Article

Automatic Optimization of a Parallel-Plate Avalanche Counter with Optical Readout

María Pereira Martínez, Xabier Cid Vidal and Pietro Vischia

Article

Automatic Optimization of a Parallel-Plate Avalanche Counter with Optical Readout

María Pereira Martínez [1,*,†], Xabier Cid Vidal [1,*,†] and Pietro Vischia [2,3,*,†]

1 Instituto Galego de Física de Altas Enerxías (IGFAE), Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain
2 Departamento de Física, Universidad de Oviedo, 33007 Oviedo, Spain
3 Instituto de Ciencias y Tecnologías Espaciales de Asturias (ICTEA), 33007 Oviedo, Spain
* Correspondence: maria.pereira.martinez@cern.ch (M.P.M.); xabier.cid.vidal@cern.ch (X.C.V.); pietro.vischia@cern.ch (P.V.)
† MODE Collaboration.

Abstract: An automatic optimization procedure is proposed for some operational parameters of a Parallel-Plate Avalanche Counter with Optical Readout, a detector designed for heavy-ion tracking and imaging. Exploiting differentiable programming and automatic differentiation, we model the reconstruction of the position of impinging 5.5 MeV alpha particles for different detector configurations and build an optimization cycle that minimizes an objective function. We analyze the performance improvement using this method, exploring the potential of these techniques in detector design.

Keywords: differentiable programming; machine learning; detector optimization; nuclear physics

1. Introduction

Over the last decades, the availability of high-performance computing and the development of deep learning [1] have transformed the optimization of complex systems. When the dimensionality of the space of relevant design parameters exceeds a few units or the relationships between different parameters are not trivial, automated processes can be developed to identify configurations corresponding to the minimum of a carefully specified objective function.

The core of these optimization searches is differentiable programming (DP), a paradigm in which computer programs can be differentiated end-to-end automatically, enabling gradient-based optimization of parameters within the program leveraging automatic differentiation (AD) [2–5].

Despite the long-established use of AD in various fields, such as circuit design, aerodynamic design, and engineering in general [6–8], its application to particle and nuclear physics detectors remains challenging. This difficulty primarily arises from the intrinsic stochasticity introduced by the quantum nature of the physical processes involved. As a result, building differentiable pipelines becomes complex, since most Monte Carlo (MC) particle simulators, such as GEANT4 [9–11], are not inherently differentiable. Although efforts have been made to overcome this limitation for specific physical processes [12], the full optimization of particle detectors remains an open challenge. To date, only a handful of studies have addressed this problem [12–17].

This approach is becoming feasible due to the ongoing efforts of collaborative research such as MODE Collaboration [12], which aims to utilize DP and AD for optimizing detector designs in particle physics by developing modular and customizable differentiable

pipelines for the optimization of objective functions. The recent success of MODE Collaboration in applying these concepts to fully model a muon tomography system and develop a package called TomOpt [18] led to this study, where this package is adapted to a gaseous detector designed for heavy-ion tracking and imaging: the Optical Parallel-Plate Avalanche Counter (O-PPAC).

The basic design of the O-PPAC, introduced by Cortesi et al. in Reference [19], consists of two parallel squared electrodes separated by a small 3 mm gap filled with a low-pressure scintillating gas, with an array of small, collimated photo-sensors along the edges of the gap. The position of an impinging particle is reconstructed using the information provided by the distributions of detected photons along the edges of the gas gap.

The main goal of this study is to identify the optimal values of two detector parameters: the pressure of the scintillating gas (p) and the length of the collimator (L), both of which affect the spatial resolution. To achieve this, a differentiable pipeline is developed to minimize the reconstruction error as a function of the detector parameters. This is performed by developing a surrogate model that replicates an existing GEANT4 simulation of this detector fully described in Reference [19], where 5.5 MeV alpha particles traverse the detector perpendicularly to the electrodes.

The surrogate acts as a differentiable approximation of the detector's behavior, enabling a gradient-based optimization using AD. In this study, PyTorch's [20] AD features are employed.

The structure of the document is as follows: Section 2 briefly introduces the operational principle of the detector and provides a detailed explanation of the optimization methods applied. Section 3 presents the results obtained from the optimization process. Lastly, Section 4 discusses the conclusions drawn from this work.

2. Materials and Methods

2.1. The O-PPAC Detector

Despite being an older detector concept, position-sensitive parallel-plate avalanche counters (PPACs) remain widely used today in various subatomic physics applications, particularly for heavy-ion position and timing measurements [21].

As discussed in Reference [19], unlike conventional PPACs that use charge division or delay-line methods, electroluminescence-based detectors benefit from high-sensitivity solid-state photosensors, such as SiPMs, which provide better signal-to-noise ratios and energy resolution, making them well suited for heavy-ion tracking and imaging.

One example of this approach is the O-PPAC design [19], which consists of two parallel electrodes separated by a narrow 3 mm gap. This gap is filled with a low-pressure scintillating gas, such as CF4, known for its high electroluminescence light yield. Along the edges of the avalanche gap, arrays of small, collimated SiPMs are strategically arranged to maximize light collection and improve detection efficiency.

When an ionizing particle crosses the active volume, it releases a small amount of energy in the form of ionization electrons, which are multiplied in the gas by a uniform electric field established between the two metalized parallel plates. The scintillation light emitted during the avalanche process, known as secondary scintillation or electroluminescence, is reflected back and forth by the two metallized electrode foils and guided to be recorded by the arrays of collimated photo-sensors (see Figure 1). The collimation is crucial for the precise localization of the impinging particles, as it narrows down the detected photon distributions, so its peak is more heavily weighted near the position of the avalanche.

In this study, the parameters of interest were the pressure of the scintillating gas and the length of the collimator, as they are crucial parameters for the characteristics of the detected photon distributions.
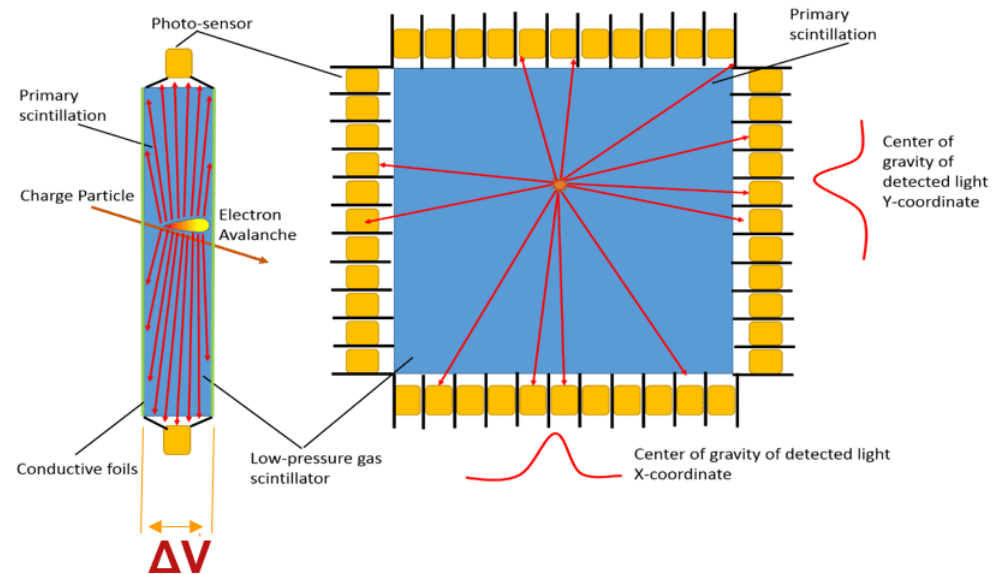
Figure 1. Schematic representation of the operational principle of the Optical Parallel-Plate Avalanche Counter (O-PPAC). The photosensors are highlighted in yellow, the collimator in black, and the gas volume in blue. The left side shows a lateral view of the O-PPAC, illustrating the avalanche process initiated when a particle traverses the active medium, with the scintillating photons and their trajectories highlighted in red. On the right, the front view of the detector is presented, demonstrating the effect of collimation. Image adapted from Reference [19].

The pressure of the gas is directly related to the electroluminescence yield, as explained in Reference [22]. As a general rule, higher pressure implies a higher electroluminescence yield, which translates into a higher number of detected photons in each event and, therefore, higher statistics in the detected photon distributions.

As the width of the collimator walls (1 mm) is determined by the manufacturing constraints of the SiPMs selected by the detector designers [19] and alternative technologies were not explored in this study, the optimization efforts were focused on the other relevant parameter regarding collimation: the collimator length. The optimal collimator length results from a trade-off between achieving more accurate light spot localization with larger L and obtaining higher statistics with smaller L.

It is also worth mentioning that the number of SiPMs was fixed to 33 per wall, the SiPM effective area was 2×3 mm^2, the pitch was 3 mm long, and the cathode and anode metalized foils (Al) were characterized by a reflectivity of 90%. The remaining simulation parameters not mentioned in this study are exactly as detailed in Reference [19], with the exception of the collimator length and pressure, which are studied in this work.

2.2. End-to-End Optimization

As mentioned earlier, applying gradient-based optimization techniques requires the detector response to be in a differentiable form. However, GEANT4 simulations remain non-differentiable, which makes it challenging to directly apply gradient-based methods.

To overcome this, approximate surrogate models can be used as an alternative to Monte Carlo simulations [23]. These models, which are typically obtained by employing some form of supervised training on events previously generated by the MC simulator, not only enable differentiation but also offer practical benefits: once trained, they run significantly faster than a gradient-aware MC simulation and often provide a smoother approximation, which is better suited for gradient-based optimization [24,25].

In this study, a surrogate model is obtained by training a Neural Network (NN) on a grid of MC simulated points of the parameter space. After the training step, a differentiable

model that can be inserted into a differentiable pipeline just as a closed-form expression is obtained, which is the key to this approach. Specifically, once trained, this NN will predict the reconstructed position of the particle as a function of the detector parameters and its initial position.

Once the model is trained, the next step is to build an optimization loop that will minimize an objective function by iteratively updating the values of the detector parameters according to the gradients of this function at each step.

In the following subsections, each step involved in the optimization of the detector is described in detail.

2.2.1. Simulation

The first step to optimize the detector is to generate a set of datasets corresponding to a grid of the interest parameters and the beam position. As stated before, our goal is to simulate a reduced number of datasets and then train an NN model that extrapolates all the possible configurations with a differentiable model. For this, the GEANT4 simulation described in Reference [19] is employed.

The grid of parameters employed for the simulation is provided in Table 1. It is important to highlight that each dataset consisted of 10,000 events, each involving a 5.5 MeV alpha particle entering the detector perpendicular to the parallel plates.

Table 1. Summary of the simulated parameter values. The simulation explored all possible combinations of the listed parameters, resulting in a total of 2025 unique configurations.

Parameter	Simulated Values
Pressure (Torr)	10, 20, 30, 40, 50
Collimator length (mm)	5, 16.25, 27.5, 38.75, 50.
X position of the beam (cm)	−4, −3, −2, −1, 0, 1, 2, 3, 4
Y position of the beam (cm)	−4, −3, −2, −1, 0, 1, 2, 3, 4

As can be observed, the pressure and the collimator length were bound to lie within a certain interval. The minimum pressure value is justified by the photon distribution statistics, as trial simulations showed that below 10 Torr, few to no photons were detected. The upper limit of 50 Torr is determined by the fact that higher pressure requires an increase in the voltage between the parallel plates, which cannot exceed a certain maximum.

In a similar manner, the lower limit of the collimator length is set taking into account the poor precision obtained, as the dispersion of the distribution when $L < 5$ mm is too high. The upper limit is set due to the poor statistics obtained when $L > 50$ mm.

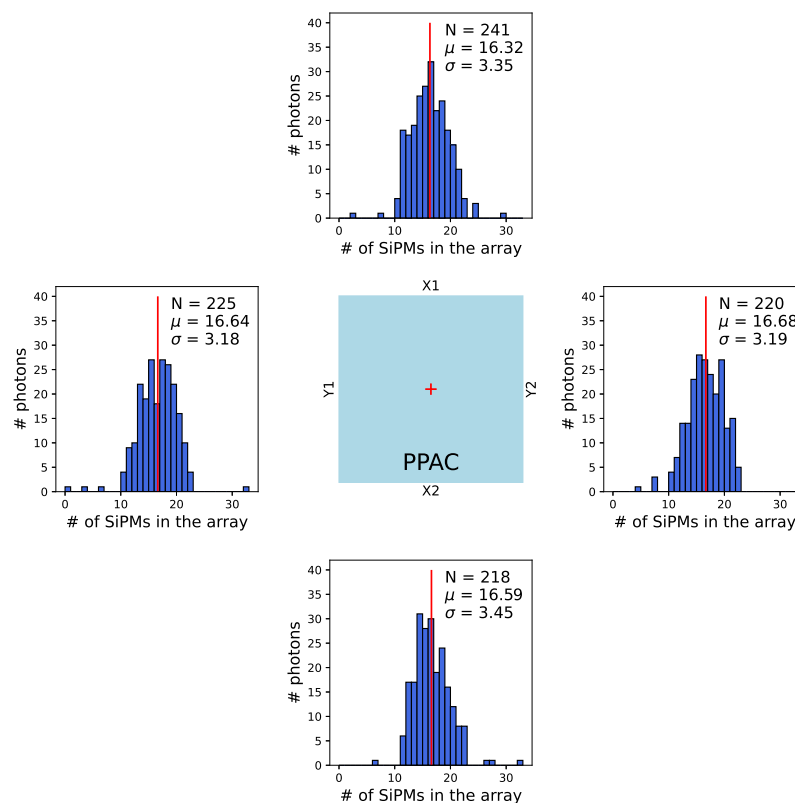### 2.2.2. Reconstruction of the Position

The reconstruction of the avalanche location, which corresponds to the position of the charged-particle crossing the detector volume $(\hat{x}, \hat{y})$, is achieved by combining the data recorded by the four photo-sensor arrays located in each wall of the PPAC, as illustrated in Figure 2.

This task can be performed in several ways, but the simplest is to compute the arithmetic mean between the photon distribution peaks recorded by each pair of opposing arrays weighted by the total number of detected photons and the dispersion of each distribution [19]:

$$\hat{x} = \left(\frac{P_{x1} \cdot N_{x1}}{\sigma_{x1}} + \frac{P_{x2} \cdot N_{x2}}{\sigma_{x2}} \right) \Big/ \left(\frac{N_{x1}}{\sigma_{x1}} + \frac{N_{x2}}{\sigma_{x2}} \right) \tag{1}$$

$$\hat{y} = \left(\frac{P_{y1} \cdot N_{y1}}{\sigma_{y1}} + \frac{P_{y2} \cdot N_{y2}}{\sigma_{y2}} \right) \Big/ \left(\frac{N_{y1}}{\sigma_{y1}} + \frac{N_{y2}}{\sigma_{y2}} \right), \tag{2}$$

where P and σ correspond to the mean and the standard deviation of the distribution on each wall and N is the total number of photons detected in each wall.

Figure 2. Illustration of the reconstruction process for a simulated event induced by a 5.5 MeV alpha particle at the center of the parallel plates. The photon counts registered by each photosensor are used to generate four distributions, one for each wall. The resulting plots provide the statistical information required for reconstruction, with the mean of each distribution highlighted in red. The particle's position along each axis is determined by analyzing the signals from opposing walls.

2.2.3. Surrogate Model

As stated before, the algorithm employed to parametrize the reconstruction is a Neural Network developed employing PyTorch [20]. This particular NN receives four inputs (x, y, p, L) and predicts two outputs ($\hat{x}$, $\hat{y}$), i.e., the reconstructed position.

In order to optimize, train, and evaluate the model, the simulated events were divided into different datasets, as illustrated in Table 2. For instance, a small fraction (around 5×10^4 events) was employed for hyperparameter tuning, a bigger dataset for training (around 1 million), and the remaining MC events were used for the evaluation of the NN.

Table 2. Number of simulated events used in the different phases of the surrogate model development process, including hyperparameter tuning, training, and evaluation.

Dataset	Training	Validation	Evaluation
Hyperparameter tuning	2.5×10^4	2.5×10^4	—
Training	5×10^5	5×10^5	—
Evaluation	—	—	2×10^7

It is worth mentioning that the splitting in the different datasets was made maintaining the proportion of the different parameters, so that all combinations are equally represented in each dataset.

To optimize the performance of the NN, a hyperparameter tuning process was conducted to identify the optimal set of model parameters. The package employed for hyperparameter tuning was Optuna [26], which utilizes the Tree-structured Parzen Estimator (TPE) as its default algorithm for sampling candidates in the search space. TPE is a widely utilized Bayesian optimization method that iteratively constructs two probability density functions: one for the hyperparameters of successful trials (i.e., favorable configurations), and another for unsuccessful trials (i.e., unfavorable configurations). The algorithm employs these probability density functions to sample new hyperparameters that are likely to improve the objective function. This method has demonstrated greater efficiency and effectiveness compared to other well-established hyperparameter tuning methods, such as Grid Search or Randomized Search.

The hyperparameter tuning process was conducted in two distinct phases: the initial phase focused on architecture optimization, while the subsequent phase aimed at further enhancing performance and regularization.

To achieve this objective, the hidden size, number of hidden layers, and learning rate are optimized in the initial phase. The considered values of these hyperparameters, along with the optimal parameters obtained from 100 trials, are detailed in Table 3.

Table 3. Results from the first hyperparameter tuning with Optuna, showing the best trial out of 100. The table lists the hyperparameters considered, their tested values, and the optimal values based on model performance.

Hyperparameter	Considered Values	Best Trial
Hidden size	32, 64, 128, 256, 512	64
Number of layers	2, 3, 4, 5, 6, 7, 8	3
Learning rate	From 0.001 to 0.1	0.0352

As stated before, the second phase of the hyperparameter tuning was focused on further improving the performance of the NN, on regularization and stability. For this purpose, hyperparameters like dropout, optimizer, learning rate scheduler, batch normalization, and activation function were studied. Again, the considered options and the Optuna's best trial out of 1000 trials is specified in Table 4.

Table 4. Results from the second hyperparameter tuning with Optuna, showing the best trial out of 1000. The table lists the hyperparameters considered, their tested values, and the optimal values based on model performance.

Hyperparameter	Considered Values	Best Trial
Dropout	0, 0.1, 0.2, 0.3, 0.4, 0.5	0
Optimizer	Adam, NAdam, Adamax	Adamax
Step Size Scheduler	1, 2, 3, 4, 5	2
Gamma Scheduler	0.9, 0.99, 0.999	0.9
Use Batch Normalization	True, False	False
Activation Function	ReLU, SELU, ELU, LeakyReLU	ELU

In Table 5, the evaluation of the models for each round of hyperparameter tuning is shown. Specifically, the root mean squared error (RMSE) between the reconstructed position and the NN prediction is evaluated for the large dataset. As shown, the evaluation yields an RMSE of 0.035 cm both for the x- and y-axes after the second phase, which is

an improvement with respect to the model resulting from the first phase of hyperparameter tuning.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} |\vec{x}_{NN}^{i} - \hat{\vec{x}}^{i}|^2} \tag{3}$$

Table 5. NN prediction error (RMSE) after each step of the hyperparameter tuning process. The table shows the root mean square error (RMSE) for both the x- and y-coordinates at each step of the tuning.

Step	RMSE($\hat{x}$, x_{NN}) [cm]	RMSE($\hat{y}$, y_{NN}) [cm]
1	0.040	0.041
2	0.035	0.035

Additionally, the learning curve for the final model is illustrated in Figure 3. As can be seen, the validation loss curve exhibits an erratic behavior during the first 20 epochs, but it stabilizes afterwards, with the training loss value closely matching the validation loss, which suggests a low level of overfitting.
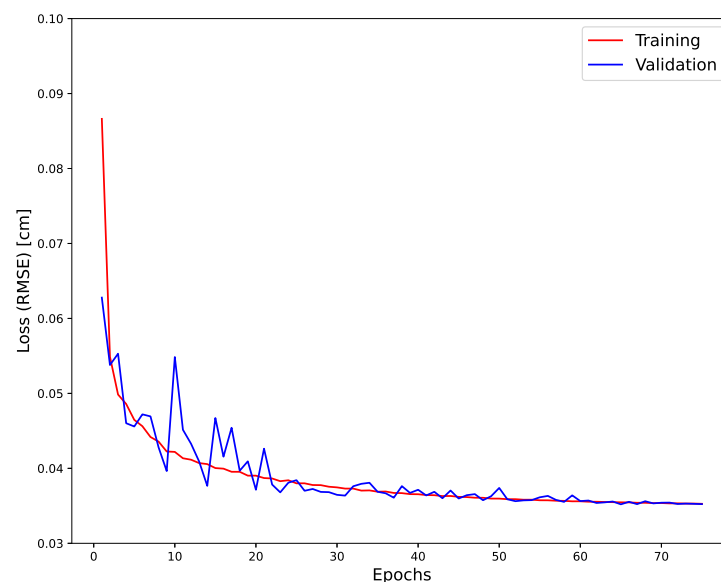
Figure 3. Learning curve of the NN surrogate model following the second phase of hyperparameter tuning. The plot illustrates the model's training progress, with early stopping applied using a tolerance (δ) of 0.001 and a patience of 10 epochs to prevent overfitting and reduce computational time.

2.2.4. Optimization

This section provides an overview of the optimization process and its implementation in the repository [27]. It describes the core components of the optimization package and offers a detailed explanation of the optimization loop. It is important to note that this code is derived from TomOpt package [28].

Below are the key components necessary for implementing the optimization, along with their descriptions:

Alpha Batch:

In order to perform the optimization loop, a class named AlphaBatch is built. This class has the main goal of producing batches of alpha beams in the form of random Pytorch tensors. The random numbers are generated uniformly both in the x- and y-axes in the interval $[-4, 4]$ cm, as the surrogate model is trained on that interval.

Volume:

The Volume refers to the detector volume itself, represented as a class defined by the detector parameters at each step in the optimization loop. It is initialized with values for pressure and collimator length, which are the parameters of interest.

The primary purpose of this class is to accept a specific detector configuration and an alpha batch, then to predict the reconstructed position for each impinging alpha particle using the surrogate model.

Moreover, the Volume class includes a method that ensures the parameter values remain within a predefined interval during the optimization loop. If a parameter exceeds the boundaries of this interval during training, its value is clamped to the nearest limit. This functionality is crucial for the optimization process, as the surrogate model is trained only within specific parameter ranges and should not be extrapolated outside these intervals.

Objective function:

The objective function is the metric minimized during the optimization process. It can be designed to prioritize parameter combinations that enhance both the detector's performance and cost-efficiency. In this study, the objective function was performance-focused, using the root mean square error (RMSE) between the reconstructed position and the beam position as the optimization criterion.

Volume Wrapper and optimization Loop:

The Volume class includes a Wrapper that contains the fit method, responsible for optimizing the detector by determining the optimal configuration based on the NN reconstruction model. This optimization is conducted within a loop designed to iteratively identify the best detector configuration. The structure of this loop is illustrated in Figure 4.
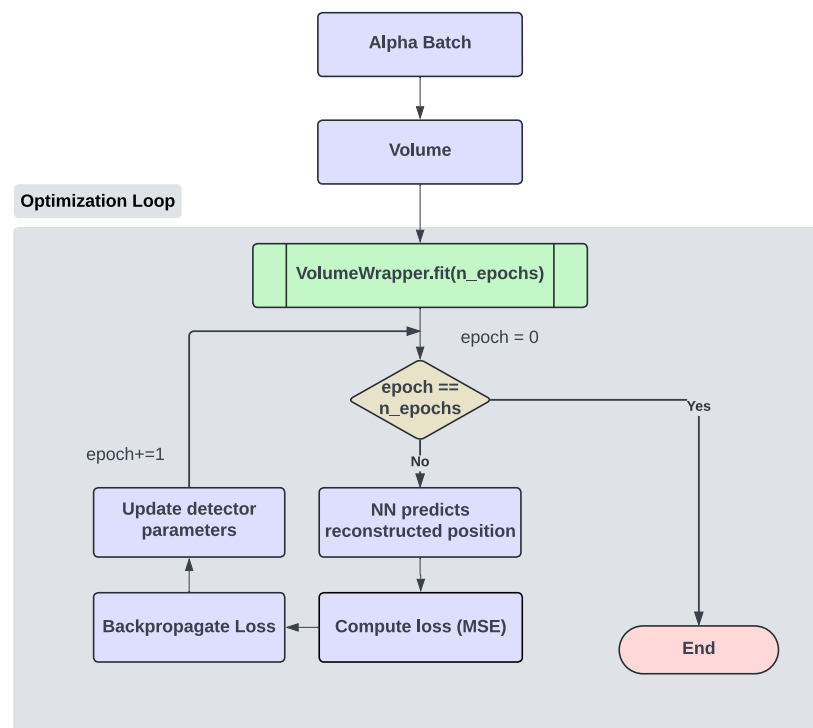
Figure 4. Breakdown of the detector optimization loop, outlining the process of initializing and updating the parameters (*p* and *L*) based on the gradient of the loss function. The loop runs through multiple epochs, where each iteration includes model prediction, loss calculation, and parameter updates to minimize the loss function.

As explained in Figure 4, the first step is to initialize an alpha batch with N alphas generated in random positions in the interval $[-4, 4]$ cm both in the x- and y-axes. Then, the volume wrapper class is initialized with the alpha batch and initial values of the pressure (p) and the collimator length (L).

Before initializing the loop, the optimizer is initialized to update the values of both parameters based on the gradient of the loss function, according to a previously specified learning rate. In this study, the Adam optimizer was employed with a learning rate of 0.1 and default values for (β_1, β_2).

At each epoch, the NN reconstruction model receives the inputs, which are the alpha batch and the initial detector configuration. The model then predicts the reconstructed position of each alpha particle.

These predicted positions, along with the real beam positions, are fed into the loss function. The next step involves backpropagating the loss to compute the gradients of this function with respect to the parameters to be optimized (p, L). Once the gradients are obtained, the optimizer takes a step, updating the detector parameters according to the computed gradients.

This process is repeated over a certain number of epochs until all values stabilize and, as a result, the detector configuration that minimizes the loss function is obtained.

3. Results
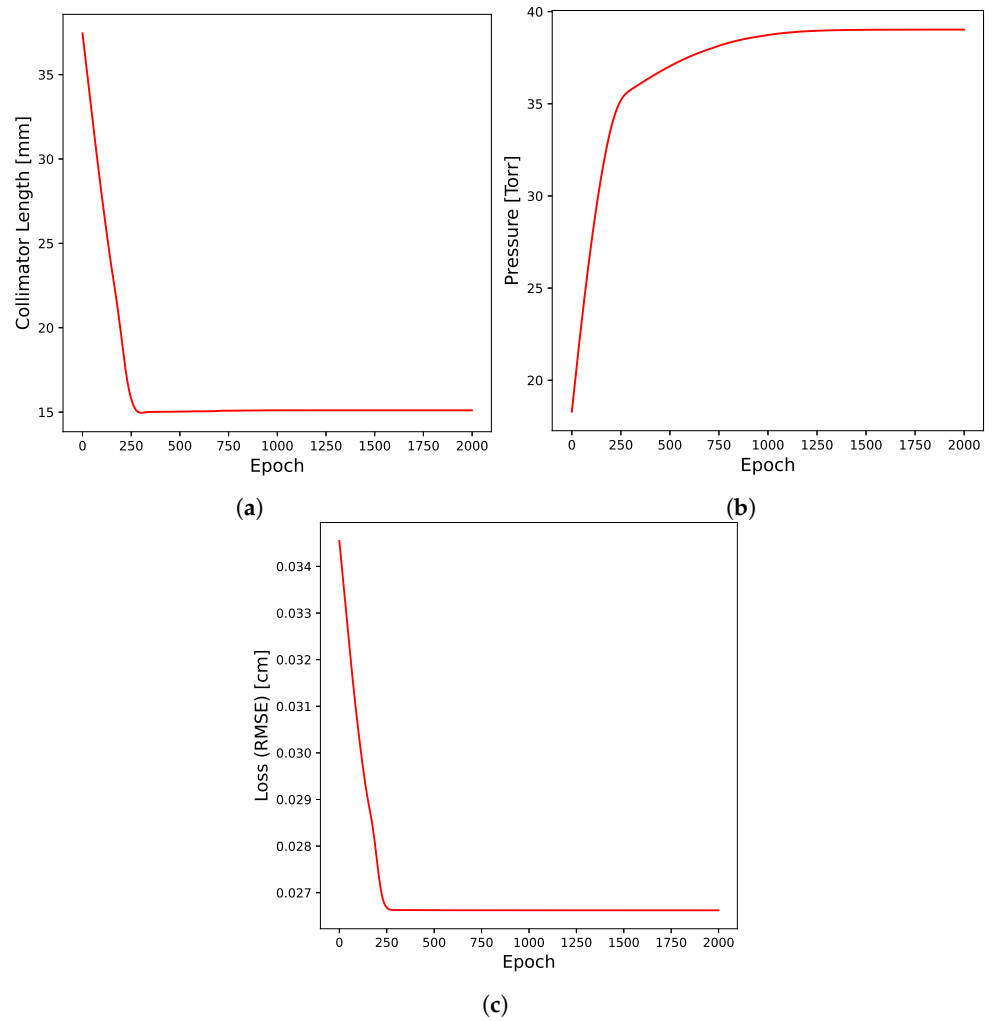
3.1. Optimization for a Random Initial Configuration

First, a single optimization loop was carried out with an alpha batch of 10,000 alphas located at random positions and a random initial configuration of the detector during 1500 epochs. In Figure 5, the evolution of the loss function, as well as the evolution of the parameters throughout the optimization loop is presented.

The optimal values of the pressure and the collimator length found with this method are presented in Table 6.

Table 6. Optimal values of pressure (p) and collimator length (L) found for a random initial configuration and an Alpha Batch of 10^5 particles, randomly distributed following a uniform distribution in the interval $[-4, 4]$ cm.

Parameter	Found Optimal Value
Pressure	39.03 Torr
Collimator length	15.11 mm

The result for the collimator length agrees with a previous result obtained in Reference [19], where traditional optimization techniques were employed, involving simulations of the detector under different parameter combinations, assessing their performance, and selecting the optimal one. In Figure 6, the spatial resolution as a function of the collimator length for different number of SiPMs per wall is illustrated. The minimum of the curve for 33 SiPMs, which is the value used in this study, corresponds to a collimator length of approximately 15 mm. This is very close to the value obtained through automatic optimization for a similar pressure (30 Torr).

(**a**)

(**b**)

(**c**)

Figure 5. Evolution of the (**a**) collimator length, (**b**) pressure, and (**c**) loss function throughout the optimization loop for a random initial configuration of pressure and collimator length. The figures show the progression of these parameters as the optimization loop iterates, with the values adjusting to minimize the loss function and improve the detector configuration.
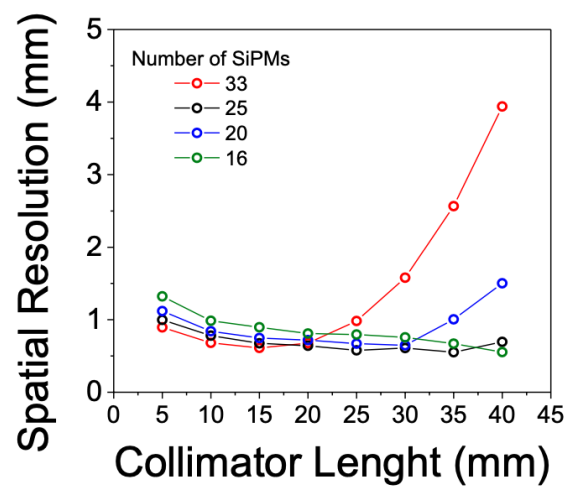

Figure 6. Position resolution as function of the collimator length, for SiPMs arrays of different granularity—the number of photo-sensor per array ranges from 12 to 33 elements. The pressure was fixed to 30 Torr. Image from Reference [19].

3.2. Optimization for a Detector Configuration Grid

To verify whether the previous result is independent of the initial configuration of the parameters—which would indicate that the identified minimum is the absolute minimum of the objective function—the optimization loop was repeated across a grid of initial detector configurations. The values considered for both parameters are detailed in Table 7.

This grid translates into 400 different configurations and, therefore, 400 optimization loops that are carried out just as described in Section 3.1.

As shown in Figure 7, all configurations converge to the same optimal values for pressure and collimator length, indicating that this minimum of the loss function is the absolute minimum within the studied range.

(**a**)

(**b**)

(**c**)
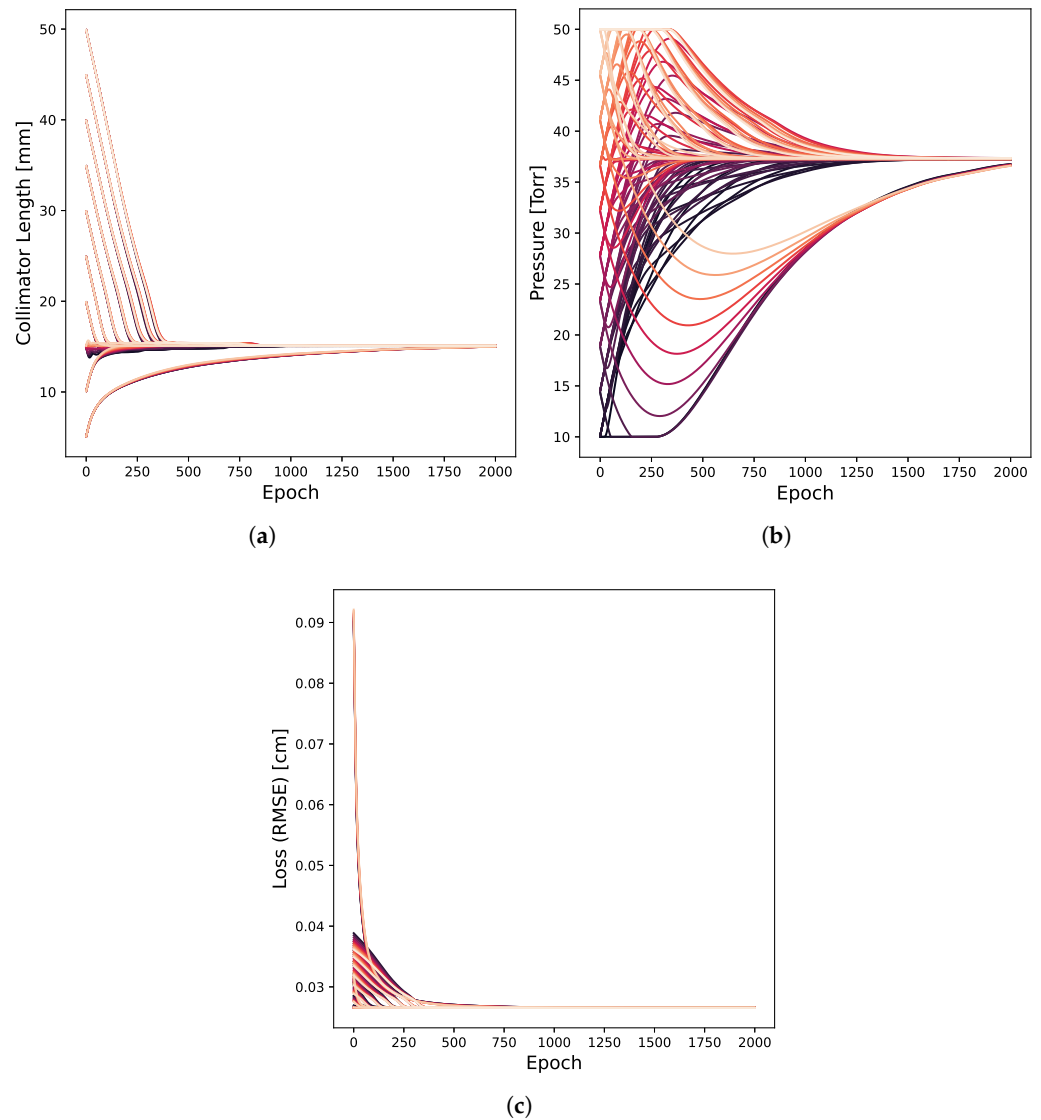
Figure 7. Evolution of the (**a**) collimator length, (**b**) pressure, and (**c**) loss function throughout the optimization loop for a grid of initial configurations of pressure and collimator length. Despite different starting points, the optimization process consistently converges to the same final result, suggesting that the found minimum of the loss function is the absolute minimum.

Lastly, Figure 8 presents a 3D representation of several optimization curves. This visualization effectively 'samples' the function $E(p, L)$, where E represents the reconstruction error in the z-axis, p is the pressure in Torr in the x-axis, and L is the collimator length in mm in the y-axis. From this, it can be inferred that the collimator length has a greater

impact on the reconstruction error than the pressure, as the gradient along *L* is steeper than the gradient along *p*.

Table 7. Initial values of the parameter grid considered for the optimization loop, specifying the range of values tested for both the pressure (*p*) and collimator length (*L*).

Parameter	Values
Pressure	20 values uniformly distributed between 10 and 50 Torr
Collimator length	20 values uniformly distributed between 5 and 50 mm
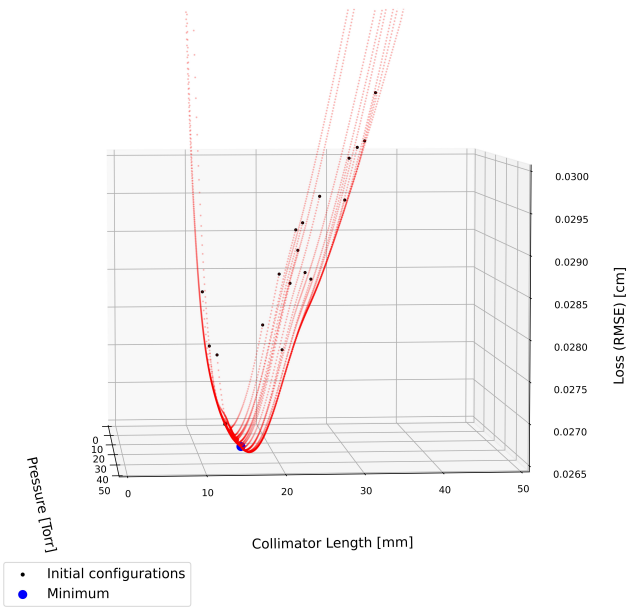
(**a**)
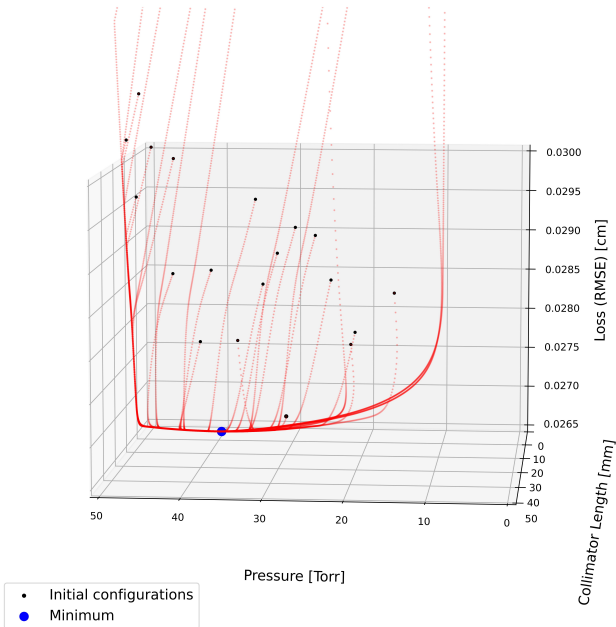
(**b**)

Figure 8. 3D representation of several optimization curves along (**a**) the collimator length axis and (**b**) the pressure axis, starting from 30 random initial configurations. The minimum of the curve along both axes is clearly observed. Additionally, a significantly higher gradient is observed along the collimator length axis compared to the pressure axis.

4. Discussion

This study aimed to optimize a Parallel-Plate Avalanche Counter with Optical Readout (O-PPAC) for heavy-ion tracking and imaging, focusing on two key detector parameters: pressure and collimator length. A surrogate model was developed to predict the outputs of a GEANT4 simulation based on inputs provided to the simulator. By integrating this model into an optimization loop, we identified the optimal combination of parameters that minimizes reconstruction error. The optimization was performed on a batch of 10^5 alpha particles, and the optimal parameters for a randomly generated distribution of particles covering an area of 8×8 cm^2 were found to be a pressure of 39.03 Torr and a collimator length of 15.11 mm.

As shown in Section 3, the optimal collimator length is in agreement with previous results obtained with traditional methods, as reported in Reference [19], which validates the approach used in this study. The pressure value, on the other hand, is relatively high within the specified range, which could be attributed to the higher statistics observed at increased pressures. However, this result cannot be explained purely by the statistics, as this would suggest that the optimal pressure should be the highest value within the imposed limits. We hypothesize that other factors, potentially related to the physical behavior of the detector or the limitations of the simulation, may be influencing the outcome.

To ensure the robustness of the results, a grid of different initial configurations was explored. This analysis confirmed that the minimum value of the loss function is independent of the initial values of pressure and collimator length, suggesting that the identified minimum represents the absolute minimum of the reconstruction error. Furthermore, a 3D representation of the optimization curves was used to illustrate the loss function and the difference in the gradient along the two axes. A significantly larger gradient was found along the collimator length axis, indicating that this parameter has a greater influence on the loss function compared to pressure.

These findings align with recent efforts to apply DP techniques to particle and nuclear physics detectors. As remarked before, despite the extensive application of DP in other technical fields, its application to particle detectors remains challenging mainly due to the stochasticity of quantum processes. However, the development of modular differentiable pipelines such as TomOpt [18] is gradually making detector and experiment optimization in nuclear and particle physics more feasible. As proof, this study was successfully conducted by adapting TomOpt software to the specific problem of the O-PPAC optimization.

5. Summary and Conclusions

Recent advancements in deep learning and computational capabilities have significantly enhanced the ability to optimize complex systems. Differentiable programming and automatic differentiation are at the forefront of this transformation, enabling automated optimization of complex processes. This study demonstrates the potential of applying these techniques to optimize detectors in nuclear and particle physics. By developing a surrogate model, we were able to identify the optimal parameters for an O-PPAC detector designed for heavy-ion tracking and imaging.

The next steps in this research will focus on incorporating the position reconstruction process directly into the differentiable pipeline, as the expressions applied to the photon distributions in order to obtain the reconstructed position are differentiable. This will involve exploring generative models that can directly predict photon distributions instead of the reconstructed position. Future work will also involve optimizing additional parameters, potentially including cost-related factors in the loss function. Furthermore, we plan to extend this research to a more complex system, which incorporates the O-PPAC as a

fundamental component. This could include exploring the effects of higher particle rates, different particle types, and other operational conditions that may influence performance.

In conclusion, this study highlights the potential of differentiable programming and machine learning in the design and optimization of particle detectors. The ongoing work and future directions will further refine this approach, expanding its applicability to a wider range of detector designs and experimental setups.

References

1. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: http://www.deeplearningbook.org (accessed on 3 June 2024).
2. Blondel, M.; Roulet, V. The Elements of Differentiable Programming. *arXiv* **2024**, arXiv:2403.14606. [CrossRef]
3. Baydin, A.G.; Pearlmutter, B.A.; Radul, A.A.; Siskind, J.M. Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.* **2017**, *18*, 5595.
4. Aehle, M.; Blühdorn, J.; Sagebaum, M.; Gauger, N.R. Forward-Mode Automatic Differentiation of Compiled Programs. *arXiv* **2022**, arXiv:2209.01895. [CrossRef]
5. Aehle, M.; Blühdorn, J.; Sagebaum, M.; Gauger, N.R. Reverse-Mode Automatic Differentiation of Compiled Programs. *arXiv* **2022**, arXiv:2212.13760. [CrossRef]
6. Barthelemy, J.F.M.; Hall, L.E. Automatic differentiation as a tool in engineering design. *Struct. Optim.* **1995**, *9*, 76–82. [CrossRef]
7. Feldmann; Melville; Moinian. Automatic differentiation in circuit simulation and device modeling. In Proceedings of the 1992 IEEE/ACM International Conference on Computer-Aided Design, Santa Clara, CA, USA, 8–12 November 1992; pp. 248–253. [CrossRef]
8. Bombardieri, R.; Cavallaro, R.; Sanchez, R.; Gauger, N.R. Aerostructural wing shape optimization assisted by algorithmic differentiation. *Struct. Multidiscip. Optim.* **2021**, *64*, 739–760. [CrossRef]
9. Agostinelli, S.; Allison, J.; Amako, K.; Apostolakis, J.; Araujo, H.; Arce, P.; Asai, M.; Axen, D.; Banerjee, S.; Barrand, G.; et al. Geant4—A simulation toolkit. *Nucl. Instruments Methods Phys. Res. Sect. Accel. Spectrometers, Detect. Assoc. Equip.* **2003**, *506*, 250–303. [CrossRef]
10. Allison, J.; Amako, K.; Apostolakis, J.E.A.; Araujo, H.A.A.H.; Dubois, P.A.; Asai, M.A.A.M.; Barrand, G.A.B.G.; Capra, R.A.C.R.; Chauvie, S.A.C.S.; Chytracek, R.A.C.R.; et al. Geant4 developments and applications. *IEEE Trans. Nucl. Sci.* **2006**, *53*, 270–278. [CrossRef]

11. Allison, J.; Amako, K.; Apostolakis, J.; Arce, P.; Asai, M.; Aso, T.; Bagli, E.; Bagulya, A.; Banerjee, S.; Barrand, G.J.N.I.; et al. Recent developments in Geant4. *Nucl. Instr. Meth. A* **2016**, *835*, 186–225. [CrossRef]

12. Baydin, A.; Cranmer, K.; Manzano, P.; Delaere, C.; Derkach, D.; Donini, J.; Dorigo, T.; Giammanco, A.; Kieseler, J.; Layer, L.; et al. Toward Machine Learning Optimization of Experimental Design. *Nucl. Phys. News* **2021**, *31*, 25–28. [CrossRef]

13. Ratnikov, F. Using machine learning to speed up and improve calorimeter R&D. *J. Instrum.* **2020**, *15*, C05032. [CrossRef]

14. Cisbani, E.; Dotto, A.D.; Fanelli, C.; Williams, M.; Alfred, M.; Barbosa, F.; Barion, L.; Berdnikov, V.; Brooks, W.; Cao, T.; et al. AI-optimized detector design for the future Electron-Ion Collider: The dual-radiator RICH case. *J. Instrum.* **2020**, *15*, P05009. [CrossRef]

15. Koser, D.; Waites, L.; Winklehner, D.; Frey, M.; Adelmann, A.; Conrad, J. Input Beam Matching and Beam Dynamics Design Optimizations of the IsoDAR RFQ Using Statistical and Machine Learning Techniques. *Front. Phys.* **2022**, *10*, 875889. [CrossRef]

16. Van der Veken, F.; Azzopardi, G.; Blanc, F.; Coyle, L.; Fol, E.; Giovannozzi, M.; Pieloni, T.; Redaelli, S.; Ferrando, B.; Schenk, M.; et al. Machine learning in accelerator physics: Applications at the CERN Large Hadron Collider. *PoS* **2020**, *044*, AISIS2019. [CrossRef]

17. Dorigo, T. Geometry optimization of a muon-electron scattering detector. *Phys. Open* **2020**, *4*, 100022. [CrossRef]

18. Strong, G.C.; Lagrange, M.; Orio, A.; Bordignon, A.; Bury, F.; Dorigo, T.; Giammanco, A.; Heikal, M.; Kieseler, J.; Lamparth, M.; et al. TomOpt: Differential optimisation for task- and constraint-aware design of particle detectors in the context of muon tomography. *Mach. Learn. Sci. Tech.* **2024**, *5*, 035002. [CrossRef]

19. Cortesi, M.; Ayyad, Y.; Yurkon, J. Development of a parallel-plate avalanche counter with optical readout (O-PPAC). *J. Instrum.* **2018**, *13*, P10006. [CrossRef]

20. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8024.

21. Di Carlo, S.; Cortesi, M. Parallel-plate avalanche counters for heavy-ion beam tracking: History and mysteries. *Phys. Rev. Accel. Beams* **2024**, *27*, 044801. [CrossRef]

22. Cortesi, M.; Yurkon, J.; Stolz, A. Measurements of secondary scintillation in low-pressure CF_4 with a SiPM, from a parallel-plate avalanche geometry. *J. Instrum.* **2016**, *11*, P04017. [CrossRef]

23. Shirobokov, S.; Belavin, V.; Kagan, M.; Ustyuzhanin, A.; Baydin, A.G. Black-Box Optimization with Local Generative Surrogates. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 14650–14662.

24. Aehle, M.; Arsini, L.; Barreiro, R.B.; Belias, A.; Bury, F.; Cebrian, S.; Demin, A.; Dickinson, J.; Donini, J.; Dorigo, T.; et al. Progress in End-to-End Optimization of Detectors for Fundamental Physics with Differentiable Programming. *arXiv* **2023**, arXiv:2310.05673. [CrossRef]

25. Kasim, M.F.; Watson-Parris, D.; Deaconu, L.; Oliver, S.; Hatfield, P.; Froula, D.H.; Gregori, G.; Jarvis, M.; Khatiwala, S.; Korenaga, J.; et al. Building high accuracy emulators for scientific simulations with deep neural architecture search. *Mach. Learn. Sci. Technol.* **2021**, *3*, 015013. [CrossRef]

26. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD), Anchorage, AK, USA, 4–8 August 2019.

27. Pereira, M.; Vischia, P.; Cid, X. OPPAC_OPT: Automatic Optimisation of a Parallel-Plate Avalanche Counter with Optical Readout. 2024. Available online: https://github.com/Mariapm267/OPPAC_opt (accessed on 26 November 2024).

28. Strong, G.; Lamparth, M.; Lagrange, M.; Vischia, P.; Nardi, F.; Giammanco, A.; Orio Alonso, A.; Zaraket, H. *GilesStrong/Tomopt: v.0.1.0: 1st Publication Version*; Zenodo: Geneva, Switzerland, 2024. [CrossRef]