



# Deep machine learning for the PANDA software trigger

Peiyong Jiang<sup>1,2,3,a</sup> , Klaus Götzen<sup>1</sup> , Ralf Kliemt<sup>1,4</sup>, Frank Nerling<sup>1,3,5</sup> , Klaus Peters<sup>1,3,5</sup>

<sup>1</sup> GSI Helmholtzzentrum für Schwerionenforschung GmbH, Darmstadt, Germany

<sup>2</sup> Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou, China

<sup>3</sup> Helmholtz Forschungsakademie Hessen für FAIR (HFHF), GSI Helmholtzzentrum für Schwerionenforschung, Campus Frankfurt, Frankfurt am Main, Germany

<sup>4</sup> Ruhr-Universität Bochum, Bochum, Germany

<sup>5</sup> Goethe-Universität Frankfurt, Frankfurt am Main, Germany

Received: 29 November 2022 / Accepted: 11 April 2023 / Published online: 27 April 2023  
© The Author(s) 2023

**Abstract** Deep machine learning methods have been studied for the software trigger of the future PANDA experiment at FAIR, using Monte Carlo simulated data from the GEANT-based detector simulation framework PandaRoot. Ten physics channels that cover the main physics topics, including electromagnetic, exotic, charmonium, open charm, and baryonic reaction channels, have been investigated at four different anti-proton beam momenta. Different classification concepts and network architectures have been studied. Finally a residual convolutional neural network with four residual blocks in a binary classification scheme has been chosen due to its extendability, performance and stability. The presented study represents a feasibility study of a completely software-based trigger system. Compared to a conventional selection method, the deep machine learning approach achieved a significant efficiency gain of up to 200%, while keeping the background reduction factor at the required level of 1/1000. Furthermore, it is shown that the use of additional input variables can improve the data quality for subsequent analysis. This study shows that the PANDA software trigger can benefit greatly from the deep machine learning methods.

## 1 Introduction

### 1.1 Motivation

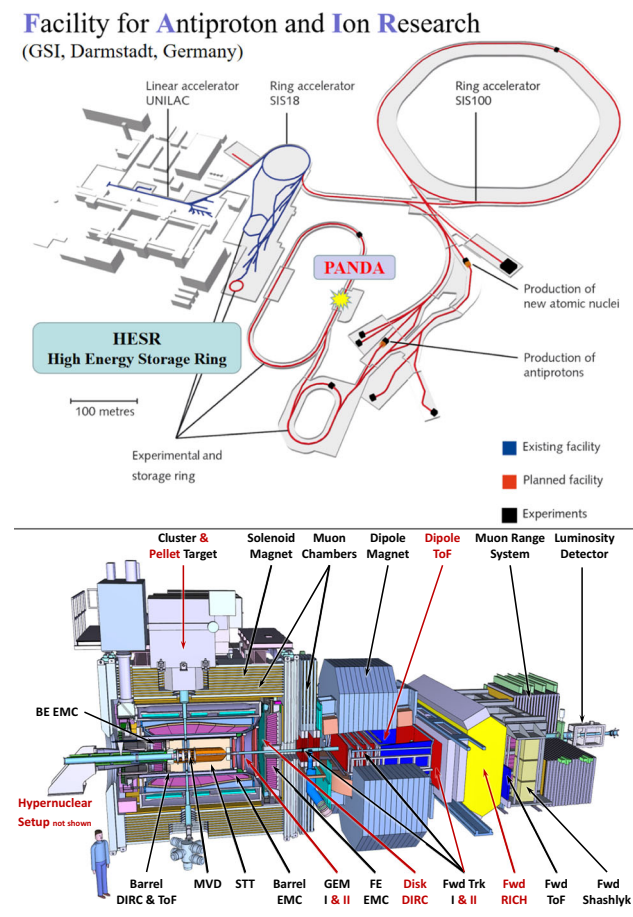
The antiProton ANnihilation at DArmstadt experiment (PANDA) at the Facility for Antiproton and Ion Research (FAIR) [1] aims to explore hadron physics by means of

antiproton induced reactions [2,3]. A wide array of physics topics and open questions are within the scope of PANDA, including strangeness, charm and exotic physics, nucleon structure, and hadrons in nuclei [4,5]. Some examples are the investigation of the charmonium spectrum, hyperon spectroscopy, electromagnetic form factors as well as the search for exotic states, e.g. hybrids, glueballs and hypernuclei. At centre-of-mass energies between 2 and 5.5 GeV the total  $\bar{p}p$  cross section ranges between 50 and 100 mb [6], while the signals of interest have cross sections between a couple of microbarns and a few nanobarns or even picobarns. Thus, with signal cross sections being many orders of magnitude smaller than the total proton-antiproton cross section, extracting the physics of interest is challenging in terms of background suppression.

With an expected average reaction rate of about 20 MHz, and with the average event size of 10–20 kB, the full data rate will be roughly 200 GB/s or more. Because there is only a tiny fraction of events containing physics processes of interest, it would be inefficient to store all the data. In order to identify the interesting events in an online environment, a sophisticated trigger system is necessary. The PANDA triggering system is foreseen to reduce the rate of events to be stored by an approximate factor of 1000 to 20 kHz, reducing the stored data to 200 MB/s, which still leads to about 1 PB stored data per year.

Currently operating and upcoming experiments are embracing a new paradigm to trigger by online event selection in various degrees. The high-level trigger of the ALICE experiment performs the full event reconstruction, calibration and high-level data quality assurance in almost real time [7]. The LHCb collaboration is currently making efforts to implement full online analysis chains for event filtering [8]. XENON1T has been collecting data altogether with a software-trigger

<sup>a</sup> e-mails: [p.jiang@gsi.de](mailto:p.jiang@gsi.de); [jiangpeiyong@impcas.ac.cn](mailto:jiangpeiyong@impcas.ac.cn) (corresponding author)



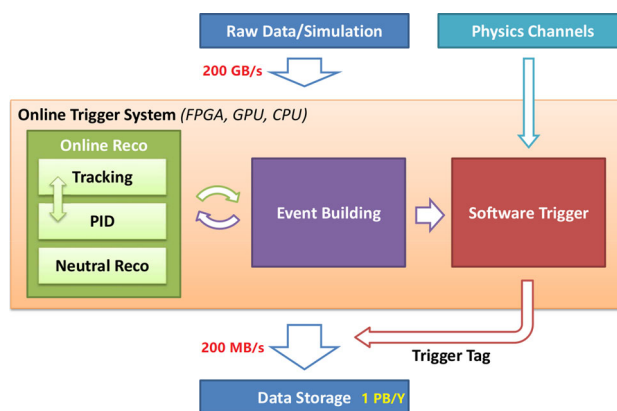
**Fig. 1** Overview of the FAIR facility (top). The PANDA experimental setup (bottom) with the initial detectors (in black) and staged upgrades (in red) [3] (and references therein)

[9]. At the future FAIR Facility, the CBM [10] and PANDA experiments will also employ fully software-based trigger systems.

In PANDA, the kinematic similarity of signal and background reactions combined with a wide physics interest and the high interaction rate puts the key challenge on the trigger system and consequently the selections have to be made on full event candidates with high-level information. To cope with the challenges for proper signal-background separation, deep machine learning methods with neural networks (NN) are studied for the PANDA Software Trigger to improve performance compared to a conventional cut-and-count method.

## 1.2 Experiment environment

FAIR is an international accelerator facility under construction in Darmstadt, Germany [1,3]. It expands the existing accelerator complex in a large scale, shown in Fig. 1. An antiproton beam will be prepared in a cascade of accelerators, making use of almost the whole facility. The synchrotron High-Energy Storage Ring (HESR) [11] will store, cool and



**Fig. 2** Schematic overview of the complete PANDA online trigger system

accelerate the antiprotons, which will have momenta in the range between 1.5 GeV/c and 15 GeV/c, corresponding to centre-of-mass energies in the antiproton-proton reactions at PANDA between about 2 and 5.5 GeV. With a luminosity of up to  $L = 2 \times 10^{32} \text{ cm}^{-2} \text{ s}^{-1}$  in the “high luminosity mode” and with a beam momentum resolution of  $dp/p < 2 \times 10^{-5}$  in the “high resolution mode” HESR is an essential component for PANDA that allows for precise measurements.

The PANDA experiment, shown in Fig. 1, consists of two spectrometers, one barrel-shaped surrounding the interaction point, the other covering the forward direction, both providing measurements for precise charged particle tracking and particle identification as well as electromagnetic calorimetry.

## 1.3 Software trigger in PANDA

The PANDA online trigger system will consist of online reconstruction, event building, and the Software Trigger, as illustrated in the schematic in Fig. 2. First, the online reconstruction of neutral particles and charged particles will be performed as completely as possible using fast algorithms including electromagnetic calorimetry (EMC), and the Particle IDentification (PID) information will be assigned to the corresponding tracks. This will be combined with the event reconstruction process, possibly in an iterative manner, to provide the online event candidate with reconstructed final state particle information to the Software Trigger module. The event candidate will then be processed in the Software Trigger module by the selection algorithms. Then the event candidates will be tagged if they are consistent with a signal signature. Finally, an event candidate will be written to the data storage if any of the trigger algorithms accepts it to be a signal event. At the highest interaction rates event pile-up in the order of 2–3 events into one event candidate will play a role. Thus the software trigger will focus on finding pieces of interesting signatures inclusively.

**Table 1** List of ten physics channels for the Software Trigger study together with the corresponding codes used in this document and the trigger signatures

Physics topic	Reaction channel	Code	Trigger
Electromagnetic	$\bar{p}p \rightarrow e^+e^-$	<i>ee</i>	$\bar{p}p \rightarrow e^+e^-$
Exotics	$\bar{p}p \rightarrow \phi_{(1)}\phi_{(2)}$	<i>Phi</i>	$\phi \rightarrow K^+K^-$
Charmonium	$\bar{p}p \rightarrow \eta_c\pi^+\pi^-$	<i>Etac</i>	$\eta_c \rightarrow K_S^0K^-\pi^+$
	$\bar{p}p \rightarrow J/\psi\pi^+\pi^-$	<i>J2e</i>	$J/\psi \rightarrow e^+e^-$
	$\bar{p}p \rightarrow J/\psi\pi^+\pi^-$	<i>J2mu</i>	$J/\psi \rightarrow \mu^+\mu^-$
Open charm	$\bar{p}p \rightarrow D^0\bar{D}^0$	<i>D0</i>	$D^0 \rightarrow K^-\pi^+$
	$\bar{p}p \rightarrow D^+D^-$	<i>Dch</i>	$D^+ \rightarrow K^-\pi^+\pi^+$
	$\bar{p}p \rightarrow D_s^+D_s^-$	<i>Ds</i>	$D_s^+ \rightarrow K^+K^-\pi^+$
Baryons	$\bar{p}p \rightarrow \Lambda\bar{\Lambda}$	<i>Lam</i>	$\Lambda \rightarrow p\pi^-$
	$\bar{p}p \rightarrow \Lambda_c\bar{\Lambda}_c$	<i>Lamc</i>	$\Lambda_c \rightarrow pK^-\pi^+$
Background	$\bar{p}p$ generic	DPM	-

## 2 Physics channels

The PANDA experiment has a rich physics program, so that the composition of active trigger signatures will be adapted depending on the current physics aim [4,5]. Therefore, the PANDA Software Trigger system needs to identify many types of physics reactions and offer a highly flexible configuration.

For this study a total of ten channels are considered to verify the feasibility of the Software Trigger inspired by the physics motivation given in the PANDA Physics Book [4]. These ten physics channels have simple and clear decay modes in their respective field, spanning a number of event topologies which may occur in the experimental runs, while

covering the main physics topics, namely exotic hadrons, charmonium, open charm, and baryon states, including the resonances  $\phi, \eta_c, J/\psi, D^0, D^+, D_s^+, \Lambda,$  and  $\Lambda_c$ .

The physics topics, the reaction physics channels together with the corresponding codes and the trigger signatures are listed in Table 1. Signal events are generated with EvtGen [12] using a simple phase space decay model (PHSP) as well as the VSS (decay of vector to two scalar particles) and VLL (decay of vector to two charged leptons) models for the decays of  $\phi$  and  $J/\psi$ , respectively. More complex decay patterns e.g. in the Dalitz decay channels are omitted in order to populate the phase space more evenly for data quality studies. Generic inelastic background reactions are generated by the Dual Parton Model (DPM) [13], which models the various production cross sections in antiproton-proton reactions. These background events have to be rejected effectively by the triggering algorithm.

For each reaction type, a specific selection procedure, the trigger line, is defined. To determine the performance, the individual trigger lines are under investigation as well as the complete trigger system for the ten reaction types to be tagged simultaneously.

A trigger line includes the reconstruction of a certain composite resonance/particle candidate and the classification whether this composite candidate originates from signal or background events. A comprehensive list of input quantities (Table 2) is computed based on reconstructed properties, momenta, angles, particle identification probabilities of the composite candidates as well as final state particles involved.

The same set of input observables is used for the direct comparison of the various approaches. Eventually, this set of observables is extended by additional event shape observ-

**Table 2** Summarised list of the standard input observables as well as additional event shape input observables. (lab: laboratory frame, cms: centre-of-mass frame)

Category	Description
Candidate kinematics	Momentum $p$ of reconstructed candidate (lab/cms)
	Transverse momentum $p_t$ of reconstructed candidate
	Polar angle $\theta$ of reconstructed candidate (lab/cms)
Daughter kinematics	Kinematic variables from each daughter of the candidate
	Electron/Muon/Kaon/Pion/Proton PID probability of each daughter
Event info and multiplicities	Maximum (transverse) particle momentum in event
	Sum of (transverse) momenta of charged particles in event (cms)
	Sum of cluster energies in EMC
	Maximum of cluster energies in EMC
	Particle multiplicities
	Minimum/maximum/sum of momenta
	Number of loose ( $P_{PID} > 0.25$ ) candidates
Event shape	Aplanarity of the event
	Sphericity of the event
	Magnitude of the event thrust vector
	Reduced Fox–Wolfram Moments 1–4

ables to determine the optimal performance achievable using all available quantities for the final design choice. These event specific observables comprise minimum and maximum momenta, momentum sums, event planarity and sphericity, thrust magnitude, Fox–Wolfram moments [14] and multiplicities are deduced, to further support the trigger decision.

One trigger line consists of four parts:

- Reconstruction of composite candidates by final state combinatorics
- Preselection with a broad cut on the invariant mass of the candidate of  $\pm 10\sigma$  around its peak position
- Calculation of the necessary observables, Table 2
- Selection by algorithm (focus of this work)

The selection target is determined by achieving the required background suppression factor. In this study, the overall required suppression factor is  $s = 1/1000$ , i.e. the number of background events is suppressed by 99.9% with all active trigger lines acting simultaneously. Applying  $n_{\text{trig}}$  trigger lines simultaneously, the background suppression for each trigger line is required to satisfy  $s_i = s/n_{\text{trig}}$ . In turn, due to the possibility of different trigger lines accepting the same background event, the actual total background suppression factor  $s_{\text{all}}$  can be a little bit better, such that  $s_{\text{all}} \leq s = \sum s_i$ .

### 3 Data preparation

The data for the physics channels and background are obtained by Monte-Carlo simulations with the PandaRoot software [15]. This simulation and reconstruction software framework is based on the publicly available software package FairRoot [16] v18.2.0 and the the external package collection FairSoft in version “jun19p1”.

EvtGen [12] provides the particles for propagation through the detector volumes with the GEANT 3 transport modeler [17], which is followed by detailed detector simulations, digitisation and reconstruction. Per particle candidate, tracking, calorimetry and PID information, such as particle energy loss, time of flight, Cherenkov angle, EMC energy deposit, is being provided. All events are self-contained and no event time relevant effects, such as event mixing or incomplete events, are considered. In Table 3, the number of simulated events is shown for the chosen channels where the centre-of-mass energy permits the reaction. Also the Monte-Carlo truth information, matched to the reconstructed particle candidates, is provided to distinguish combinatorial background from signal events for the training stage.

## 4 Methods

Our figure of merit for comparing different approaches to the triggering process will be the triggering efficiency, defined as the ratio of the number of triggered events ( $N_{\text{trig}}$ ) to the number of events passing the trigger line reconstruction ( $N_{\text{rec}}$ ):

$$\epsilon = \frac{N_{\text{trig}}}{N_{\text{rec}}}$$

while achieving a fixed background reduction. Each trigger line receives events that were passing the detector reconstruction, performs combinatorics to form resonance candidates and applies a  $10\sigma$  mass window cut as preselection, with  $\sigma$  being the width/resolution of the individual reconstructed resonance.

The triggering efficiency can be defined for two cases. Individually the efficiency focuses on the selection performance of each trigger line, serving one particular physics channel. This is a useful quantity to optimise the triggering algorithms in question. For the complete trigger setup, the total efficiencies will be affected by cross-tagging. Here events can be rejected by their intended trigger line, but are accepted accidentally by one or more of the other trigger lines. This is an unintentional but fortunate effect. The total triggering efficiency is the practically relevant measure for the experiment.

Since we require a fixed background reduction for each neural network (trigger line) individually, the resultant signal trigger efficiencies already represent a comparable measure for network performance. As an additional quality measure we provide the integral (AUC = area-under-curve) of the receiver-operator-characteristics (ROC), which is more common in Machine Learning contexts. The ROCs differ mainly in the upper corner where the background suppression is around 0.9. The signal efficiency drops quickly when the background suppression is close to 1 and high signal efficiencies at a background reduction factor of  $1/(n \times 1000)$ , with  $n$  being the number of trigger lines, correlate with the AUCs, but not always consistently.

#### 4.1 Cut-and-count method

The conventional benchmark trigger scheme is following a cut-and-count approach employing an optimised set of trigger line specific one-dimensional cuts on the measured observable distributions. This technique has been used previously to study trigger concepts for PANDA.

In order to identify this set of criteria all input observables are evaluated individually for each trigger line. We integrate the distributions of both signal and background events for each of the  $n$  observables in both directions up to a threshold value retaining 90% of the signal events. This threshold defines the cut to be applied. From these  $2n$  possible cuts, we



**Table 3** Numbers of simulated events of each physics channel at each energy, given in million events. The dashes mark the cases, where the reaction is energetically not possible

$\sqrt{s}$ [GeV]	<i>ee</i>	<i>Phi</i>	<i>Etac</i>	<i>J2e</i>	<i>J2mu</i>	<i>D0</i>	<i>Dch</i>	<i>Ds</i>	<i>Lam</i>	<i>Lamc</i>	DPM
2.4	3.1	1.3	–	–	–	–	–	–	2.4	–	20.0
3.8	2.4	1.2	6.0	2.7	1.2	1.2	1.4	–	1.3	–	20.0
4.5	2.2	1.2	6.6	2.3	1.2	1.2	1.3	1.6	1.2	–	20.0
5.5	2.2	1.2	7.7	2.1	1.2	1.2	1.2	1.3	1.2	1.7	20.0

select and apply the one leading to the largest suppression of background events. This procedure is iterated until the total background suppression matches the required factor. The last criterion in the set is adapted in the way, that the background requirement is exactly met, so that the corresponding signal efficiency can be larger than 90%.

The complete trigger configuration is given by 30 sets of one-dimensional selection criteria, one for each trigger line at each accessible centre-of-mass energy.

#### 4.2 Deep learning methods

The neural network forms its answer from the input data in a single number or a set of numbers in case of a multiple classification. Performing a selection cut on this output will determine the rejection rate of background events as well as the efficiency of signal event acceptance.

The PyTorch framework [18] is chosen to provide the underlying functionality to build the neural networks, which are trained and evaluated with the prepared simulation data. The simulated data is divided into two parts, approximately in an one-to-one ratio based on the event number. The first part undergoes Monte-Carlo truth matching preprocessing to ensure purity without undesired combinatorial effects and serves as the training set. The second part is used in two ways: one is preprocessed in the same way as the training set and used as the validation set to monitor the neural network's loss, while the other is used directly as the testing set to evaluate software trigger performance.

### 5 Neural network optimisation

Neural network setups are quite diverse and mostly tailored for the problem at hand. For the Software Trigger, several choices are made based on the performance, especially the signal efficiency and background reduction. Individual networks are optimised by:

- Network depth [19]
- Width or convolutional kernel size of each layer
- Choice of the optimiser, learning rate and related items, e.g. momentum [20],

- Choice of the activation function [21–26],
- Weight decay generalisation [27]
- Weight initialisation [28].

During training the network models are stored and the best training epoch is chosen, based on the highest signal efficiency at the fixed background rejection rate for the testing data set.

#### 5.1 Multi-class and binary approaches

The first question investigated is if a single network serving all trigger lines simultaneously (multi-class classification) performs better than a single network for each physics channel (binary classification). Allowing multi-class classification would mean fewer but bigger networks to be trained. Binary classification in contrast allows for easier adaption and modification of the list of channels. The multi-class approach reduces the number of required training experiments for each trigger setup to find the optimal network size at the cost of longer training times due to the increased complexity.

A set of Dense NNs [19] (DNN) is used to study this issue exemplarily for one setting at  $\sqrt{s} = 4.5$  GeV by the means of a Bayesian approach for the best triggering efficiency, by optimising the networks parameters iteratively starting from multiple parameter sets. In each iteration the next parameter set is being predicted by Gaussian Processes to find the optimum [29]. The input vector size is the largest number of observables amongst the trigger channels. Zero padding is used where the channel features fewer observables. Background and the 9 signal channels are labeled individually at the output of the networks. The cut on the network output is performed for each trigger line individually to achieve the required background suppression. The results are shown in Table 4, where the individual trigger efficiencies serve as a figure of merit. Here, only signal candidates that match the Monte-Carlo truth are under investigation, eliminating combinatorial effects. In those channels where the triggering efficiency is not close to 100%, the binary classification outperforms the multi-class approach by up to a factor of two.

Therefore, we choose the binary classification approach with one neural network per trigger line and energy point.

**Table 4** The comparison on the individual trigger efficiencies of truth matched events between multi-class classification and binary classification based on a DNN for an overall background rejection of 1/1000 over all trigger lines

Channels	Multi-class [%]	Binary [%]
<i>ee</i>	100.0	–
<i>Phi</i>	87.9	–
<i>Etac</i>	48.1	76.2
<i>J2e</i>	98.6	–
<i>J2mu</i>	99.9	–
<i>D0</i>	69.9	93.4
<i>Dch</i>	48.4	78.6
<i>Ds</i>	32.7	73.7
<i>Lam</i>	48.1	94.7

## 5.2 NN type selection

In order to identify the optimal network architecture and meta-configuration, seven different types of networks are studied, listed in Table 5: A dense neural network, a convolutional neural network (CNN) [30], both with and without residual blocks [31–33], a CNN with bottleneck residual blocks [34] as well as 1D and a 2D Long-Short-Term-Memory network [35].

Some of the trigger lines perform reasonably well and show stable results under all kinds of network configurations due to particularities of the decay kinematics. For example, a  $J/\psi$  decaying into two leptons will leave two strongly correlated high-momentum tracks in the detector, which is significantly different from the average multi-pion background event and in consequence always leads to high selection quality.

For meaningful optimisation, these “simple” cases are ignored, and the networks are optimised in depth and layer size for a high triggering efficiency for the three more challenging channels *Etac*, *Dch* and *Lamc*. For instance, the depth is optimised by obtaining the trigger performance as a function of used layers/blocks, and ten runs are carried out for each NN framework of a certain depth. The performance is evaluated by the individual efficiencies as well as a combined efficiency (“normalised trigger performance”  $\hat{\epsilon}$ ), defined as the geometric mean

$$\hat{\epsilon} = (\epsilon_{Etac} \cdot \epsilon_{Dch} \cdot \epsilon_{Lamc})^{1/3}$$

of the efficiencies of the channels *Etac*, *Dch* and *Lamc*.

The absolute value of  $\hat{\epsilon}$  is the primary measure of how well a network type performs. The standard deviation of the individual efficiencies from a series of training runs with varying random initialisation of the internal weights is considered as a measure for the training stability. Furthermore the networks are investigated for their robustness under changing the net-

**Table 5** The list and the description of the NNs investigated for the PANDA Software Trigger in this note

Abbreviation	Description
DNN	Dense NN
DNNRes	Dense NN with residual blocks
CNN	Convolutional NN
CNNRes	CNN with residual blocks
CNNResBN	CNN with Bottleneck residual blocks
LSTM1D	Long-Short-Term Memory with 1D input
LSTM2D	Long-Short-Term Memory with 2D input

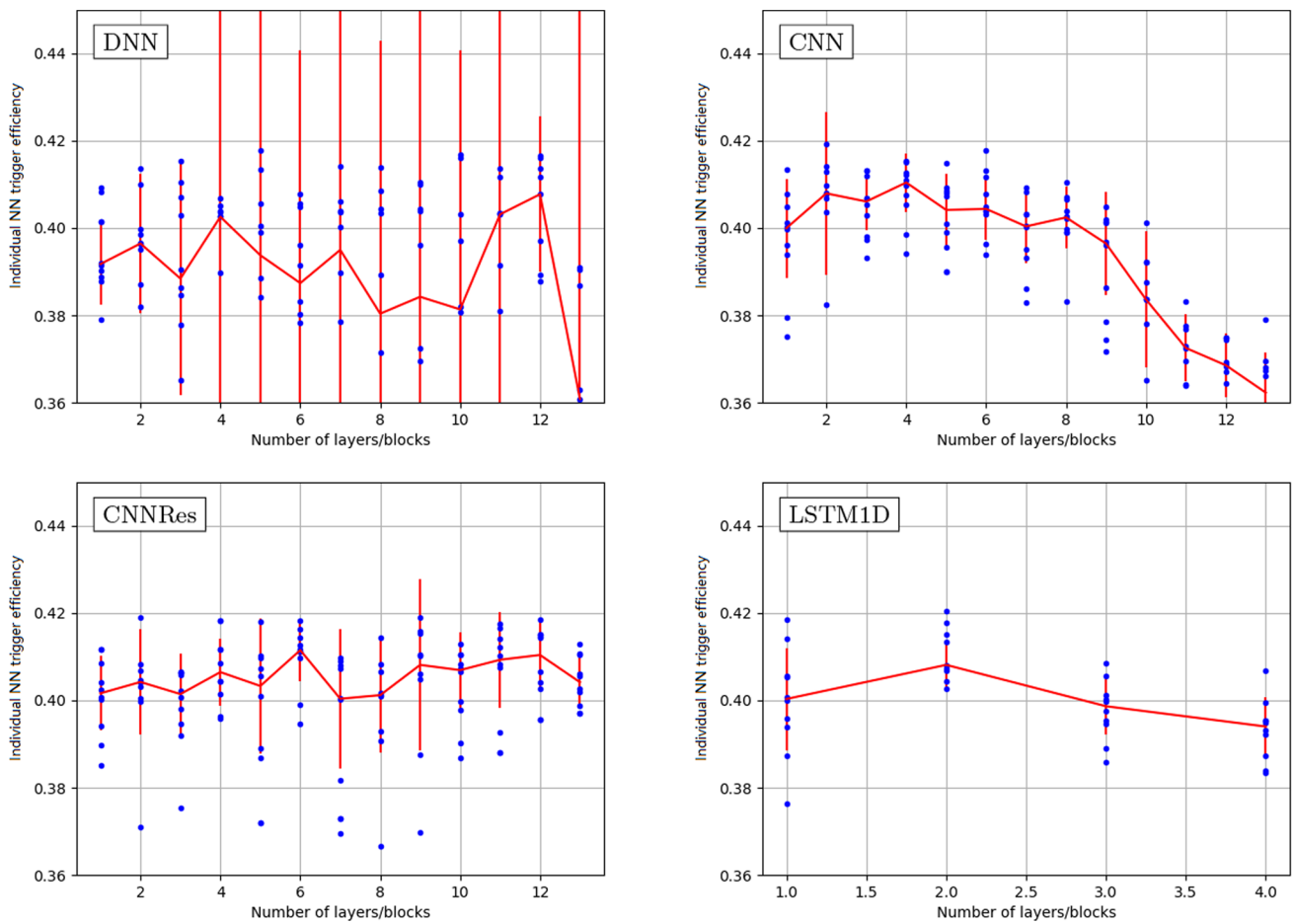
work size, in terms of the number of layers or blocks. This may be required in a future scenario for more complex triggering setups, e.g. with more measured particles, hence an increased number of input observables.

In Fig. 3, we present exemplarily four network types (DNN, CNN, CNNRes and LSTM1D) and their individual efficiencies for just the channel *Dch*. For each network size, ten networks have been trained and evaluated, providing a median and standard deviation from the individual results. The straight forward approach would be a DNN (Fig. 3, top left), however, the triggering efficiency and stability for training are inferior as they are visible from the large spread of the results. For a NN with increasing depth, in general significant degradation of performance is expected because of shattered gradients [33] as well as the risk of overtraining. This can be most clearly seen in the case of the CNN with decreasing performance using a larger number of layers (Fig. 3, top right). Using a CNN with residual blocks (Fig. 3, bottom left) mitigates this behaviour. The LSTM model also looks stable but shows a worse generalisation than CNNRes, with regard to the different physics channels (Fig. 3, bottom right)

For the selection of the best architecture the combined efficiencies  $\hat{\epsilon}$ , shown in Fig. 4, are compared. Here all considered network types are presented, which contains the information of a total of  $\approx 2200$  network training and evaluation runs, three channels times ten networks per marker. We identify the CNNRes type network (orange) as optimal for the given purpose based on its performance in triggering efficiency as well as training stability and robustness concerning the varying sizes.

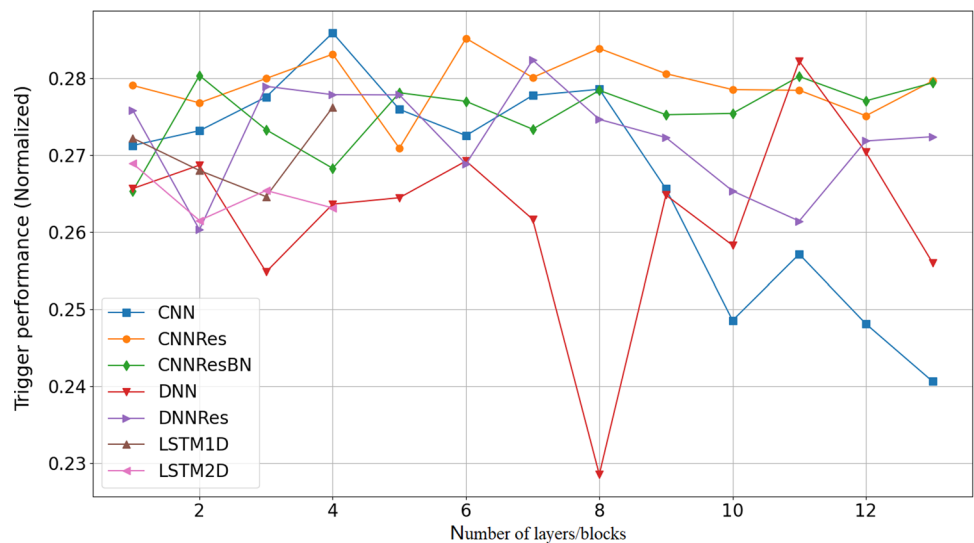
## 5.3 Chosen network architecture

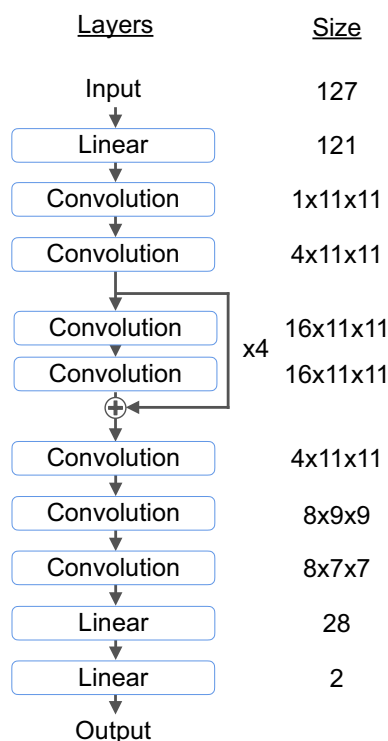
The final choice to evaluate the performance of a neural network approach for the PANDA Software Trigger is the CNNRes, a Convolutional Neural Network with four residual blocks. It is “deep” enough to ensure that the NN has enough capacity, while performing relatively stable when adding more blocks, which may be considered within a pro-



**Fig. 3** Individual trigger efficiencies (blue dots) for the channel *Dch* at 5.5 GeV/c for four of the seven NN types with varying depth parameters as well as the median with standard deviation (red makers and bars) from varying the weight seeds

**Fig. 4** Normalised triggering performance  $\hat{\epsilon}$  as function of the used layers/blocks in the seven network types, combined for the three channels *Etac*, *Dch* and *Lamc* at 5.5 GeV/c





**Fig. 5** Schematic view of the chosen network architecture

duction triggering environment, e.g. featuring many more trigger lines.

In Fig. 5, the details of the network are presented. In the first stage the 127 input channels are mapped to a  $11 \times 11$  matrix with a fully connected 121 large linear layer in between. Here, the network has the possibility to sort any linear combination of observables next to each other, which will enhance the performance of the image recognition type architecture to follow. Then two convolutional layers extend the dimensionality to  $16 \times 11 \times 11$ , allowing for a detailed feature extraction performed by four residual blocks - pairs of convolutional layers with a residual path. Three convolutional and two linear layers then reduce the dimensions and perform the classification for the output. Between each layer batch normalisation [36] is being performed.

## 6 Results and discussion

### 6.1 Individual trigger performance

Measuring the performance of the neural network approach is done by the individual trigger efficiency determined per channel with that corresponding single trigger line active. The cuts on the network output are tuned to achieve a background suppression of 1/1000 in total, with an equal fraction of background contribution by each trigger channel for

a certain centre-of-mass energy. For example at the energy of  $\sqrt{s} = 4.5$  GeV with nine of the ten channels being energetically accessible, the targeted background suppression per channel is 1/9000. In the real experiment the balance of background suppression between various channels might be optimised. Here we choose equal sharing. Another commonly established measure of the network capabilities is the ROC curves and, more condensed, the corresponding AUCs. Both, the individual triggering efficiencies and the network AUCs are summarised in Table 6.

In all cases the triggering with the aid of the neural networks improves the efficiencies compared to the cut-and-count approach. This improvement is of course expected, mainly because possible correlations between observables are exploited by the new algorithm. Figure 6 shows the relative efficiency gain, where the bars in light colours represent the increase based on the same set of input variables, the dark coloured bars the performance gain using the extended set of observables listed in Table 2. It reaches almost up to 200% for some of the channels, corresponding to about a factor of three in performance. The actual benefit strongly depends on the channel in question. For example the channel  $ee$  has already a very good triggering efficiency from the cut-and-count approach, leaving no room for improvements. In the case of the two  $J/\psi$  channels  $J2e$  and  $J2mu$ , the triggering efficiency is dropping with increasing beam momentum (see Table 6). This effect can be explained by the stronger background suppression requirements as more channels are being triggered, which is almost completely recovered by the new approach. For the open charm and  $Etac$  and  $Lamc$  channels the triggering efficiency has about doubled.

Adding more observables, describing the overall event topology, leads in many cases to a substantial gain in triggering efficiency. Hence, in a future software trigger setup it would be desirable to provide these event shape observables, even if computationally expensive in an online environment. Based on the AUC values all networks show good (0.8–0.9) to excellent ( $> 0.9$ ) classification performance.

### 6.2 Simultaneous trigger performance

In a realistic setup, all active trigger lines have the ability to trigger events simultaneously. Therefore it can happen, that a signal event of a certain type missed by the dedicated trigger line is accidentally tagged by another one. This cross tagging by simultaneous triggering has the potential to further improve the triggering efficiency. Nevertheless, at the same time the background level will not exceed the requirements.

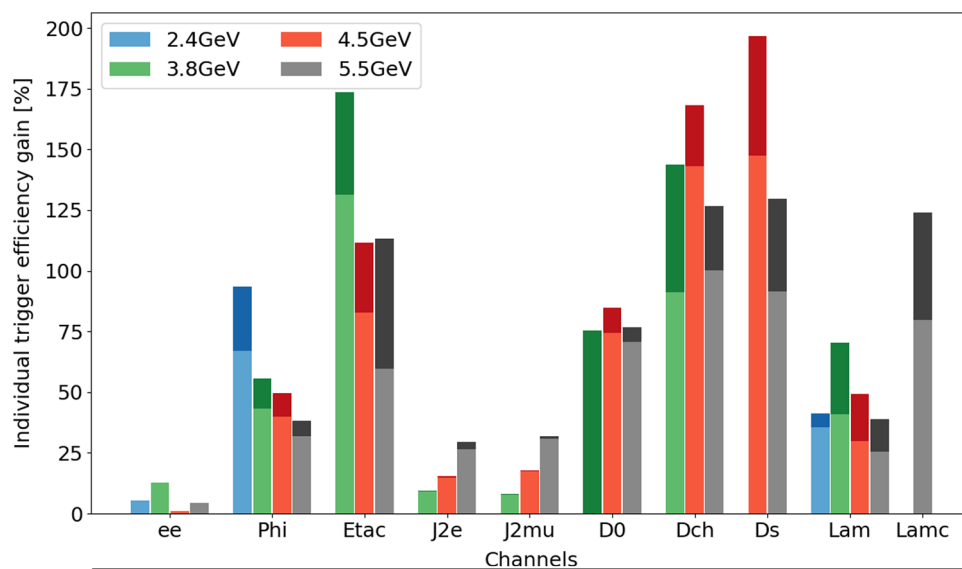
Figure 7 quantifies this effect in the current scenario, showing the increase in the triggering efficiencies, also presented in Table 6, which are calculated as the fraction of triggered events from events that passed the reconstruction in the trigger line designed for the channel. The light, medium and



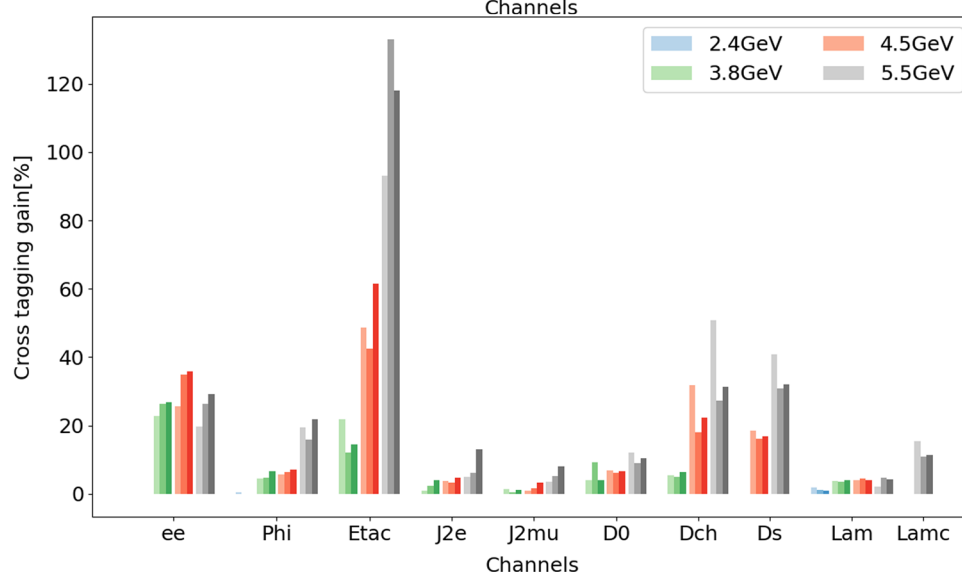
**Table 6** Collection of individual trigger efficiency results for a target 1/1000 background suppression, the AUC values of the ROC curves as well as the simultaneous trigger efficiency gains for all accessible channels

eff [%]	<i>ee</i>	<i>Phi</i>	<i>Etac</i>	<i>J2e</i>	<i>J2mu</i>	<i>D0</i>	<i>Dch</i>	<i>Ds</i>	<i>Lam</i>	<i>Lamc</i>
<b>Individual trigger efficiencies [%]</b>										
Conventional trigger										
2.4 GeV	94.79	38.09	–	–	–	–	–	–	21.50	–
3.8 GeV	88.80	54.64	9.71	91.34	92.52	39.50	12.50	–	40.25	–
4.5 GeV	98.88	56.24	11.92	85.85	84.62	31.42	10.19	6.38	49.21	–
5.5 GeV	95.94	61.47	17.87	76.31	75.09	36.79	17.71	11.87	56.27	6.59
NN trigger										
2.4 GeV	99.98	63.67	–	–	–	–	–	–	29.14	–
3.8 GeV	99.98	78.34	22.45	99.53	99.75	39.54	23.89	–	56.67	–
4.5 GeV	99.99	78.71	21.77	98.46	99.40	54.77	24.77	15.78	63.81	–
5.5 GeV	99.99	80.94	28.53	96.44	98.35	62.74	35.43	22.74	70.55	11.85
NN trigger with additional event shape observables										
2.4 GeV	100.00	73.72	–	–	–	–	–	–	30.33	–
3.8 GeV	100.00	85.05	26.56	99.76	99.88	69.30	30.48	–	68.52	–
4.5 GeV	100.00	84.13	25.21	99.21	99.64	58.01	27.32	18.92	73.47	–
5.5 GeV	100.00	84.95	38.10	98.86	98.90	65.02	40.16	27.26	78.19	14.75
<b>Individual NN performance</b>										
AUC values for the NN trigger										
2.4 GeV	1.000	0.948	–	–	–	–	–	–	0.978	–
3.8 GeV	1.000	0.987	0.869	0.999	0.999	0.960	0.854	–	0.987	–
4.5 GeV	1.000	0.988	0.912	0.998	0.999	0.951	0.866	0.865	0.990	–
5.5 GeV	1.000	0.986	0.929	0.996	0.998	0.936	0.890	0.852	0.992	0.821
AUC values for the NN trigger with the additional event shape observables										
2.4 GeV	1.000	0.959	–	–	–	–	–	–	0.977	–
3.8 GeV	1.000	0.988	0.887	0.999	1.000	0.963	0.838	–	0.990	–
4.5 GeV	1.000	0.990	0.913	0.998	0.999	0.956	0.877	0.890	0.994	–
5.5 GeV	1.000	0.989	0.941	0.999	0.998	0.955	0.900	0.873	0.994	0.834
<b>Cross-tagging efficiency gains as effect of simultaneous triggering [%]</b>										
Conventional trigger										
2.4 GeV	0.01	0.43	–	–	–	–	–	–	1.96	–
3.8 GeV	22.74	4.52	21.93	0.83	1.45	4.12	5.36	–	3.74	–
4.5 GeV	25.53	5.68	48.65	3.68	1.00	6.94	31.82	18.51	4.12	–
5.5 GeV	19.68	19.37	93.05	5.01	3.53	12.05	50.72	40.75	2.18	15.50
NN trigger										
2.4 GeV	0.00	0.00	–	–	–	–	–	–	1.26	–
3.8 GeV	26.39	4.75	12.17	2.28	0.47	9.36	5.02	–	3.57	–
4.5 GeV	34.92	6.51	42.59	3.42	1.75	6.12	18.13	16.22	4.58	–
5.5 GeV	26.37	15.80	132.98	6.22	5.11	9.08	27.23	30.85	4.71	10.99
NN trigger with additional event shape observables										
2.4 GeV	0.01	0.00	–	–	–	–	–	–	0.98	–
3.8 GeV	26.93	6.72	14.57	3.92	1.23	4.13	6.48	–	4.10	–
4.5 GeV	35.89	7.22	61.43	4.71	3.22	6.54	22.28	16.78	4.08	–
5.5 GeV	29.18	21.76	118.00	12.93	8.09	10.37	31.27	32.09	4.15	11.46

**Fig. 6** Individual triggering efficiency gains of the neural networks compared to the conventional approach, based on the same set of input observables (light colours), and on the extended set of observables (dark colours)



**Fig. 7** Cross-tagging efficiency gain as effect of simultaneous triggering of multiple reactions, for the conventional benchmark approach (light colours), the NN approach with the same set of observables (medium colours), and the NN approach with the extended set of observables (dark colours)



dark coloured bars correspond to the conventional cut-and-count benchmark algorithm, the neural network approach and the neural network approach based on the extended variable set, respectively. In some cases this adds a substantial increase in triggering efficiency. For example the channel *Etac* at 5.5 GeV gains a factor of two in total triggering efficiency by cross tagging in this particular setup. It is important to note that these “accidental” gains are highly dependent on the actual composition of trigger lines, the centre-of-mass energy, and the actual trained networks.

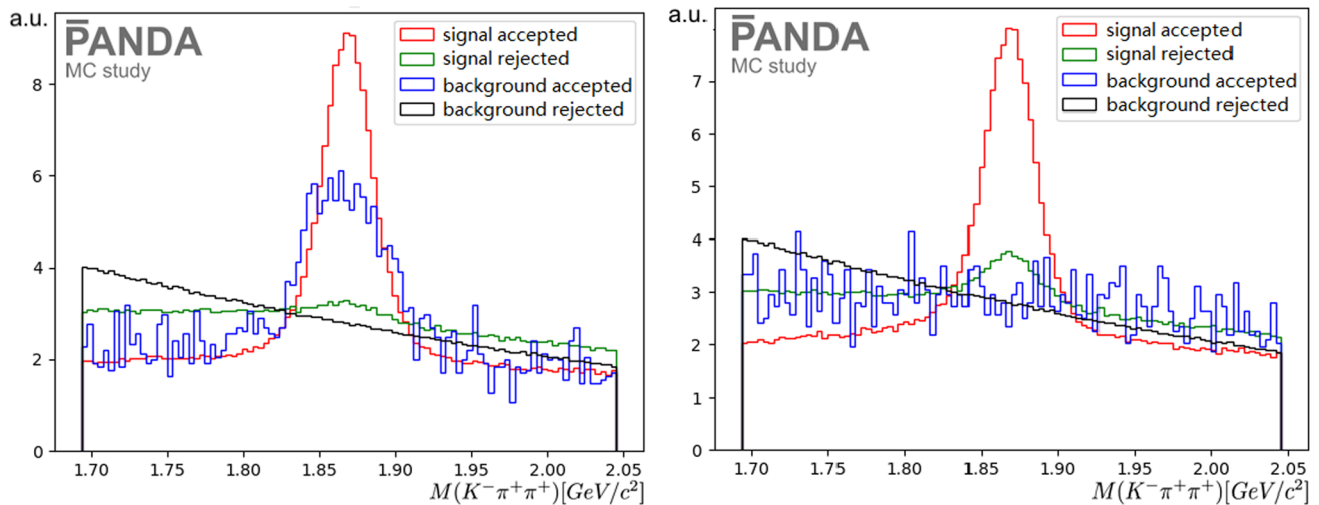
### 6.3 Computing performance

The bulk of training processes goes into finding the optimal solution for each network architecture, which is time consuming and requires the intensive usage of GPU and CPU resources. A dedicated server with one Intel<sup>®</sup> Xeon<sup>®</sup> W-2135 CPU with 3.70 GHz and one NVIDIA<sup>®</sup> 3090 GPU is

used, equivalent in performance to 171 cores of the AMD<sup>®</sup> EPYC<sup>®</sup> 7551 32-Core Processors on the computing cluster at GSI. Processing times are in the order of two days for a single trigger line on 12 cores of the computing cluster. The model inference is able to run in the order of 1.4 M candidates per second on the GPU server with six trigger lines in parallel. The main bottleneck is the CPU based data reading and preparation. Since the training can be performed in parallel for the trigger lines this is quite promising that no dedicated GPU hardware is strictly necessary to train new trigger lines. On the other hand the processing speed in the inference on GPUs makes this option interesting for an online use scenario before the data stream reaches the main computing cluster.

### 6.4 Data quality and choice of observables

The input features and observables have varying impacts on the trigger decision. Some features are correlated, some are



**Fig. 8** Candidate mass distributions of accepted and rejected signal (red and green, respectively) and background (blue and black, respectively) for the channel  $Dch$  at 4.5 GeV/c. Left, all kinematic input

observables, right without the energy of the candidate and the invariant mass of the recoiling system. The histograms are normalised to the same integral

not relevant to the classification problem at hand, which differs between the channels and even between the data sets of the same channel at different beam energies. The ranking of the observable importance can guide the observable choice and thus may reduce the required computations in the online processing before.

Furthermore, it is important to maintain a certain quality of the data during the selection process, which in turn means that the underlying physics constrains the choice of observables.

The  $Dch$  channel ( $D^+ \rightarrow K^- \pi^+ \pi^+$ ) for example has a significant enhancement in the invariant mass distribution of background events around the signal peak position (blue histogram in Fig. 8, left) when using both the energy of the  $D^+$  candidate and the invariant mass of the recoil system at the same time as input for the NN. This effect originates from the specific kinematics of the two body reaction  $\bar{p}p \rightarrow D^+ D^-$  considered here. However, it is the goal to trigger  $D^+$  particles from many different reactions inclusively. Dropping these observables from the NN input removes the undesirable peaking background (Fig. 8, right) at the cost of some signal efficiency. We observed now a smooth distribution (blue histogram) in the signal region similar to the rejected background (black histogram).

When analysing complex decay patterns and mixtures of resonances, it is important to cover the phase space without steep drops or holes in the triggering efficiency distributions introduced by the triggering algorithm. Complex fitting algorithms, such as a partial-wave analysis [37], benefit most from a flat triggering efficiency distribution if possible, but they require at least a smooth dependency on any kinematic observable.

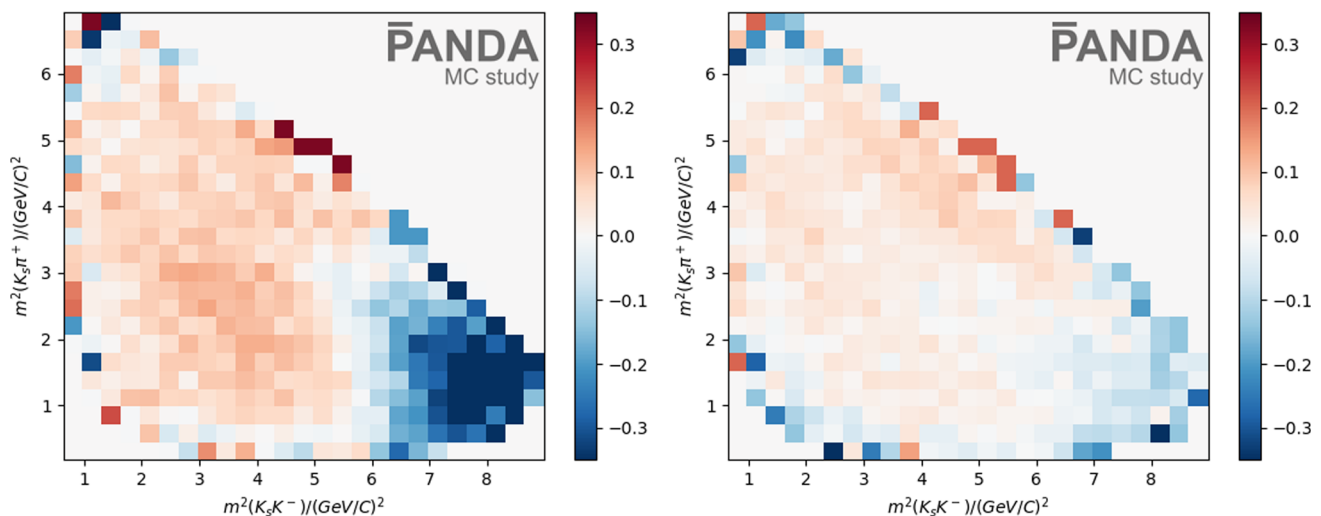
For example a three-body-decay, such as the channel  $\eta_c \rightarrow K_s K^- \pi^+$ , could be studied in the Dalitz plot [38] representation. We find that the relative efficiency of the neural network trigger introduces a cut-off in one corner of the Dalitz plot (Fig. 9, left) with the observables that are used in the cut-and-count approach. It could be that the network identified this corner to be particularly occupied with background. Introducing the event shape observables (Table 2) as additional input, the training is able to produce a significantly more flat triggering efficiency (Fig. 9, right), being beneficial for later Dalitz plot analysis.

## 7 Summary and outlook

Based on a set of ten channels with typical event topologies in the reach of PANDA physics investigated at four anti-proton beam energies, it is demonstrated, that PANDA will greatly benefit from a neural network supported software trigger system.

As one result the use of single networks per trigger channel outperforms an approach with multi-class classification. As an additional advantage, it will make the setup much easier to extend the system with a larger set of simultaneous trigger lines and improves the computational scalability. From seven network architectures, the convolutional network with four residual blocks showed the best results for the three channels with the lowest triggering efficiency, while producing stable results under different training attempts and size changes.

In all cases the network approach performs better than the cut-and-count method, which further improves when adding more observables. In the comparison the triggering efficien-



**Fig. 9** Trigger efficiency deviation from the mean value in Dalitz coordinates for the channel  $Etac$  at 3.8 GeV/c, without (left) and with (right) the additional observables

cies show gains of up to 200% and all networks show good up to excellent performance.

Data quality is an important topic that deserves to be studied in more detail. It would be desirable to give feedback about e.g. the triggering efficiency flatness in the Dalitz plot coordinates to the neural networks during the training process. This also holds for the background flatness in critical distributions, such as the candidate invariant mass in order to avoid peaking structures.

For the study presented here, the background suppression requirement is set equally among the physics channels. However, this approach neglects the differences in reconstruction and varying background levels in the different channels. Finding an approach to achieve the best background suppression requirements for each trigger line, while maintaining the desired overall background reduction, is a complex challenge. Enhancing the triggering efficiency of one channel will reduce the triggering efficiency of another channel, which has to be carefully balanced in the future PANDA experiment.

**Acknowledgements** We acknowledge the internal review work of the accompanying analysis memo by A. Belias, M. Papenbrock and T. Stockmanns, and we thank all members of the PANDA Collaboration. This work has been supported by the Bundesministerium für Bildung und Forschung (BMBF), Germany; the GSI Helmholtzzentrum für Schwerionenforschung GmbH, Germany; the Helmholtz Forschungsakademie Hessen für FAIR (HFHF), Germany; the Helmholtz-Institut Mainz, Germany; the Institute of Modern Physics, Chinese Academy of Sciences, China. And in particular, P. Jiang has received financial support via the Helmholtz – OCPC Postdoc Program and the Office of China Postdoc Council, Ministry of Human Resources and Social Security, China.

**Data Availability Statement** This manuscript has no associated data or the data will not be deposited. [Authors' comment: The datasets

generated during and/or analysed during the current study are available from the corresponding author on reasonable request.]

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Funded by SCOAP<sup>3</sup>. SCOAP<sup>3</sup> supports the goals of the International Year of Basic Sciences for Sustainable Development.

## References

1. P. Spiller, G. Franchetti, The FAIR accelerator project at GSI. *NIMA* **556**, 305–309 (2006)
2. M. Destefanis for the PANDA Collaboration, The PANDA experiment at FAIR. *Nucl. Phys. B (Proc. Suppl.)* **245**, 199–206 (2013)
3. A. Belias, FAIR status and the PANDA experiment. *JINST* **15**, C10001 (2020)
4. PANDA Collaboration, Physics Performance Report for PANDA: Strong Interaction Studies with anti-protons, FAIR/PANDA/Physics Book, pp. 28–30 (2009)
5. G. Barucca, F. Davi, G. Lancioni et al., PANDA phase one. *Eur. Phys. J. A* **57**, 184 (2021)
6. Particle Data Group, R.L. Workman, V.D. Burkert et al., Review of particle physics. *PTEP* **2022**, 083C01 (2022)
7. S. Acharya et al., ALICE Collaboration, *Comput. Phys. Commun.* **242**, 25–48 (2019)
8. R. Aaij, M. Alexander, A. Albero et al., Design and performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC. *JINST* **14**(04), P04013 (2019)

9. E. Aprile, J. Aalbers, F. Agostini et al., The XENON1T data acquisition system. *JINST* **14**(07), P07016 (2019)
10. B. Friman, C. Hohne, J. Knoll, The CBM physics book: compressed baryonic matter in laboratory experiments. *Lect. Notes Phys.* **814**, 1–980 (2011)
11. A. Lehrach, O. Boine-Frankenheim, F. Hinterberger et al., Beam performance and luminosity limitations in the high-energy storage ring (HESR). *Nucl. Instrum. Meth. A* **561**, 289 (2006)
12. D.J. Lange, The EvtGen particle decay simulation package. *Nucl. Instrum. Meth. A* **462**, 152–155 (2001). [https://doi.org/10.1016/S0168-9002\(01\)00089-4](https://doi.org/10.1016/S0168-9002(01)00089-4)
13. A. Capella, U. Sukhatme, C.-I. Tan et al., Dual parton model. *Phys. Rep.* **236**, 225–329 (1994)
14. A.J. Bevan et al., The physics of the B factories. *Eur. Phys. J. C* **74**, 3026 (2014)
15. S. Spataro for the PANDA Collaboration, The PandaRoot framework for simulation, reconstruction and analysis. *J. Phys. Conf. Ser.* **331** (2011)
16. M. Al-Turany, D. Bertini, R. Karabowicz et al., The FairRoot framework. *J. Phys. Conf. Ser.* **396**, 022001 (2012)
17. CERN Program Library Long Writeup W5013 (1993, last retrieved Oct. 2022). <https://cds.cern.ch/record/1073159/files/cer-002728534.pdf>
18. A. Paszke, S. Gross, C. Soumith et al., Automatic differentiation in PyTorch, NIPS-W (2017)
19. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015)
20. S. Ruder, An overview of gradient descent optimizer. [arXiv:1609.04747](https://arxiv.org/abs/1609.04747) [cs.LG] (2016)
21. J. Han, C. Moraga, The influence of the sigmoid function parameters on the speed of backpropagation learning, IWANN. From Natural to Artificial Neural Computation, pp. 195–201 (1995)
22. V. Nair, G. Hinton, Rectified Linear Units Improve Restricted Boltzmann Machines, Proc. 27th ICML, pp. 807–814 (2010)
23. A. Maas, A. Hannun, A. Ng, Rectifier non-linearities improve NN acoustic models. *Proc. ICML* **30**(1), 3 (2013)
24. D. Hendrycks, K. Gimpel, Gaussian Error Linear Units (GELUs) (2016)
25. G. Klambauer, T. Unterthiner, A. Mayr et al., Self-normalizing neural networks. *Proc. NIPS17*, pp. 972–981 (2017)
26. X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks. *Proc. AISTATS*, PMLR **15**, 315–323 (2011)
27. A. Krogh, J. Hertz, *A Simple Weight Decay Can Improve Generalization*, *Advances in Neural Information Processing Systems*, vol. 4 (Morgan-Kaufmann, Burlington, 1991)
28. X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks. *Proc. ICAIS*, PMLR **9**, 249–256 (2010)
29. J. Mockus, *Bayesian Approach to Global Optimisation: Theory and Applications* (Kluwer Academic, Dordrecht, 2012), pp.79–116
30. K. O’Shea, R. Nash, An introduction to convolutional neural networks. [arXiv:1511.08458](https://arxiv.org/abs/1511.08458) [cs.NE] (2015)
31. K. He, X. Zhang, S. Ren et al., Deep Residual Learning for Image Recognition, CVPR, pp. 770–778 (2016)
32. S. Xie, R. Girshick, P. Dollár et al., Aggregated Residual Transformations for Deep Neural Networks, CVPR, pp. 1492–1500 (2017)
33. D. Balduzzi, M. Frean, L. Leary et al., The shattered gradients problem: if resnets are the answer, then what is the question? *Proc. ICML*, PMLR **70**, 342–350 (2017)
34. M. Lin, Q. Chen, S. Yan, Network in network (2013)
35. S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
36. S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift. *Proc. ICML*, PMLR **37**, 448–456 (2015)
37. K.J. Peters, A primer on partial wave analysis. *Int. J. Mod. Phys. A* **21**, 5618–5624 (2006)
38. R.H. Dalitz, Decay of  $\tau$  mesons of known charge. *Phys. Rev.* **94**, 1046–1051 (1954)