Machine Learning in Fundamental Physics

Machine learning approach in the prediction of differential cross sections and structure functions of single pion electroproduction in the resonance region

A.V. Golda,[1, *] A.A. Rusova,[1, 2] E.L. Isupov,[2, †] V.I. Mokeev,[3] and V.V. Chistyakova[1]

[1]*Lomonosov Moscow State University, Faculty of Physics, Department of General Nuclear Physics. Russia, 119991, Moscow, Leninskiye Gory, bld. 1, str. 2.*
[2]*Skobeltsyn Institute of Nuclear Physics, Lomonosov Moscow State University (SINP MSU). Russia, 119991, Moscow, Leninskiye Gory, bld. 1, str. 5.*
[3]*Thomas Jefferson National Accelerator Facility. USA, VA 23606, Virginia, Newport News.*

This work explores artificial intelligence methods in the task of predicting differential cross sections in exclusive reactions of positively charged pion production induced by virtual photons. A fully connected neural network devoid of any prior theoretical knowledge about the scatterring process was trained on experimental data from the CLAS detector. We present a comparison of the network's predictions with experimental data in the form of graphs showing the dependence of differential cross sections on kinematic variables in the excitation energy regions of nucleon resonances, as well as a comparison of the structure functions depending on the values of invariant mass of the final hadron system. Based on this algorithm we can interpolate both the cross-section values and structure function values in different regions of phase space. The neural network approach preserves all correlations of the multidimensional space of kinematic variables, it is model independent and does not consume any a priori knowledge of the process, it is easily extensible to a high dimensional space, which can serve as a good basis for building Monte Carlo event generators or detailed rection analysis.

1. INTRODUCTION

From the first theoretical models of artificial neural networks [1] to modern transformer-based architectures [2], neural networks have undergone tremendous development in artificial intelligence methods. These machine learning methods have increasingly become essential tools for researchers across various scientific fields over time. Particle physics is no exception — artificial intelligence algorithms are now used to build event generators [3], suppress background process contributions, track particles [4], and much more. The availability of large datasets in particle physics makes it possible to effectively develop such machine learning models. In this work we propose an approach for training a neural network-based algorithm to accurately predict the differential cross-section values of exclusive reactions in different regions of phase space.

This study explores artificial intelligence methods for predicting the differential cross-sections of reactions in the processes of positively charged pion production induced by virtual photon:

$$\gamma^* + p \rightarrow n + \pi^+, \tag{1}$$

———

* E-mail: golda.av15@physics.msu.ru
† E-mail: isupov@jlab.org

where as a source of virtual photons was used a beam of electrons with energy ($E = 5.754\ GeV$, $E = 5.499\ GeV$, $E = 1.515\ GeV$) depending on the experiment. In considering the pion production reaction a dataset was used to predict the reaction cross-sections in various regions of phase space.

The task addressed in this study is regression, and the primary algorithm developed for this task is a deep fully connected neural network with an architecture consisting of **19** hidden layers, trained with a loss function that does not incorporate any prior theoretical knowledge of the process. The training data were taken from experiments using the CLAS detector, stored in the **CLAS Physics Data Base** [5].

2. EXCLUSIVE ELECTRON SCATTERING REACTIONS ON A PROTON. DIFFERENTIAL CROSS SECTIONS AND STRUCTURE FUNCTIONS

The initial studies of nucleon resonances began with the investigation of elastic scattering reactions of π-mesons on nucleons. The advantage of this method lies in the large cross-section of the meson and target nucleon interaction. However, the incident particle (meson) is not point-like and its internal structure is unknown. Moreover, not all resonances can be de-

Figure 1. Feynman diagram of the electromagnetic excitation of nucleon resonances.

Figure 2. Schematic representation of the kinematics of the pion electroproduction reaction on a proton.

tected in elastic pion-nucleon scattering, as many resonances decay with the emission of mesons other than pions and/or with the emission of more than one meson. Therefore, modern research into the structure of excited nucleon states is conducted in exclusive reactions induced by real and virtual photons.

Nucleon resonance parameters include its mass, width, quantum numbers, and photo- and electro-excitation amplitudes $A_{1/2}$, $A_{3/2}$, $S_{1/2}$. The vertex of the interaction between a virtual photon and a proton leading to the formation of a resonance is characterized by three helicity amplitudes $A_{1/2}$, $A_{3/2}$, $S_{1/2}$, which depend only on the square of the photon's transferred four-momentum. The indices indicate the helicity of the initial particle system (i.e., the projection of the total spin of the initial particle system onto the direction of the photon's momentum in the center-of-mass system). The amplitudes $A_{1/2}$, $A_{3/2}$ correspond to the interaction of a transversely polarized photon with a nucleon. Only these amplitudes contribute to the photo-excitation of nucleon resonances. The amplitude $S_{1/2}$ describes the excitation of resonances by longitudinally polarized virtual photons. If the photon is real in the reaction, then $S_{1/2} = 0$. Physical observables such as the cross-section in the center-of-mass system, photon beam asymmetry, recoil nucleon polarization, and target asymmetry, are uniquely expressed through the complex amplitudes of the reaction. The complex reaction amplitudes represent a superposition of contributions from nucleon resonances and non-resonant mechanisms. The Feynman diagram of the electromagnetic excitation of nucleon resonances is shown in Fig. 1.

In exclusive reactions, the kinematic characteristics of all particles in the final state are measured. It is convenient to introduce the following variables characterizing the initial state of the system: Q^2 – photon virtuality and W – the invariant mass of the final hadron system. Electron-nucleon excitation experiments allow the study of cross-sections at various val-
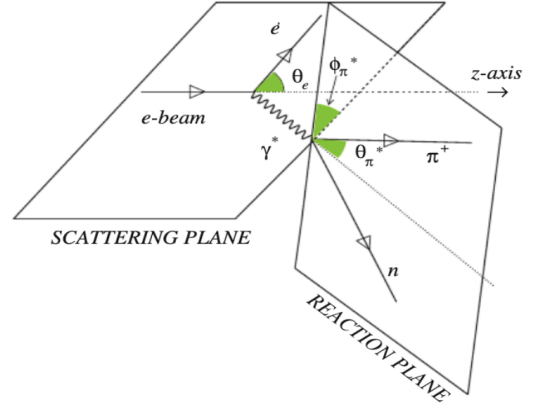
ues of photon virtuality Q^2:

$$Q^2 = -q^2,$$

where q is the four-momentum transferred by the virtual photon. The expression for the photon virtuality is as follows:

$$Q^2 = -q^2 = -(P - P')^2 = \vec{q}^2 - (E - E')^2$$
$$= 2(EE' - |\vec{q}||\vec{q}'|\cos\theta) - 2m_e^2 \approx 2EE'(1 - \cos\theta_e)$$
$$\approx 4EE'\sin^2\frac{\theta_e}{2},$$

where E is the initial electron energy, E' is the final electron energy, and

$$\theta_e = arccos\frac{p'_z}{E'}$$

is the electron scattering polar angle in the laboratory system, with p'_z being the projection of the scattered electron's momentum onto the z axis. In all formulas the speed of light is assumed to be equal to one:

$$c = 1.$$

The invariant mass of the final hadron system W is given by:

$$W^2 = (P_p + q)^2 = M^2 + 2M(E - E') - Q^2,$$

where M is the proton mass and P_p is the proton's four-momentum.

Other important characteristics of the reaction (1) are the kinematic variables: angle θ_π^* and angle ϕ_π^*. This reaction is considered in the photon-proton center-of-mass system and a schematic representation of the process is shown in Fig. 2.

The angle θ_π^* is the polar angle of the pion emission and the angle ϕ_π^* is the angle between the reaction plane and the scattering plane.

The scattering cross-sections are related to the probabilities of transitions between the initial and final

states of interacting particles and are measured experimentally by evaluating the count rate and momentum distributions of particles produced in scattering reactions at various collision energies. It is important to note that in this work we deal with the differential cross-sections of scattering under the influence of a single virtual photon (1) rather than scattering cross section of an electron on a proton.

Due to the rotational invariance of the amplitudes, the differential scattering cross-sections $\frac{d\sigma}{d\Omega}$ depend on the angle ϕ_π^* (the angle between the reaction plane and the scattering plane) as follows:

$$\begin{aligned}
\frac{d\sigma}{d\Omega} = {} & A(E, Q^2, W, cos\theta_\pi^*) + \\
& B(E, Q^2, W, cos\theta_\pi^*) \cdot cos2\phi_\pi^* + \\
& C(E, Q^2, W, cos\theta_\pi^*) \cdot cos\phi_\pi^*,
\end{aligned} \quad (2)$$

where $\boldsymbol{A(E, Q^2, W, cos\theta_\pi^*)}$, $\boldsymbol{B(E, Q^2, W, cos\theta_\pi^*)}$, $\boldsymbol{C(E, Q^2, W, cos\theta_\pi^*)}$ are structure functions. Structure functions contain all the information about the dynamics of the process. Structure functions, as expressed in (2), can be calculated from the differential scattering cross-sections by integration. The integrals to compute the structure functions are as follows:

$$A(E, Q^2, W, cos\theta_\pi^*) = \frac{1}{2\pi} \int_0^{2\pi} \frac{d\sigma(\phi_\pi^*)}{d\Omega} d\phi_\pi^*,$$

$$B(E, Q^2, W, cos\theta_\pi^*) = \frac{1}{\pi} \int_0^{2\pi} \frac{d\sigma(\phi_\pi^*)}{d\Omega} cos2\phi_\pi^* \, d\phi_\pi^*,$$

$$C(E, Q^2, W, cos\theta_\pi^*) = \frac{1}{\pi} \int_0^{2\pi} \frac{d\sigma(\phi_\pi^*)}{d\Omega} cos\phi_\pi^* \, d\phi_\pi^*.$$

These integrals can be replaced by numerical integration using the trapezoidal method. After training the algorithm, we can create a uniform grid of parameters (a feature grid), one of which is the angle ϕ_π^*. Based on this grid, the differential scattering cross-sections $\frac{d\sigma}{d\Omega}$ are predicted, and in the case of numerical integration using the trapezoidal method, these integrals are replaced by summation, and $\frac{d\sigma}{d\Omega}$ is replaced by an array of predicted values.

The study of single meson electroproduction reactions provides insight into the dynamics of processes occurring in nucleon resonances. Low-lying resonances (with masses around **1 GeV**) predominantly decay via the single-pion channel. We can see the Feynman diagram of nucleon resonance excitation through the electromagnetic channel in Fig. 1. Our work aims to study the so-called non-perturbative region (invariant mass of the final hadron system between **1** and **2 GeV**), which is responsible for the formation of the nucleon structure and nucleon resonances.

3. CLAS DETECTOR

This work is based on data from the **CLAS** (CEBAF Large Acceptance Spectrometer) detector, whose schematic is shown in Fig. 3. The CLAS detector at the Thomas Jefferson Laboratory operates with the Continuous Electron Beam Accelerator Facility (**CEBAF**), whose schematic is shown in Fig. 4. CEBAF consists of two parallel linear accelerators connected by arcs of bending magnets. Polarized electrons, generated by the impact of polarized laser radiation on a gallium arsenide cathode, leave the injector with an energy of **40 MeV** [6]. After this, the electrons are accelerated in the north and south linear accelerators. With each pass through the linear accelerator, the energy of the electrons increases by **2.4 GeV**. The final energy of the beams can reach **12 GeV**, for which additional accelerating cryomodules are used. The time interval between adjacent bunches is **0.67 ns**, while the time resolution of the detector is **1 ns**, so the electron beam can be considered continuous. The fast response and high data acquisition rate allow the accelerator to operate with a luminosity of $10^{34} cm^{-2} s^{-1}$, reaching values of up to $10^{38} cm^{-2} s^{-1}$ in the fixed-target collision mode [7].

The CLAS detector is based on **6** superconducting coils arranged around the path of the primary beam, which create a toroidal magnetic field. The detector consists of layers of various types of detectors for capturing different characteristics of particles. These include the main types of detectors, which are: drift chambers, Cherenkov gas counters, scintillators, and calorimeters. Drift chambers help determine the trajectories of charged particles, Cherenkov counters are necessary for electron identification, scintillation counters are needed to measure the speed of particles by calculating the time-of-flight, and calorimeters are used to measure particle showers. In the CLAS detector, charged particles are recorded in almost all directions, except for directions at extremely small and large values of the cosine of the polar angle, and also except for azimuthal directions occupied by the six toroidal field coils, allowing us to assert that the detector has a full solid angle of particle registration **4π**. The CLAS detector was created to study the physics of strong interactions, providing insights into fundamental ques-
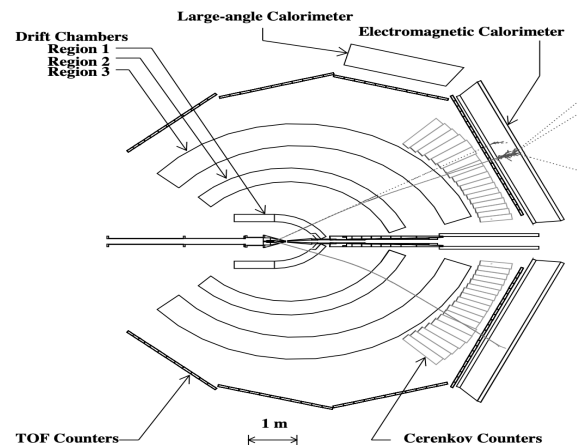

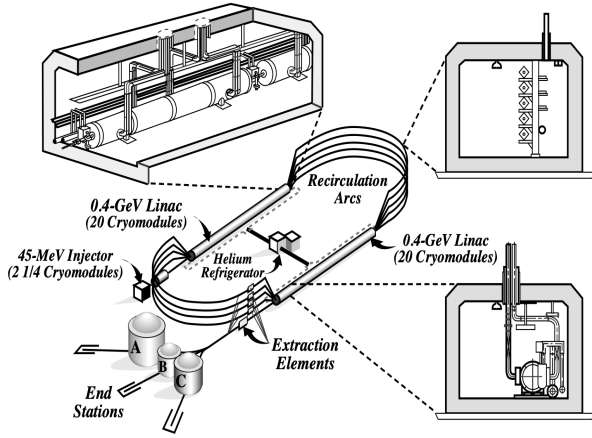Figure 3. Schematic representation of the CLAS detector.

Figure 4. Schematic representation of the CEBAF accelerator complex.

tions such as the confinement of colored objects within baryons and mesons, and the origin of 98% of hadron mass. The CLAS detector was built to study resonance physics in the nonperturbative domain. It has full angular coverage - 4π. The detector is capable of measuring all final state particles and their kinematic variables, such as scattering angles and four-momenta.

4. MACHINE LEARNING MODEL

4.1. Input Data

The reaction of a positively charged π-meson on a proton during the interaction of a virtual photon is uniquely described by five kinematic variables: E, W, Q^2, θ_π^*, ϕ_π^*. Although theoretically this set of variables uniquely describes the kinematics of the studied reaction, for predicting differential cross-sections, any input data that might potentially contribute positively to prediction accuracy can be used. One can engage in feature engineering based on the existing dataset, deriving new features from physical principles, or including entirely new values, such as detector responses at the hardware level (or others). However, in our work, we fixed the set of five variables, potentially sacrificing model quality by not adding new features. This assumption was made because at the prediction stage, we need to provide the model with the same set of features on which the model was trained, which might be impractical due to the difficulty or impossibility of computing these features. For example, if the feature set during training included the voltage on a specific photomultiplier tube (as an example of a potential feature), then to predict cross-section values at a specific point in phase space, we would also need to provide this voltage as input, even though it may be unknown until the actual experiment is conducted. Additionally, this assumption was made based on the fact that new features built from the existing five, such as degrees (squares, cubes, fourth powers) or trigonometric

functions (or other basic nonlinear dependencies) of Q^2 or W, do not contribute additional value to the accuracy of the models we trained, as do their sums or weighted products — we verified this empirically.

Thus, we fixed the dataset with five independent kinematic variables. The data were taken from the **CLAS Physics Data Base** [5]. The data is presented in tabular format, where each value is a real number with precision up to 3 decimal places. In addition to the five independent kinematic variables and the differential scattering cross-sections themselves, the dataset also includes the statistical error in determining the differential cross-sections.

The dataset contains approximately 10^5 measurements, where each row corresponds to 5 kinematic variables (features), one differential cross-section value (target variable) with its statistical error.

4.2. Model Selection

In this work, we address the task of predicting real numbers based on the given dataset—a regression problem. There are several classic machine learning algorithms to solve this task, such as linear regression models (the basic linear regression algorithm and its variations with regularization like RIDGE, LASSO); support vector machines (SVM); decision trees; ensemble models—random forests, various implementations of boosting (XGBoost). Additionally, the task can be solved using neural network approaches. In this work we tested all the listed algorithms on the provided dataset and selected a deep fully connected neural network as the primary algorithm for several reasons. Firstly, because we could not train these classic machine learning algorithms to the desired result, except for the boosting approach. The dataset itself contains information on how we should evaluate the quality of our model (dictating the baseline solution)—it is stored in the values of the statistical error of the differential scattering cross-sections (the target variable). Certainly, the trained algorithm should aim to minimize the difference between predictions and actual values as much as possible. However, if our predictions deviate from actual values by less than the average statistical error of the values themselves, it can be considered that the algorithm handles the task well; at least, such an algorithm can predict the error values no worse than the experiment. This may be treated as the baseline of any model. When training the classic machine learning algorithms we could not achieve prediction quality below this threshold (could not beat the baseline). Secondly, classic machine learning algorithms do not fully possess the necessary flexibility during training. The actual physical problem we set has a lot of auxiliary information that we could include in the training process to improve prediction quality, and not all presented algorithms can incorporate it. For example, the statistical error values of the differential cross-sections dictate the importance or weight of each individual object in the dataset. Additionally,
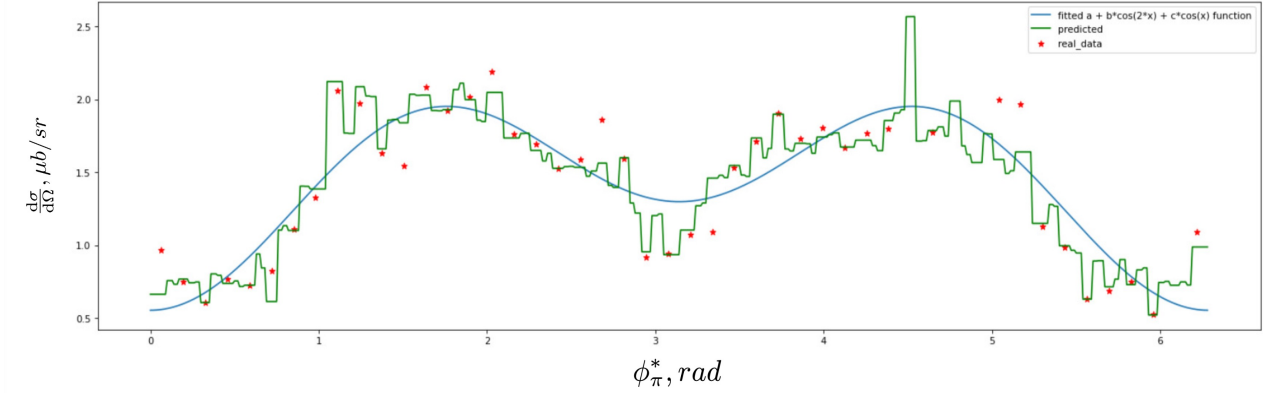
Figure 5. Comparison of XGBoost predictions and experimental data for differential scattering cross-sections $\frac{d\sigma}{d\Omega}$ as a function of the angle ϕ_π^*. The green curve represents the predictions of the trained XGBoost model. Red dots indicate actual data from the training set, and the blue curve represents fitting of real data according to the formula (2).

we know that differential scattering cross-sections have a strict physical dependence on the angle ϕ_π^*, which could also be used to improve training quality or to impose constraints on the network's predictions. However, of the algorithms listed, only boosting and neural network solutions can support such training modes. In fact, we were left with two approaches: neural network and boosting. But we rejected boosting due to the lack of smoothness in predictions (Fig. 5), even after introducing regularization, our predictions seemed overfitted. Therefore, we chose the neural network approach.

4.3. Neural Network Architecture

Today there is a wide range of types of neural networks that work with tabular data and solve regression problems [8]. Currently, there is no universal method for determining how well a given network architecture will predict the target variable, and the space of architectures is inherently infinite. Additionally, in this paper, we did not aim to obtain the most accurate machine learning algorithm, nor did we aim to compare algorithms with each other. On the contrary, we aimed to propose the simplest architecture that could solve the task with no less than the specific accuracy dictated by the average statistical error of the differential scattering cross-sections. Consequently, we chose an architecture entirely based on fully connected layers.

The current implementation is a fully connected neural network (Fig. 6), consisting of an input layer with **4** neurons corresponding to the size of the input data representing the kinematic characteristics of the reaction, excluding the initial energy of the electron beam $\boldsymbol{E}$, since after dividing the dataset by energy into 3 datasets, this parameter is constant for all reactions (described in 4.4.3), **19** hidden layers (variable length, with a maximum of **2000** neurons), and one output layer since the network solves a regression task for the

reaction's differential cross-section. The architecture of the network can be represented as a vector $\boldsymbol{T}$, where each element represents the number of neurons in the corresponding layer:

$$T = [4, 60, 80, 100, 120, 140, 240, 340, 440, 640,$$
$$2000, 1040, 640, 340, 240, 140, 100, 80, 60, 20, 1].$$

The activation function is the standard ReLU function [9]. In total, the network contains approximately $\boldsymbol{5 * 10^5}$ trainable parameters - weights and biases.

We do not claim that this architecture is optimal, but we would like to explain the principles we followed in designing this architecture. Firstly, according to the Cybenko Theorem (Universal Approximation Theorem) [10], we know that a single hidden layer in a fully connected neural network can approximate any nonlinear function. On the other hand, we know that such a single layer would be very large (in terms of the number of neurons). As in many cases, we need to balance between layer size and network depth. Secondly, we wanted to keep the training time of a single network low to test different possible architectures and training modes with various heuristics. Thus, we limited ourselves to approximately $\boldsymbol{10^7}$ parameters - weights and biases, which on average corresponds to about two hours of training on a modern hardware (almost any modern laptop) with our fixed dataset. Empirically, through manual experimentation, we found that the accuracy of the results improves if the number of neurons in the network increases monotonically from the input to the central hidden layer and decreases monotonically from the central hidden to the output layer, creating a diamond-shaped network structure. It was also found that such an architecture does not negatively impact the network's convergence and training time.

Modern deep learning frameworks provide extensive variability in how we can implement a given neural network. This implementation is based on the PyTorch framework [11].
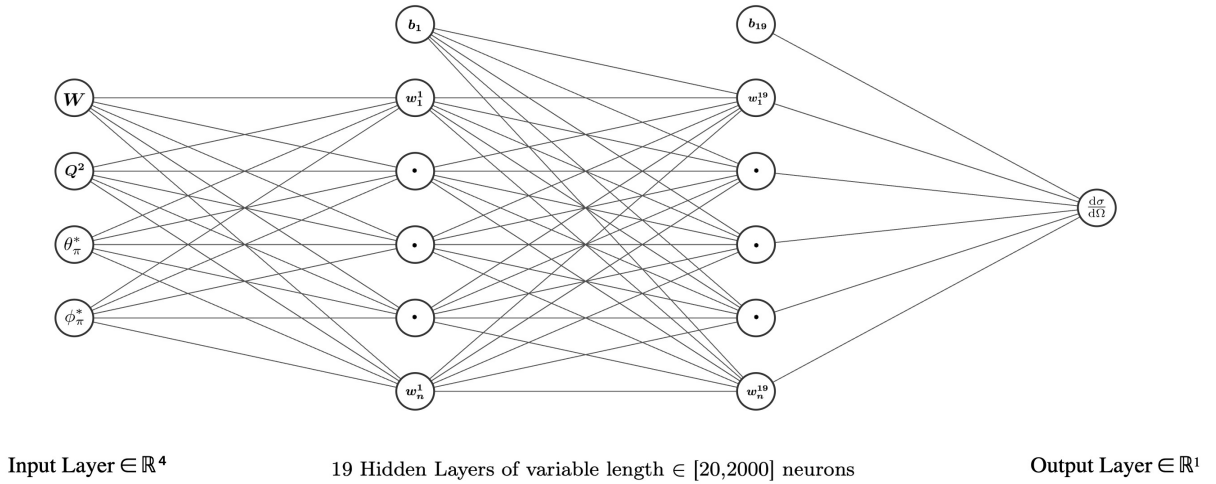
Figure 6. Schematic representation of the neural network architecture. In the input layer (on the left) are the symbols corresponding to the input variables, and in the output layer (on the right) is the symbol for the differential scattering cross-section corresponding to the target variable. The hidden layers had variable lengths, with the symbols b_i representing the biases of the layer, and the neuron weights represented by w_i^i.

4.4. Network Training Process

Such networks are not new, their foundations were introduced in the early 1960s along with the backpropagation method. However, classical use of networks does not yield satisfactory results. During our work, a large number of network variants (about **200** networks) were trained and validated in various architectural variations, including different heuristics and corresponding hyperparameters.

Here are some heuristics we used to improve prediction quality.

4.4.1. Augmentation

Since we had only 10^5 differential cross-sections (objects in the dataset), we decided to artificially increase our dataset through augmentation. Augmentation was performed by subsampling from a Gaussian distribution. Specifically, for each dataset object, we slightly shifted the feature values by an amount equal to a random value from a Gaussian distribution with parameters: μ - feature value, σ - standard deviation of all values of this feature divided by coefficient k_f. This coefficient for the feature set was chosen so that the augmented feature values did not deviate significantly from the original values (less than 1 percent of the value itself) and in our case was $k_f = 30$. Augmentation of the target variable was performed similarly, except for the parameters of the Gaussian distribution. Since we know the values of the statistical error of the target variable, we can augment it within the entire range of error. In our case, we shifted the target

variable values by an amount equal to a random value from a Gaussian distribution with the same parameters: μ - the target variable value (specific differential cross-section), and σ - statistical error of the differential cross-section, divided by coefficient k_t. This coefficient for the target variable was taken as $k_t = 3$, based on the reasoning that within three σ (three standard deviations) lies **99%** of the data, and thus, in **99%** of the augmentation, we would not exceed the statistical error. We used an augmentation factor of $m = 25$, meaning that the original dataset was increased by m times, with each row of the dataset being multiplied or "duplicated" **25** times from the Gaussian distribution with corresponding distribution parameters.

4.4.2. Data Standardization

We standardized the data by subtracting the sample mean and dividing by the sample standard deviation. Sample characteristics were taken for each feature individually, and the standardization procedure was performed independently for each feature.

4.4.3. Splitting the Dataset into 3 Separate Models

Initially we trained a single unified neural network on the entire dataset. However, because the experimental data is essentially a set of multiple experiments, then we see that it forms into some clusters grouped by initial energy of the electron beam E (Fig. 7). These clusters have different sizes in terms of the data volume - which leads to "disproportional"
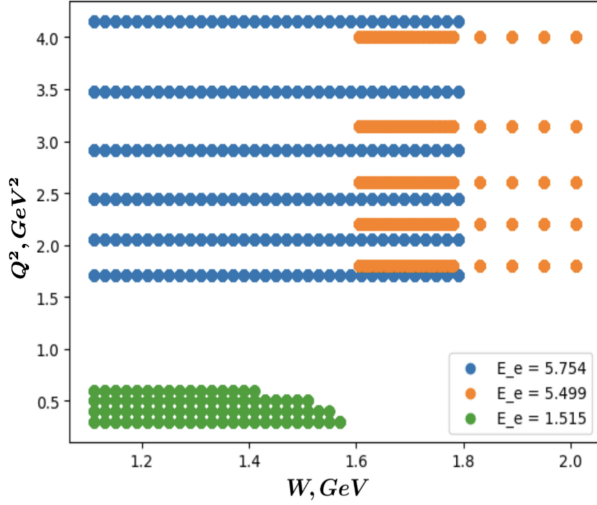
Figure 7. Population density map.

Figure 8. Plot of learning rate changes as a function of training step - both quantities are dimensionless.

learning, where the network picks up relationships in one part of the phase space better than another and transfers them to the entire phase space due to lack of data. Moreover, when training the unified network, we observe an undesirable effect, where we can see that in a certain volume of phase space (mainly when W is between 1.6 and 1.8 GeV) the values of predictions have non-monotonic and non-smooth nature. It happens mostly due to the fact that at different values of energy, the values of W and Q^2 overlap, while the values of the target variable differ - the neural network picks this up and transfers it to the value for other energies creating that non-smooth nature in predictions.

Therefore, we decided to split the dataset into **3** different datasets based on the initial energy of the electron ($E = 5.754$ GeV, $E = 5.499$ GeV, $E = 1.515$ GeV) and train **3** separate models. This approach significantly improves prediction accuracy, both for individual models and collectively.

4.4.4. Introducing Weights into the Loss Function

The value of the statistical error of the differential scattering cross-sections (or target variable) gives us information on which objects in training we would like to prioritize in terms of penalization. That is, if the value of the statistical error of the target variable is small, we would like the network to focus on learning from such objects and apply a larger penalty for incorrect predictions. In this work, we used **RMSELoss**

$$RMSELoss = \sqrt{\frac{\sum_i^N (y_i - \hat{y}_i)^2}{N}}, \qquad (3)$$

and the weighted RMSELoss is given by

$$wRMSELoss = \sqrt{\frac{\sum_i^N w(y_i - \hat{y}_i)^2}{N \sum_i^N w}}, \qquad (4)$$
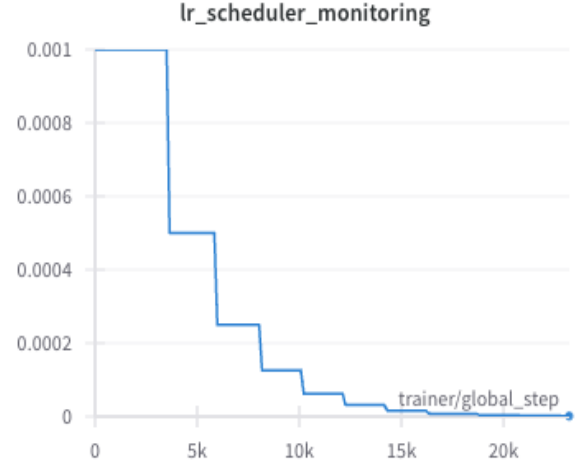
where w equals the reciprocal of the statistical error of the differential cross-sections, index i iterates over all values from the dataset from 0 to N, y_i - real values of the target variable (differential scattering cross-sections), $\hat{y}_i$ - predicted values of the target variable.

4.4.5. Early Stopping Conditions

To reduce training time and automate the process, we used early stopping conditions. If the value of our loss function RMSE (3) or wRMSE (4) decreased by less than 10^{-5} over the last **10** epochs, we stopped the training, assuming that further epochs would not contribute significantly.

4.4.6. Dynamic Learning Rate Adjustment

The learning rate is a key training parameter that needs to be tuned and controlled for optimal results. In our approach, we used dynamic learning rate adjustment - the ReduceLROnPlateau implementation from PyTorch [11]. We started with a value of **0.001** and each time the mean absolute error **MAE** on the validation dataset plateaued (did not decrease) over the last **5** epochs, we reduced the learning rate by a factor of **2** (Fig. 8).

4.4.7. Experiment Versioning

Since we trained about **200** networks while searching for the most effective algorithm through various architectural options and heuristic combinations, it was necessary to develop a system for storing and describing experiments. For this purpose, we used the free cloud platform **Weights&Biases** [12], which stores
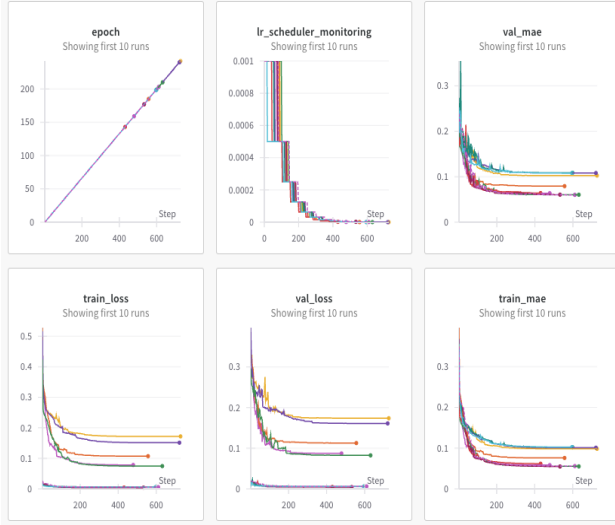
Figure 9. Snapshot of the **Weights&Biases** platform interface. The graphs show key quality metrics and training mode configurations as a function of step, specifically (left to right, top to bottom): number of training epochs (top left), learning rate values (top center), **MAE** metric for the validation dataset (top right), loss function value for the training dataset (bottom left), loss function value for the validation dataset (bottom center), **MAE** metric for the training dataset (bottom right). Each graph contains several curves representing different network configurations.

Figure 10. Snapshot of the **Weights&Biases** interface for the final models. The graphs show key quality metrics and training configuration over time, specifically (from left to right, top to bottom): number of training epochs (top left), learning rate values (top center), **MAE** metric for the validation dataset (top right), loss function value for the training dataset (bottom left), loss function value for the validation dataset (bottom center), **MAE** metric for the training dataset (bottom right). Each graph contains three curves of different colors representing different network experiments for different electron beam energies.

versions of experiments, including training parameters, network architecture, and all intermediate and final metrics in tabular and interactive graph formats. The platform's interface also allows quick tracking of all key training parameters, including validation loss and mean squared error (Fig. 9). By monitoring these parameters, we primarily preferred architectures and configurations that yielded the lowest mean squared error on the validation dataset.

4.4.8. Modification of Loss Function Considering Known Dependence of $\frac{d\sigma}{d\Omega}$ on ϕ_π^*

From a physical standpoint, we know that differential cross-section values depend in a certain way on the angle ϕ_π^* (2). Since our final predictions are differential cross-sections, we could additionally penalize the neural network if it predicted values significantly deviating from this dependence. Thus, our loss function **RMSELoss** (3) becomes:

$$RMSELoss = \sqrt{\frac{\sum_i^N (y_i - \hat{y}_i)^2}{N}} +$$

$$\alpha \sum_i^N \frac{y_i - (A_i + B_i cos(2\phi_i) + C_i cos(\phi_i))}{N},$$

where index i ranges from 0 to N; y_i - experimental values of the target variable (differential cross-sections); ϕ_i - angle between the reaction plane and the scattering plane for a specific differential cross-section

value (one of the features corresponding to the target variable for the specific i-th object); A_i, B_i, C_i - values of structure functions obtained by numerically integrating the dependence of differential cross-sections on the angle ϕ_π^* (2) over the interval from 0 to 2π; α - weighting coefficient, chosen accordingly. The final models whose results we present in this paper did not include this correction because we were not satisfied to improve the results significantly. Such a method of introducing corrections to the loss function is well studied and often used in practice [13, 14], but within the scope of this paper it should be more thoroughly tested and applied in the next stages of the project.

4.4.9. Optimizer

We used the ADAM stochastic optimization algorithm [15] and specifically its implementation in the PyTorch [11] framework.

5. RESULTS

Machine learning algorithms can be validated in various ways, and since this work is focused on specific physical processes rather than abstract entities, we can also provide some physical checks in addition to classical metrics for algorithm validation.

Among the classical quality metrics for model training, we will highlight just one basic metric. The value
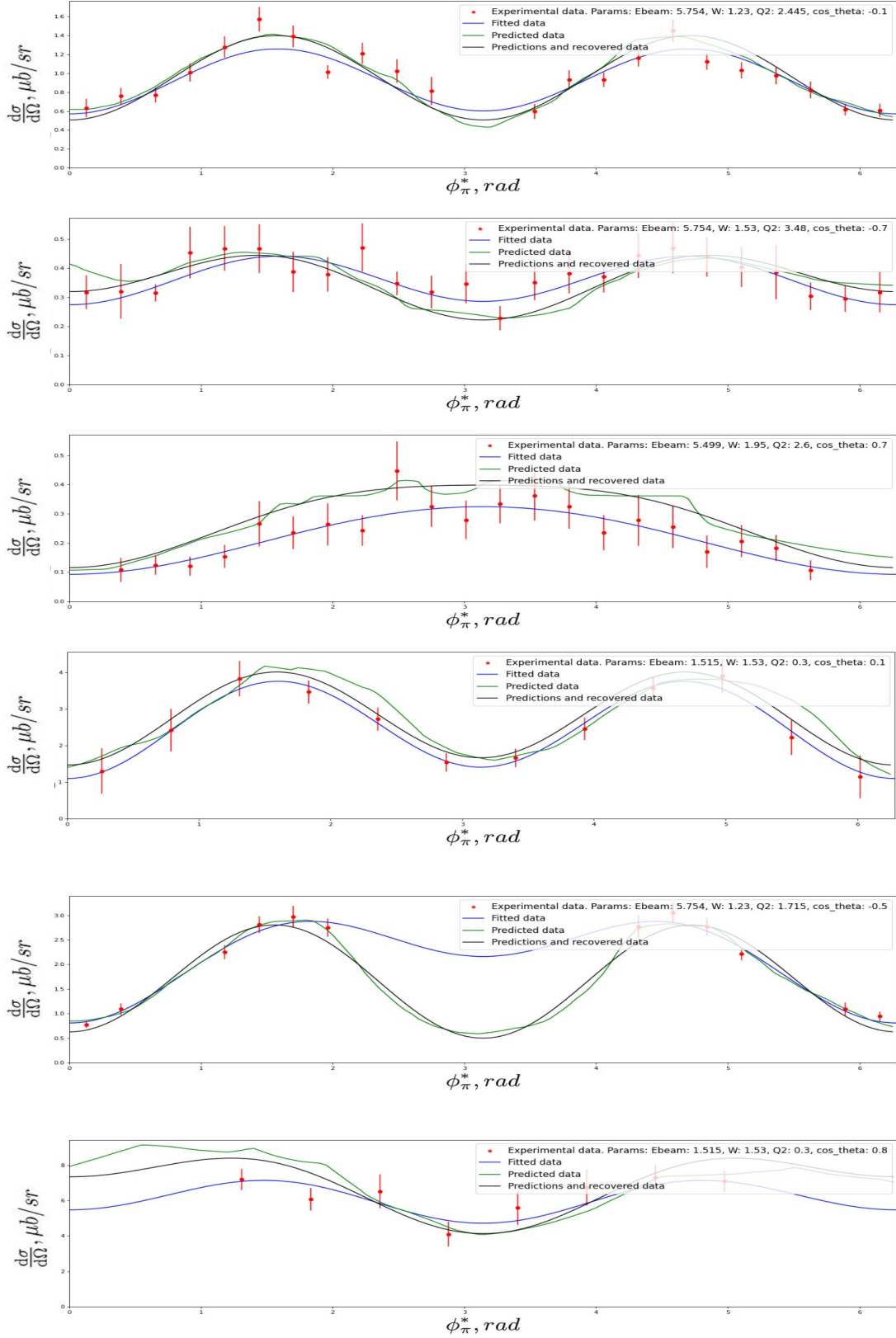
Figure 11. Comparison of model predictions for differential scattering cross sections and experimental data as a function of angle ϕ_π^* in a fixed region of phase space, specifically: $E = 5.754\,GeV$, $W = 1.23\,GeV$, $Q^2 = 2.445\,GeV^2$, $\theta_\pi^* = -0.1$ (first from the top); $E = 5.754\,GeV$, $W = 1.53\,GeV$, $Q^2 = 3.48\,GeV^2$, $\theta_\pi^* = -0.7$ (second from the top); $E = 5.499\,GeV$, $W = 1.95\,GeV$, $Q^2 = 2.6\,GeV^2$, $\theta_\pi^* = 0.7$ (third from the top); $E = 1.515\,GeV$, $W = 1.53\,GeV$, $Q^2 = 0.3\,GeV^2$, $\theta_\pi^* = 0.1$ (fourth from the top); $E = 5.754\,GeV$, $W = 1.23\,GeV$, $Q^2 = 1.715\,GeV^2$, $\theta_\pi^* = -0.5$ (fivth from the top); $E = 1.515\,GeV$, $W = 1.53\,GeV$, $Q^2 = 0.3\,GeV^2$, $\theta_\pi^* = 0.8$ (sixth from the top).
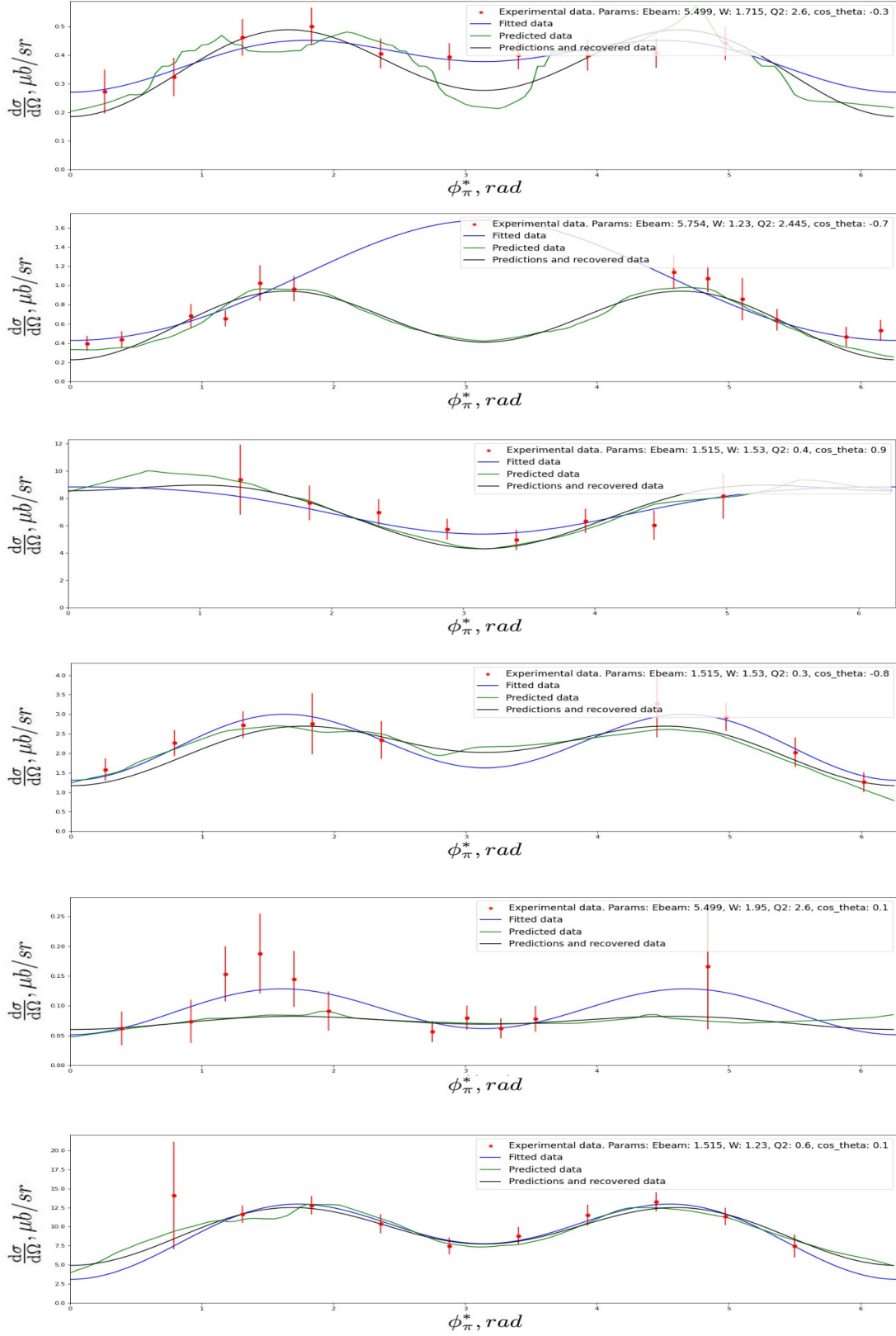
Figure 12. Comparison of model predictions for differential scattering cross sections and experimental data as a function of angle ϕ_π^* in a fixed region of phase space, specifically: $E = 5.499\,GeV$, $W = 1.715\,GeV$, $Q^2 = 2.6\,GeV^2$, $\theta_\pi^* = -0.3$ (first from the top); $E = 5.754\,GeV$, $W = 1.23\,GeV$, $Q^2 = 2.445\,GeV^2$, $\theta_\pi^* = -0.7$ (second from the top); $E = 1.515\,GeV$, $W = 1.53\,GeV$, $Q^2 = 0.4\,GeV^2$, $\theta_\pi^* = 0.9$ (third from the top); $E = 1.515\,GeV$, $W = 1.53\,GeV$, $Q^2 = 0.3\,GeV^2$, $\theta_\pi^* = -0.8$ (fourth from the top); $E = 5.499\,GeV$, $W = 1.95\,GeV$, $Q^2 = 2.6\,GeV^2$, $\theta_\pi^* = 0.1$ (fivth from the top); $E = 1.515\,GeV$, $W = 1.95\,GeV$, $Q^2 = 0.5\,GeV^2$, $\theta_\pi^* = 0.4$ (sixth from the top).
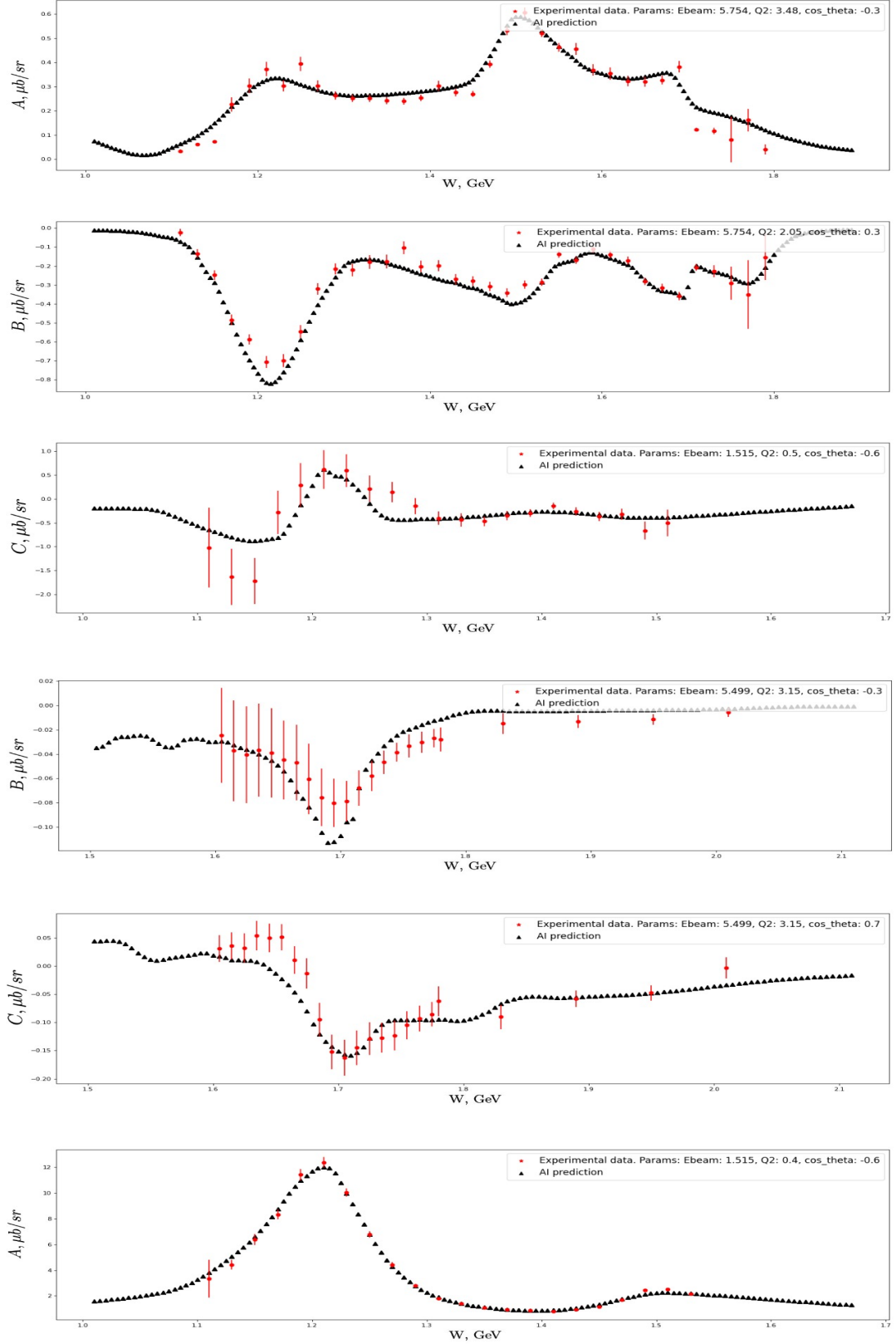
Figure 13. Comparison of model predictions (black dots) for recalculated structure functions depandance from invariant mass of the final hadron system W with experimental data (red dots) with statistical error in a fixed region of phase space, specifically: Function A, $E = 5.754\,GeV$, $Q^2 = 3.48\,GeV^2$, $\theta_\pi^* = -0.3$ (first from the top); Function B, $E = 5.754\,GeV$, $Q^2 = 2.05\,GeV^2$, $\theta_\pi^* = 0.3$ (second from the top); Function C, $E = 1.515\,GeV$, $Q^2 = 0.5\,GeV^2$, $\theta_\pi^* = -0.6$ (third from the top); Function B, $E = 5.499\,GeV$, $Q^2 = 3.15\,GeV^2$, $\theta_\pi^* = -0.3$ (fourth from the top); Function C, $E = 5.499\,GeV$, $Q^2 = 3.15\,GeV^2$, $\theta_\pi^* = 0.7$ (fivth from the top); Function A, $E = 1.515\,GeV$, $Q^2 = 0.4\,GeV^2$, $\theta_\pi^* = -0.6$ (sixth from the top).

of the Mean Absolute Error (i.e., the average difference between prediction and actual value) is given by:

$$MAE = \frac{\sum_i^N |y_i - \hat{y}_i|}{N}, \tag{5}$$

where index i ranges over all values from the dataset from 0 to N, y_i are the actual values of the target variable (differential scattering cross-sections), and $\hat{y}_i$ are the predicted values of the target variable. This error on cross-validation of our three algorithms (trained on different initial electron beam energies) is $MAE = 0.08$ (5). For comparison, the average value of the differential cross-sections in our sample is $AVG(\frac{d\sigma}{d\Omega}) = 1.158 \pm 0.2 \, mcb/sr$. Where ± 0.2 is the average statistical error of the differential scattering cross-section measurements (target variable). Our predictions are on average within the statistical error range.

As an example, we present a graph from the Weights&Biases interface [12] with examples of validation curves and training curves, along with a graph of learning rate dynamics (Fig. 10).

In this case, it can be seen that the prediction error is generally smaller than the statistical error of the data itself. However, the most interesting validations are physical validations of the algorithm. As such validation, we constructed distributions of scattering cross-sections as a function of angle ϕ_π^* (2). The graph shows experimental data (in different but fixed regions of phase space) as red dots. The same data, but fitted to a known angular dependence, where $A(E, Q^2, W, cos\theta_\pi^*) \,/\, B(E, Q^2, W, cos\theta_\pi^*) \,/\, C(E, Q^2, W, cos\theta_\pi^*)$ are structure functions carrying information about the dynamics of the process. The graph shows that predictions are quite close to experimental data and fitted data (Fig. 11, 12). In this work, we performed similar comparisons for the entire phase space where experimental data are available (Fig.7), and we observed high prediction accuracy across the board. We also note that since the trained algorithms store information from the entire training dataset, in some regions of phase space, the algorithms show higher accuracy in interpolation mode compared to data fitting. This effect can be observed in sparsely populated regions with experimental data, for example, the second graph from the bottom (Fig. 11) and the second graph from the top (Fig. 12).

Additionally, structure functions can be uniquely obtained from the reaction cross-sections. Fig. 13 shows a comparison of model predictions (black dots) as a function of the invariant mass of the final hadron system with experimental data (red dots) in different regions of phase space. This method is, in a sense, a checksum comparison since structure functions are computed by numerical integration of cross-sections. The graph (Fig. 13) also shows that predictions are quite close to experimental data. In this work, we also performed similar comparisons for the entire phase space where experimental data are available (Fig. 7) and observed high prediction accuracy across the board.

6. CONCLUSION

This work explores artificial intelligence methods for predicting differential cross-sections in exclusive reactions involving positively charged pions under the influence of virtual photon. Based on experimental data from the CLAS detector, a set of fully connected neural networks was trained without incorporating any prior theoretical knowledge about the scattering process. A set of validations was proposed, including MAE metric, comparison of differential scattering cross-sections predictions with experimental data in a fixed phase space region, and comparison of structure functions. These validations demonstrate that these neural networks have high accuracy and that it is possible to interpolate both cross-section values and structure function values in various regions of phase space. The neural network approach preserves all correlations of the multidimensional space of kinematic variables, it is model independent and does not consume any a priori knowledge of the process, it is easily extensible to a high dimensional space, it also has better quality in some localized areas with low data density, which can serve as a good basis for building Monte Carlo event generators or detailed rection analysis.

[1] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386–408 (1958).

[2] A. Vaswani, N. Shazeer, N. Parmar, et.al., Attention is All you Need (Neural Information Processing Systems 28, 2017), pp. 2440-2448.

[3] T. Alqhamdi, Y. Alanazi, M. Battaglieri, L. Bibrzycki, A. V. Golda, et.al., Toward a generative modeling analysis of CLAS exclusive 2pi photoproduction. Phys. Rev. D 108, 9, pp. 1-14 (2023).

[4] P. Thomadakis, A. Angelopoulos, G. Gavalian, N. Chrisochoides. Using Machine Learning for Particle Track Identification in the CLAS12 Detector. Computer Physics Communications 276, 152 (2022).

[5] CLAS Physics Database,
https://clas.sinp.msu.ru/cgi-bin/jlab/db.cgi

[6] C. Leemann, D. Douglas, G. Krafft. The Continuous Electron Beam Accelerator Facility: CEBAF at the Jefferson Laboratory. Annu. Rev. Nucl. Part. Sci. 51, pp. 413-450 (2001).

[7] B. Mecking, G. Adams, S. Ahmad, et.al., The CEBAF large acceptance spectrometer (CLAS). Nuclear Instruments and Methods Physics Research Section A. 3/503:513-553 (2003).

[8] V. Borisov, T. Leemann, K. Sebler, J Haug, et.al., Deep Neural Networks and Tabular Data: A Survey. IEEE Transactions on Neural Networks and Learning Systems (2022). 10.11091. pp. 1-21.

[9] A.F. Agarap, Deep Learning using Rectified Linear Units (ReLU). arXiv preprint arXiv:1803.08375 (2018).

[10] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks (1989), 2. 5. pp. 359-366.

[11] PyTorch, Python DNN framework,
https://pytorch.org/

[12] Weights&Biases platform,
https://wandb.ai

[13] C. Meng, S. Seo, et al. "When Physics Meets Machine Learning: A Survey of Physics-Informed Machine Learning." arXiv preprint arXiv:2203.16797 (2022)

[14] L. Rueden, V. Mayer, et al. "Informed Machine Learning–A Taxonomy and Survey of Integrating Knowledge into Learning Systems." arXiv preprint arXiv:1903.12394 (2019)

[15] D. Kingma, J. Ba. Adam: A Method for Stochastic Optimization. 3rd International Conference for Learning Representations (2015).