

RESEARCH ARTICLE | JUNE 28 2023

Neuro-explicit semantic segmentation of the diffusion cloud chamber

Nicola J. Müller ; Daniel Porawski ; Lukas Wilde ; Dennis Fink ; Guillaume Trap ; Annika Engel ; Georges P. Schartz  

 Check for updates

Rev. Sci. Instrum. 94, 063304 (2023)

<https://doi.org/10.1063/5.0109284>



View
Online



Export
Citation

CrossMark



Optimize
Your Research



Use the Optimal Technology for All Vacuum Requirements

PFEIFFER  VACUUM

Neuro-explicit semantic segmentation of the diffusion cloud chamber

Cite as: Rev. Sci. Instrum. 94, 063304 (2023); doi: 10.1063/5.0109284

Submitted: 12 July 2022 • Accepted: 3 June 2023 •

Published Online: 28 June 2023



View Online



Export Citation



CrossMark

Nicola J. Müller,^{1,2,a)} Daniel Porawski,^{1,b)} Lukas Wilde,^{1,c)} Dennis Fink,^{3,d)} Guillaume Trap,^{3,4,a)}
Annika Engel,^{2,f)} and Georges P. Schmartz^{2,g)}

AFFILIATIONS

¹Bachelor's Program Data Science and Artificial Intelligence, Saarland University, Saarbrücken 66123, Germany

²Chair for Clinical Bioinformatics, Saarland University, Saarbrücken 66123, Germany

³Luxembourg Science Center, Differdange 4573, Luxembourg

⁴Foundation Jeunes Scientifiques Luxembourg, 40 Boulevard Pierre Dupong, L-1430 Luxembourg

^{a)}s8namuel@stud.uni-saarland.de

^{b)}s8dapora@stud.uni-saarland.de

^{c)}s8luwild@stud.uni-saarland.de

^{d)}dennis.fink@science-center.lu

^{e)}trap@science-center.lu

^{f)}annika.engel@uni-saarland.de

^{g)}Author to whom correspondence should be addressed: georges.schmartz@ccb.uni-saarland.de

ABSTRACT

For decades, in diffusion cloud chambers, different types of subatomic particle tracks from radioactive sources or cosmic radiation had to be identified with the naked eye which limited the amount of data that could be processed. In order to allow these classical particle detectors to enter the digital era, we successfully developed a neuro-explicit artificial intelligence model that, given an image from the cloud chamber, automatically annotates most of the particle tracks visible in the image according to the type of particle or process that created it. To achieve this goal, we combined the attention U-Net neural network architecture with methods that model the shape of the detected particle tracks. Our experiments show that the model effectively detects particle tracks and that the neuro-explicit approach decreases the misclassification rate of rare particles by 73% compared with solely using the attention U-Net.

© 2023 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0109284>

INTRODUCTION

Diffusion cloud chambers were among the first devices to allow continuous visual detection of charged particles.¹ They played an important role in subatomic physics until the end of the 1950s.² As they provide a relatively large horizontal visualization area, they mainly served in research done with accelerators but were also precious in cosmic rays studies.^{3,4} Nowadays, diffusion chambers are still widely used in education,⁵ be it in schools, universities, or museums.

Diffusion cloud chambers have many advantages: they are quite popular, compact, movable, sensitive, have high granularity, and can be operated continuously for long durations, which enables live visualization, contrary to classical cloud chambers that had to be

triggered by outer detectors. Yet, as was also the case with bubble chambers or emulsions, cloud chambers always suffered from the fact that they needed human scanning of the data, which is why all these techniques have been supplanted by wire chambers in the 1970s. Nowadays, particle physics relies heavily on the latest development in computer science and, in particular, Artificial Intelligence (AI).⁶ For instance, Thomadakis *et al.*⁷ contributed to the field by leveraging various neural networks, including convolutional architectures, to enhance the speed and accuracy of particle reconstruction for a CLAS12 detector. The practice of employing convolutional neural networks in particle physics was also evidenced by the work of Aurisano *et al.*,⁸ who applied them to the NOvA neutrino detector for identifying neutrino interactions based on their topology. Furthermore, Madrazo *et al.*⁹ represented the measured

variables of particles detected in the Compact Muon Solenoid experiment as images, which enabled them to use convolutional neural networks to differentiate the production of pairs of quarks top anti-top from other processes. Following this trend, in this work, we present and share with the community an AI model allowing us to easily digitalize diffusion cloud chamber data.

Beyond obvious applications in the education sector (helping observers identify tracks is the reason for which this algorithm was developed initially), we suspect it could have other applications in particle/nuclear physics. Given its high granularity, a diffusion cloud chamber equipped with this code allows for the continuously processing great amounts of data. In combination with other devices, our approach could, for instance, contribute to study the cores of cosmic ray air showers, and add a new way to verify the accuracy of current simulation tools.¹⁰

Typically, a cloud chamber is composed of a hollow container filled with supersaturated alcohol vapor. When charged particles pass through the chamber, they collide with surrounding molecules, thus ionizing them and creating condensation centers that attract alcohol molecules. The resulting condensation then leads to fog along the particle's path. The well-known morphologies (length, thickness, curvature) of these tracks allow the observer to infer the nature of the underlying particle (e.g., Sec. 34 in Ref. 11).

The most common types of particle tracks visible at sea level with this device are the following [Figs. 1(a)–1(d)]:

- **Leptons:** muons, electrons/positrons. Even though muons are abundant at sea level, their very thin tracks make them difficult to isolate with our system, so for this work, they will be considered as a contribution to the background noise. Electrons generate thicker tracks that are easier to detect, which are twisted or curved due to their multiple collisions. We shall focus on them for the leptons in the rest of this work. As no magnetic field was applied, positrons are indistinguishable from electrons, so we will always refer to “electron tracks.” Most of these electrons are from the leptonic component of cosmic ray air showers. The rest comes from ambient beta decays and Compton scatterings.
- **Alpha particles:** They produce thick, straight, and short tracks that are easy to identify. They are derived from natural radon in the air as well as from decaying descendant nuclei in the thoron decay chain (see below).

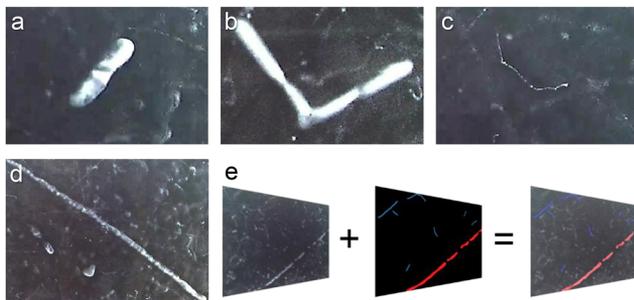


FIG. 1. Types of particle tracks and their segmentation. (a)–(d) An alpha particle track, a V track, an electron track, and a proton track. (e) Semantic segmentation of a cloud chamber image.

- **Couples of alpha particles:** In the thoron chain, the two first alpha particles are typically emitted within a tenth of a second in random directions, which results in what we will refer to as “V tracks.”
- **Protons:** They produce long straight tracks, slightly thinner than alpha particles. They come from the hadronic component of air showers.

Despite their simple features, these particle tracks can vary significantly. Additionally, the tracks quickly start to fade after their appearance, causing them to become deformed and develop gaps. Furthermore, recently vanished particle tracks can leave small regions of the cloud chamber depleted of alcohol, which temporarily hinders condensation, and thus may lead to new particle tracks appearing with gaps. The tracks may also appear truncated if the corresponding particle passed the condensing zone only partially. Background noise further complicates the detections.

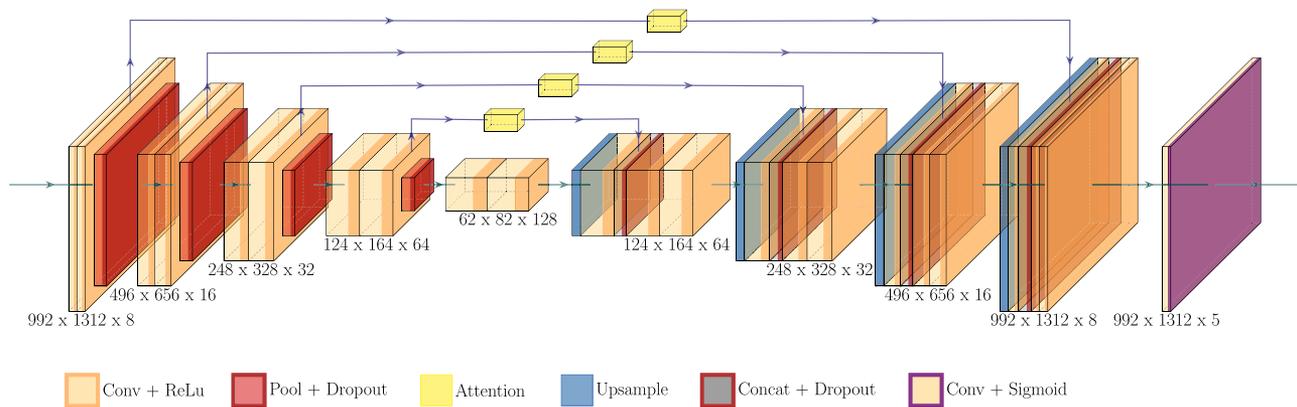
To facilitate this particle identification, we developed a model capable of automatically segmenting the cloud chamber images according to the different types of particle tracks (Fig. 5). Previous work by Barzon proposed a neural network-based solution that was designed to detect the presence of particle tracks.¹² However, his convolutional neural network could not detect and classify individual tracks. Newer methods from the field of Computer Vision, such as the U-Net architecture,¹³ allow performing both tasks with one network architecture.¹⁴ With this end-to-end approach, the U-Net, as originally conceived, achieved state-of-the-art performance in biomedical image segmentation tasks while using only a relatively small training dataset. Oktay *et al.* extended the U-Net architecture with attention gates to suppress information from the input image irrelevant to the task and focus on useful features, thus achieving higher segmentation performance.¹⁵ Despite the enormous success of the U-Net^{16,17} and its variants^{18–23} in complex image segmentation tasks, they suffer from the various drawbacks of neural networks: their complex structure and their massive number of parameters decrease interpretability, and they require vast amounts of data. On the contrary, Computer Vision algorithms that use human-designed rules or features are inherently interpretable and only rely on a small number of parameters, but they lack the performance of neural networks.²⁴

Trying to combine the best of both worlds, researchers attempt to combine neural networks and predefined rules into what is called neuro-explicit models.^{25–27} Following this hybrid approach, we were able to design an interpretable high-accuracy model that extends the functionality of existing work. Furthermore, we displayed the synergism of the neuro-explicit approach by comparing it against neural and explicit models.

METHODS

Cloud chamber

For this work, we used the large diffusion cloud chamber installed in early 2020 at the Luxembourg Science Center. This device is notably equipped with a system to inject thoron (Rn-220) into the visualization area that visitors can activate. Its visualization area is 1 square meter in size with an active thickness of about 1 cm high, making it, as far as we know, the largest diffusion cloud chamber currently in operation. (The largest diffusion cloud chamber ever

**FIG. 2.** Schematic of the attention U-Net architecture.

built seems to have been 2 square meters in size.²⁸) For data collection, a camera is mounted above the center of the square-shaped chamber, recording video with a resolution of 1920 by 1080 pixels at 30 frames per second.

Neural network architecture

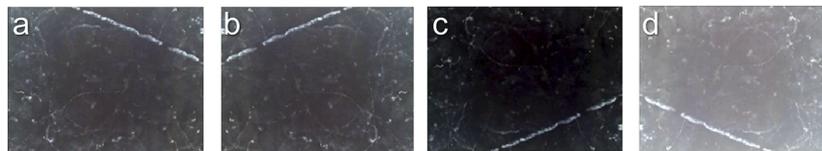
Our goal is to automatically color the particle tracks according to the particle that created them. This procedure may be subdivided into two steps: Particle track detection and particle track classification. We achieve this by training an attention U-Net for semantic segmentation,^{13,15} a type of neural network that was shown to achieve high performance and sample efficiency, making it a suitable architecture for our purposes. The attention U-Net architecture is a fully convolutional neural network, resembling an encoder-decoder structure²⁹ (Fig. 2). The network's encoder part significantly reduces the input's dimensionality through multiple blocks containing convolutional layers and pooling layers. Afterward, the network's decoder part receives the output of the encoder and restores the input to its original dimensionality through multiple blocks containing upsampling layers and convolutional layers.

This compression and decompression of the input forces the network to learn how to extract a compact representation of the information contained in the input to construct a high-dimensional output that minimizes the loss. The encoder layers are connected to the decoder layers with the same dimensionality via skip connections.³⁰ These skip connections allow the flow of spatial information from the encoder to the decoder. Each skip connection has an attention gate³¹ that ensures no redundant low-level features are sent from the encoder to the decoder.

Given an input image from a cloud chamber, our implementation of the attention U-Net predicts five different labels: alpha particle track, electron track, proton track, V track, and background. For each label, the model returns a mask containing the probability of belonging to that class for every pixel. We then combine these five masks into a single RGB mask by assigning every pixel the color of the class with the highest probability. Afterward, we lay the RGB mask over the original image to get the annotated image [Fig. 1(e)].

Data

We start building the dataset by extracting one frame per second out of 20 h of video footage provided by the Luxembourg Science Center and then searching those frames for suitable training images. In the next step, we reduce the size of the training images from 1920 by 1080 to 1312 by 992 since our analysis shows that these border regions contain the most background noise. Afterward, the masks are created by manually annotating the training images using the online annotation tool Hasty.ai. Despite the time-consuming nature of manual annotation, we could annotate 483 frames out of our video footage. The training and test sets are then built by randomly moving 10% of the samples in the whole dataset to the test set. In the last step, we use data augmentation³² to increase the size of both datasets by a factor of 8: For every original sample, we create three variants by flipping the original vertically, horizontally, and vertically and horizontally. Then, we create four more variants for the flipped variants and the original by randomly increasing or decreasing the image's brightness [Figs. 3(a)–3(d)]. These augmentation methods are suitable since particle tracks vary in orientation

**FIG. 3.** Data augmentation. (a) Example of an original image from the dataset. (b)–(d) The data augmentation creates new versions of the original image by flipping it and changing the brightness.

and brightness. The resulting training and test sets consist of 3472 and 392 samples.

To estimate the number of unique events per particle species in our training and test sets, we developed algorithms that use the predictions of our best-performing model to identify particle tracks across subsequent frames that are created by the same event. The results show the training set contains ~1150 alpha particle events, 320 proton events, 140 thoron decay events, and 3130 electron events. The test set includes ~180 alpha particle events, 20 proton events, 20 thoron decay events, and 450 electron events.

Neural network training

Our experiments showed that the most important hyperparameters for training our attention U-Net model were the batch size, corresponding to the number of samples used for a single gradient step, the initialization of the network's parameters, e.g., its weights and biases, and the dropout rate,³³ which controls the percentage of random neurons in a layer that are deactivated in each iteration. To find suitable values for these hyperparameters, we performed a grid search by training models for the batch sizes $b \in \{2, 4, 8\}$, for the initializations $i \in \{\text{He Normal},^{30} \text{unsupervised pre-training}^{34,35}\}$, and for the dropout rates $d \in \{0.1, 0.2\}$.

Since neural network performs object detection and classification simultaneously, we decided to use the sum of the dice loss and the weighted cross-entropy as a loss function.³⁶ The dice loss penalizes the network when its predicted mask has a small overlap with the target mask, and the weighted cross-entropy loss penalizes the network when its predicted class probabilities differ from the target class with respect to the frequency of the class. So taking the unweighted sum of both functions yields a loss function that encourages the network to detect the particles accurately and classify them correctly.

In order to apply unsupervised pre-training, we trained a version of the network without skip connections to output an image that is identical to the input image. The removal of the skip connections prevents the modified network from just outputting the transferred input image from the first skip connection. As a loss function, we used the mean squared error loss. The weights of the trained modified network were then utilized as initialization for the attention U-Net, where the weights of the encoder part were frozen to ensure that they would not be destroyed by the large gradients of the early training process. The decoder weights were not frozen since semantically segmenting the input image drastically differs from reconstructing the original image, whereas encoding the input image should be similar for both tasks.

For each combination of hyperparameters, an attention U-Net was trained using eight-fold cross-validation, meaning that we divided the training set into eight folds, used one as a validation set, combined the remaining folds to a training set, and repeated the training until every fold was used as a validation set. Among all iterations, the model achieving the lowest loss on the validation set was then chosen as the best-performing model of the current hyperparameter configuration. Finally, the model of the hyperparameter configuration that achieved the lowest loss on the validation set was selected as our final model. The resulting model was trained using the hyperparameter values $b = 4$, $i = \text{He Normal}$, and $d = 0.1$.

Hardware

The neural networks were trained on a server equipped with two NVIDIA Tesla V100 GPUs with 16 gigabytes of VRAM. The training took approximately five days to complete.

Evaluation of the prediction time was done using an Intel Core i7-10750H 6-Core Laptop CPU with a base speed of 2.6 GHz.

Explicit modeling

The goal of post-processing is to reduce the number of particle tracks annotated as the wrong class. The neural network's predictions are most robust with respect to the electron tracks, which is likely due to their high presence in the training set and their distinct characteristics. Thus, post-processing is only concerned with the remaining classes. Our model's two most common classification errors are the incorrect classification of V tracks and a tendency to annotate alpha particle tracks as proton tracks. The first problem is likely due to the rarity of double alpha particle emissions, which results in the number of training examples being too small, despite weighting each class' impact on the loss according to its frequency in the training set. The second problem is likely due to proton trails often having gaps, making the single parts of the tracks look like multiple alpha particle tracks. In order to fix these misclassifications, we complement the model with post-processing techniques that aim to classify V tracks and proton tracks based on their shape and then check whether the remaining tracks are alpha particle tracks or background noise (Fig. 5).

To differentiate the types of tracks, we need to check the values of certain predicted features. To achieve this, we first define a set of visual features of the particle tracks and measure their values for every sample in our training set. We then fit distributions to the empirical measurements of each feature and compute their 95%-confidence intervals. We utilize this in the post-processing by passing predicted features to statistical tests, meaning that we test whether the value of the predicted feature is within the 95%-confidence interval of the corresponding distribution. If a test is failed, we know that the currently processed track contains pixels that are likely to be incorrectly classified. We resolve the issues of incorrectly classified pixels by assigning them to the class with the second-highest probability so that the following post-processing steps can check whether the new classification is correct.

Given the neural network's prediction, the post-processing starts with the detection of V tracks consisting of six steps: (i) size filtering [Fig. 4(a)], (ii) skeletonizing [Fig. 4(b)], (iii) gap removal [Fig. 4(c)], (iv) rotation [Fig. 4(d)], (v) V fitting [Fig. 4(e)], and (vi) final annotation [Fig. 4(f)].

Step (i) begins by treating every particle track containing pixels that have been labeled as V as potential V tracks. Given those tracks, we first compute the total number of pixels belonging to every track and pass it to a statistical test. The potential V tracks that passed step (i) go through the remaining steps of V detection. Step (ii) skeletonizes³⁷ the tracks, which means that we iteratively remove random pixels from the borders of the track until we have a one-pixel wide line that resembles the structure of the original track. Some double alpha particle emissions leave tracks that have a small gap between the tracks of the emitted alpha particles. This leads to step (ii) computing separate skeleton lines for a single track. We

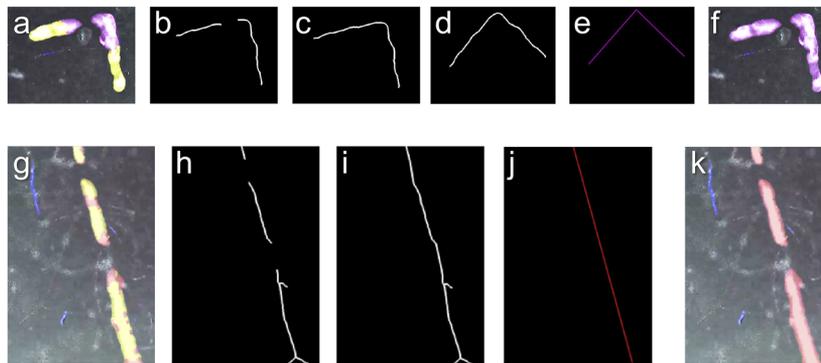


FIG. 4. The steps of the V and proton detection. (a)–(f) Individual steps of the V detection. The potential V track is skeletonized and gaps are closed. Afterward, it is rotated so that a V shape can be fitted to it, and if the resulting features pass statistical tests the track is annotated as V. (g)–(k) Individual steps of the proton detection. Identical to V detection with the difference that we fit a straight line to the skeleton line and do not rotate it.

resolve this issue in step (iii) by drawing a connection line between two skeleton lines when the minimum distance between their pixels passes statistical testing. After all potential gaps have been closed, we can determine the shape of the resulting connected skeleton line. V tracks can appear in any orientation, but we need them to be vertically oriented to check if they resemble a V shape. Therefore in step (iv), we compute versions of the skeletonized track from step (iii) rotated in steps of 12° . This guarantees that if the track is a V track, at least one version is in a vertical orientation, thus allowing us to fit a V shape to it. Next, we start to apply step (v) to the versions of the skeletonized track, beginning with standardizing the coordinates of the skeleton line's pixels. We then compute a V shape with spline regression: We define a knot at the leftmost part of the skeleton line, at the middle, and at the rightmost part of the skeleton line. The polynomials between these knots are restricted to have a degree of 1. This will compute a piece-wise defined polynomial that matches the line's pixels as much as possible while also resembling the shape of the letter V (possibly upside down). On this fit, the mean absolute error is computed. Afterward, we reverse the standardizing of the V shape coordinates to calculate the arms' lengths and the angle between the arms. The three quantities are then statistically tested. If the current version of the skeleton line passes the tests, we return that the corresponding track passed the V detection, and if not, we continue with the further rotated version of the line. If a particle track passes the V track detection, its pixels are classified as V, and otherwise, we reassign the incorrectly classified pixels.

Now that every track containing pixels labeled as V is likely to be correctly classified, the post-processing continues by checking whether tracks containing pixels annotated as protons correspond to proton tracks or alpha particle tracks. Similar to the V detection, proton detection consists of five steps: (i) size filtering [Fig. 4(g)], (ii) skeletonizing [Fig. 4(h)], (iii) gap removal [Figs. 4(i) and 4(j)], (iv) length filtering, and (v) final annotation [Fig. 4(k)]. Note that steps (i), (ii), and (v) are identical to steps (i), (ii), and (vi) of V detection. Proton and alpha particle tracks can be distinguished by their length. However, proton tracks usually have gaps, so in order

to compute the length of the whole track, we first need to close the gaps between its parts. Given the skeleton lines of potential proton tracks from step (ii), step (iii) closes the gaps between skeleton lines in the same way as step (iii) in the V detection. To ensure that we only connect skeleton lines that belong to the same track, we only close a gap when the resulting skeleton line is straight. We approximate the straightness of a skeleton line by computing a linear regression fit on the standardized line and passing the mean squared error of the fit to a statistical test. After closing all potential gaps in the skeleton lines, step (iv) approximates their length by computing the number of pixels in every skeleton line and statistically testing it. The corresponding tracks are then annotated accordingly in step (v).

At this point, all detected particle tracks only contain pixels that are assigned to the same class. Tracks annotated as alpha particle tracks consist of pixels that either have been classified as alpha particle by the neural network or by the proton or V detection. The alpha filtering checks whether these tracks are alpha particle tracks or merely background noise that has been incorrectly detected as a particle track. We achieve this by applying the size filtering of step (i) of the V and proton detection with the distinction that filtered pixels are directly classified as background noise.

RESULTS

Neural approach

We evaluate the particle track detection by calculating the mean dice coefficient³⁶ between the neural model's predictions on the test set and the corresponding targets. The dice coefficient measures how much the model's segmentation matches the manual segmentation of the cloud chamber image.

The results show that the attention U-Net effectively segments the particle tracks from the background [Fig. 6(d)]. The precision of this segmentation increases when we look at the alpha particle, V, and proton tracks separately. The lower performance on elec-

tron tracks is likely because thin tracks are significantly harder to differentiate from background noise than the other tracks.

To evaluate how well the attention U-Net classifies the detected particle tracks, we compare the target classification of every particle track in the test set with its predicted classification, given that the neural network detected the track in the first place. Since neural network sometimes classifies the pixels of a single particle track differently, we use the class with the most pixels inside the track as the predicted class of the whole track.

The experiment shows that the model correctly classifies 96% and 99% of the detected proton and electron tracks [Fig. 6(d) “neural”]. The higher misclassification rate of alpha particle and V tracks is due to the model confusing the two types of tracks: The first confusion matrix shows that 10% of alpha tracks are classified as V tracks, and 14% of V tracks are classified as alpha tracks [Fig. 6(a)]. A possible explanation for this misclassification is that V tracks are difficult to distinguish from two close alpha particle tracks. Furthermore, with a low relative abundance of 3.3%, V tracks are underrepresented in the training set [Fig. 6(e)].

Neuro-explicit approach

In order to reduce the misclassification rates of alpha particle and V tracks, we utilize our post-processing algorithms:

The neural network’s predicted mask is first given to the V detection, which determines whether tracks classified as V display the corresponding characteristics. Afterward, the proton detection analyzes the remaining tracks that have been classified as proton similarly. In the last step, predicted alpha tracks are annotated as background if their area is too small to be an actual alpha track (Fig. 5).

The results indicate that the neuro-explicit model achieves better classification performance than the neural model, reducing the total misclassification rate from 6% to 5% [Fig. 6(d) “neuro-explicit

1”]. We observe a significant drop in the misclassification rate of V tracks from 15% to 4%. There is an increase in the misclassification rates of proton tracks from 4% to 7%, which can be contributed to a 3% increase in proton tracks being classified as alpha tracks [Fig. 6(b)]. This suggests that the algorithm concerned with checking the classifications of the proton tracks fails at distinguishing proton and alpha tracks. Therefore, we test a second neuro-explicit model that only uses the V detection and the alpha filtering. The results of the second neuro-explicit model confirm our presumption since we can observe a decrease in the misclassification rate of proton tracks by 4% [Fig. 6(d) “neuro-explicit 2”]. This model reduces the total misclassification rate to 4%. The mean prediction time of the neuro-explicit model is within 1 s, approximately twice that of the neural model.

To validate that combining neural classification with explicit modeling classification is advantageous, we evaluate a model that classifies the tracks solely using explicit rules. This model first removes the electrons from the neural network’s prediction since post-processing only classifies alpha particle, proton, and V tracks. The remaining particle tracks are then given to the V detection, where the tracks that do not pass are sent to the proton detection. If these tracks do not pass the proton detection, they are sent to the alpha filtering, where they are classified as either alpha particle tracks or background.

We can observe that with respect to V tracks, the misclassification rate of the explicit model is only 3% higher than the rates of the neuro-explicit models, which further verifies the accuracy of the V detection since now every detected track is passed to it [Fig. 6(d) “explicit”]. However, the misclassification rate of alpha particle and proton tracks drastically increases to 48% and 14%. This confirms our previous result that the proton detection is less reliable than the neural network’s predictions since this increased error can be explained by significantly more tracks being passed to the proton detection.

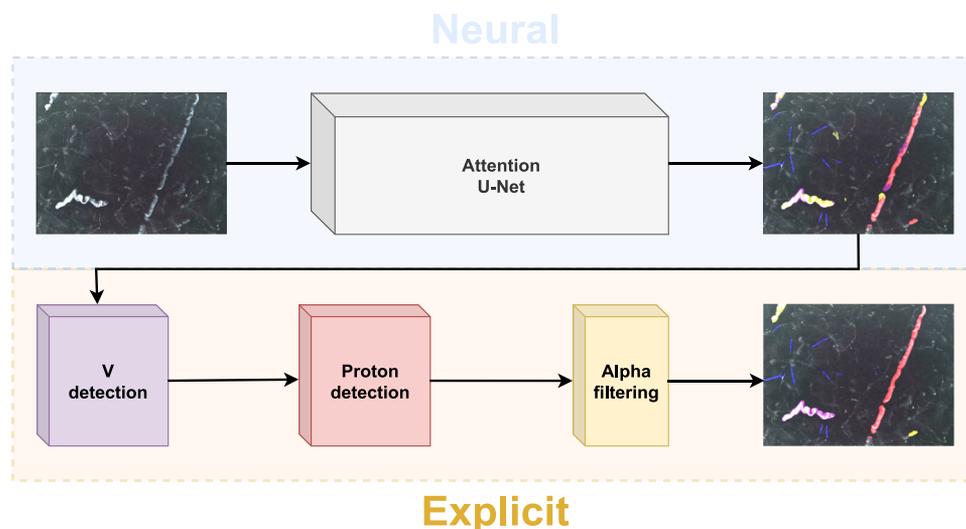


FIG. 5. Schematic of the neuro-explicit model.

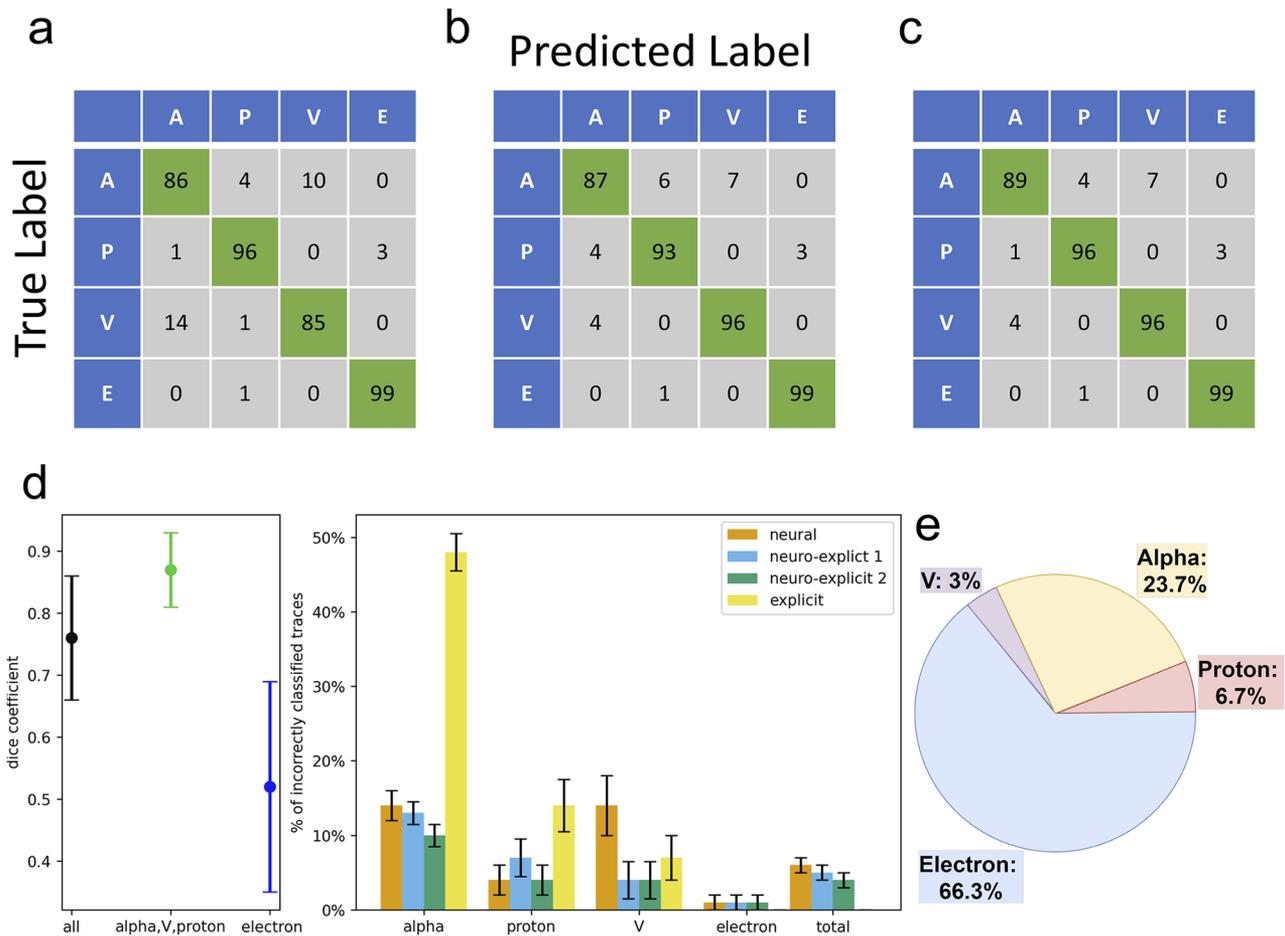


FIG. 6. Results and distribution of the training set. (a)–(c) Confusion matrices showing the percentages of correctly and incorrectly classified tracks of the neural, neuro-explicit 1, and neuro-explicit 2 models. (d) Plots showing the detection and classification accuracy for the different tracks. The classification accuracy is compared between the neural, explicit, and neuro-explicit models and the error bars correspond to the 95% confidence interval. (e) Distribution of particle tracks in our training set.

DISCUSSION

In order to automatically detect and classify particle tracks seen in a diffusion cloud chamber, we developed a model that reliably segments alpha particle, V, proton, and electron tracks from the background. We achieve this by complementing the attention U-Net neural network architecture with explicit modeling methods. By extending the neural model to a neuro-explicit model, we reduced the misclassification rate of the rare V tracks by 73%. Furthermore, we showed that incorporating neural network predictions and explicit modeling methods in classifying particle tracks can outperform solely relying on the latter.

A possible step for future work would be to increase the amount of training data, which could decrease the model's misclassification rate. The increase in training data might also enable accurately annotating the video stream from the cloud chambers' camera in real-time using, e.g., video segmentation.^{38,39} Furthermore, the model may be extended to detect rare particles like pions and kaons. The extension could be achieved by introducing known radiation

sources and applying unsupervised learning methods like anomaly detection⁴⁰ to longer recordings of the cloud chamber. Finally, we recognize the importance of reevaluating certain parameters, specifically confidence intervals, which were manually adjusted for the classifications of the explicit models. This reevaluation becomes crucial when encountering exceptionally different particle distributions, whether induced in various experimental setups or when the model is extended to account for new particles. Adapting these parameters will ensure the model's robustness and effectiveness in different scenarios.

SUPPLEMENTARY MATERIAL

A figure showing examples of annotations generated by our neuro-explicit 2 model can be found in the supplementary material. Additionally, we provide a video showcasing the model being applied to a video stream of our cloud chamber.

ACKNOWLEDGMENTS

The compute infrastructure for this project was funded by the DFG (Grant No. 469073465). Funding for open access charge was provided by Saarland University. The authors are grateful for the support by the Luxembourg Science Center.

We thank Suthep Pomjaksilp and Julien Meyer for insightful discussions and feedback on our work.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Nicola J. Müller: Data curation (lead); Software (lead); Writing – original draft (lead); Writing – review & editing (equal). **Daniel Porawski:** Data curation (supporting); Investigation (supporting). **Lukas Wilde:** Data curation (supporting); Investigation (supporting). **Dennis Fink:** Data curation (equal); Resources (lead). **Guillaume Trap:** Writing – original draft (equal); Writing – review & editing (equal). **Annika Engel:** Conceptualization (supporting); Investigation (supporting); Writing – original draft (equal); Writing – review & editing (equal). **Georges P. Schmartz:** Investigation (equal); Project administration (lead); Supervision (lead); Writing – review & editing (equal).

DATA AVAILABILITY

The video footage used to create the datasets was provided by the Luxembourg Science Center and they can be found in the GitHub repository https://github.com/nicola-mueller/cloud_chamber_publication. The scripts used to train, evaluate, and execute the models can also be found in the repository.

REFERENCES

- 1 A. Langsdorf, “Continuously sensitive diffusion cloud chamber,” *Rev. Sci. Instrum.* **10**, 91–103 (1939).
- 2 H. Slätis, “On diffusion cloud chambers,” *Nucl. Instrum.* **1**, 213–226 (1957).
- 3 K. E. Relf, “Cloud-chamber study of cosmic-ray air showers at sea level,” *Phys. Rev.* **97**, 172–180 (1955).
- 4 S. N. Vernov *et al.*, “Investigation of the core of extensive air showers,” *J. Exp. Theor. Phys.* **36**, 669–681 (1959).
- 5 T. Gyorfi and P. Raics, “Diffusion cloud chamber in education,” *Int. J. At. Nucl. Phys.* **4**, 015 (2019).
- 6 M. Feickert and B. Nachman, “A living review of machine learning for particle physics,” [arXiv:2102.02770](https://arxiv.org/abs/2102.02770) [hep-ph] (2021).
- 7 P. Thomadakis, A. Angelopoulos, G. Gavalian, and N. Chrisochoides, “Using machine learning for particle track identification in the CLAS12 detector,” *Comput. Phys. Commun.* **276**, 108360 (2022).
- 8 A. Aurisano *et al.*, “A convolutional neural network neutrino event classifier,” *J. Instrum.* **11**, P09001 (2016).
- 9 C. F. Madrazo, I. Heredia, L. Lloret, and J. Marco de Lucas, “Application of a convolutional neural network for image classification for the analysis of collisions in high energy physics,” *EPJ Web Conf.* **214**, 06017 (2019).
- 10 D. Heck, J. Knapp, J. N. Capdevielle, G. Schatz, and T. Thouw, “CORSIKA: A Monte Carlo code to simulate extensive air showers,” Report No. FZKA 6019 (Forschungszentrum Karlsruhe, 2018).
- 11 R. L. Workman *et al.*, “Review of particle physics,” *Prog. Theor. Exp. Phys.* **2022**, 083C01.
- 12 G. Barzon, “Advanced automatic analysis of cloud chamber images,” B.Sc. (2018).
- 13 O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” *Med. Image Comput. Comput. Assist. Interv.* **18**, 234–241 (2015).
- 14 Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, “A review of semantic segmentation using deep neural networks,” *Int. J. Multimedia Inf. Retr.* **7**, 87–93 (2018).
- 15 O. Oktay *et al.*, “Attention gated networks: Learning to leverage salient regions in medical images,” *Med. Image Anal.* **53**, 197–207 (2019).
- 16 T. Falk *et al.*, “U-net: Deep learning for cell counting, detection, and morphometry,” *Nat. Methods* **16**, 67–70 (2019).
- 17 H. Dong, G. Yang, F. Liu, Y. Mo, and Y. Guo, “Automatic brain tumor detection and segmentation using U-Net based fully convolutional networks,” in *Medical Image Understanding and Analysis*, edited by M. Valdés Hernández and V. González-Castro (Springer International Publishing, Cham, 2017), pp. 506–517, ISBN: 978-3-319-60964-5.
- 18 Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: Learning dense volumetric segmentation from sparse annotation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, edited by S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal and W. Wells (Springer International Publishing, Cham, 2016), pp. 424–432, ISBN: 978-3-319-46723-8.
- 19 Z. Zhang, Q. Liu, and Y. Wang, “Road extraction by deep residual U-Net,” *IEEE Geosci. Remote Sens. Lett.* **15**, 749–753 (2018).
- 20 Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: A nested U-Net architecture for medical image segmentation,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, edited by D. Stoyanov *et al.* (Springer International Publishing, Cham, 2018), pp. 3–11, ISBN: 978-3-030-00889-5.
- 21 M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, and V. K. Asari, “Recurrent residual convolutional neural network based on U-Net (R2U-Net) for medical image segmentation,” *J. Med. Imag.* **6**, 014006 (2019).
- 22 S. Koh *et al.*, “A probabilistic U-Net for segmentation of ambiguous images,” in *Advances in Neural Information Processing Systems*, edited by S. Bengio *et al.* (Curran Associates, Inc., 2018), p. 31.
- 23 N. Ibtehaz and M. S. Rahman, “MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation,” *Neural Networks* **121**, 74–87 (2020).
- 24 N. O’Mahony *et al.*, “Deep learning vs traditional computer vision,” in *Advances in Computer Vision*, edited by K. Arai and S. Kapoor (Springer International Publishing, Cham, 2020), pp. 128–144, ISBN: 978-3-030-17795-9.
- 25 M. Omran, C. Lassner, G. Pons-Moll, P. V. Gehler, and B. Schiele, “Neural body fitting: Unifying deep learning and model-based human pose and shape estimation,” in *2018 International Conference on 3D Vision (3DV)* (IEEE Computer Society, 2018), pp. 484–494.
- 26 J. Dong, S. Roth, and B. Schiele, “Deep Wiener deconvolution: Wiener meets deep learning for image deblurring,” *Adv. Neural Inf. Process. Syst.* **33**, 1048–1059 (2020).
- 27 M. Böhle, M. Fritz, and B. Schiele, “Convolutional dynamic alignment networks for interpretable classifications,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE Computer Society, 2021), pp. 10029–10038.
- 28 K. E. Relf and W. M. Powell, “Diffusion cloud chamber of unusually large dimensions,” *Rev. Sci. Instrum.* **25**, 1182–1187 (1954).
- 29 N. Kalchbrenner and P. Blunsom, “Recurrent continuous translation models,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Seattle, WA, USA, 2013), pp. 1700–1709.
- 30 K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2016), pp. 770–778.
- 31 D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *Conference Track Proceedings, 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, CA, USA, May 7–9, 2015.

- ³²C. Shorten and T. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *J. Big Data* **6**, 60 (2019).
- ³³N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
- ³⁴A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. (O’Reilly Media, Inc., 2019).
- ³⁵S. Wiehman, S. Kroon, and H. De Villiers, “Unsupervised pre-training for fully convolutional neural networks,” in *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)* (IEEE, 2016), Vol. 1–6.
- ³⁶C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support* (Springer International Publishing, 2017), pp. 240–248.
- ³⁷T. Y. Zhang and C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” *Commun. ACM* **27**, 236–239 (1984).
- ³⁸S. Valipour, M. Siam, M. Jagersand, and N. Ray, “Recurrent fully convolutional networks for video segmentation,” in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (IEEE, 2017), pp. 29–36.
- ³⁹A. Pfeuffer and K. Dietmayer, “Separable convolutional LSTMs for faster video segmentation,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (IEEE, 2019), pp. 1072–1078.
- ⁴⁰T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” in *Information Processing in Medical Imaging: 25th International Conference*, edited by M. Niethammer *et al.* (Springer International Publishing, Cham, 2017), pp. 146–157, ISBN: 978-3-319-59050-9.