P. Kirkegaard, N. J. Pedersen, and M. M. Eldrup, "Palsfit: A new program for the evaluation of positron lifetime spectra," *Phys. Stat Sol. C* **4**(10), 4004–4006 (2007).
- ⁹K. Wada and T. Hyodo, "A simple shape-free model for pore-size estimation with positron annihilation lifetime spectroscopy," *J. Phys. Conf. Ser.* **443**, 012003 (2013).
- ¹⁰M. Pietrow, "Remarks on energetic conditions for positronium formation in non-polar solids. coupled dipole method application," *Phys. Chem. Chem. Phys.* **17**, 27726–27733 (2015).
- ¹¹M. Pietrow and J. Wawrzyszczuk, "The influence of admixtures in *n*-alkanes on electron traps," *Mater. Sci. Forum* **733**, 75–79 (2013).
- ¹²J. Jegal, D. Jeong, E. S. Seo *et al.*, "Convolutional neural network-based reconstruction for positronium annihilation localization," *Sci. Rep.* **12**, 8531 (2022).
- ¹³J. L. Herraiz, A. Bembibre, and A. López-Montes, "Deep-learning based positron range correction of PET images," *Appl. Sci.* **11**(1), 1–13 (2021).
- ¹⁴M. Wędrowski, "Artificial neural network based position estimation in positron emission tomography," Ph.D. thesis (Interuniversity Institute for High Energies, Vrije Universiteit Brussel, Belgium, 2010).
- ¹⁵W. J. Whiteley, "Deep learning in positron emission tomography image deep learning in positron emission tomography image reconstruction," Ph.D. thesis (University of Tennessee, Knoxville, 2020).
- ¹⁶D. Petschke and T. E. M. Staab, "A supervised machine learning approach using naive Gaussian Bayes classification for shape-sensitive detector pulse discrimination in positron annihilation lifetime spectroscopy (PALS)," *Nucl. Instrum. Methods Phys. Res. A* **947**, 162742 (2019).
- ¹⁷N. H. T. Lemes, J. P. Braga, and J. C. Belchior, "Applications of genetic algorithms for inverting positron lifetime spectrum," *Chem. Phys. Lett.* **412**(4), 353–358 (2005).
- ¹⁸I. Pázsit, R. Chakarova, P. Lindén, and F. Maurer, "Unfolding positron lifetime spectra with neural networks," *Appl. Surf. Sci.* **149**, 97–102 (1999).
- ¹⁹R. An, J. Zhang, W. Kong, and B.-J. Ye, "The application of artificial neural networks to the inversion of the positron lifetime spectrum," *Chin. Phys. B* **21**(11), 117803 (2012).
- ²⁰V. C. Viterbo, J. P. Braga, A. P. Braga, and M. B. de Almeida, "Inversion of simulated positron annihilation lifetime spectrum using a neural network," *J. Chem. Inf. Comput. Sci.* **41**, 309–313 (2001).
- ²¹G. Rebal, A. Ravi, and S. Churiwala, *An Introduction to Machine Learning* (Springer Nature, 2019).
- ²²J. Heaton, *Introduction to the Math of Neural Networks* (Heaton Res., 2012).
- ²³F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.* **12**, 2825–2830 (2011); available at <https://jmlr.org/papers/v12/pedregosa11a.html>.
- ²⁴See https://en.wikipedia.org/wiki/Broyden-Fletcher-Goldfarb-Shanno_algorithm for "Broyden-Fletcher-Goldfarb-Shanno Algorithm" (last accessed November 20, 2022).
- ²⁵See https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html for "sklearn.neural_network.mlpregressor."
- ²⁶T. Goworek, M. Pietrow, R. Zaleski, and B. Zgardzińska, "Positronium in high temperature phases of long-chain even *n*-alkanes," *Chem. Phys.* **355**, 123–129 (2009).
- ²⁷A. Shukla, M. Peter, and L. Hoffmann, "Analysis of positron lifetime spectra using quantified maximum entropy and a general linear filter," *Nucl. Instrum. Meth. Phys. Res. Sec. A* **335**(1), 310–317 (1993).