Article

Neuromorphic Readout for Hadron Calorimeters

Enrico Lupi [1,2,*], Abhishek [3], Max Aehle [4,†], Muhammad Awais [1,2,5,†], Alessandro Breccia [2], Riccardo Carroccio [2], Long Chen [4,†], Abhijit Das [6], Andrea De Vita [1,2], Tommaso Dorigo [5,†,‡], Nicolas Ralph Gauger [4,†], Ralf Keidel [7,†], Jan Kieseler [7], Anders Mikkelsen [6], Federico Nardi [2,8], Xuan Tung Nguyen [1,4], Fredrik Sandin [5,†], Kylian Schmidt [7], Pietro Vischia [9,†,‡] and Joseph Willmore [1]

1   INFN, Sezione di Padova, Via F. Marzolo 8, 35131 Padova, Italy; awaisafzal628@gmail.com (M.A.); andrea.devita.1@studenti.unipd.it (A.D.V.); xuantung.nguyen@pd.infn.it (X.T.N.); 25jw05@stu.cpsk12.org (J.W.)
2   Dipartimento di Fisica e Astronomia, Università di Padova, Via F. Marzolo 8, 35131 Padova, Italy; alessandro.breccia@studenti.unipd.it (A.B.); riccardo.carroccio@studenti.unipd.it (R.C.); federico.nardi.4@studenti.unipd.it (F.N.)
3   National Institute of Science Education and Research, Jatni 752050, India; abhishek.2019@niser.ac.in
4   Chair for Scientific Computing, University of Kaiserslautern-Landau (RPTU), Paul-Ehrlich-Straße, 67663 Kaiserslautern, Germany; max.aehle@scicomp.uni-kl.de (M.A.); long.chen@scicomp.uni-kl.de (L.C.); nicolas.gauger@scicomp.uni-kl.de (N.R.G.)
5   Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 971 87 Luleå, Sweden; tommaso.dorigo@gmail.com (T.D.)
6   Department of Physics and NanoLund, Lund University, P.O. Box 118, 221 00 Lund, Sweden; abhijit.das@sljus.lu.se (A.D.); anders.mikkelsen@sljus.lu.se (A.M.)
7   Institute for Experimental Particle Physics, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany; keidel@hs-worms.de (R.K.); jan.kieseler@cern.ch (J.K.); kylian.schmidt@kit.edu (K.S.)
8   Laboratoire de Physique Clermont Auvergne, 63170 Aubière, France
9   Department of Physics, Universidad de Oviedo and ICTEA, 33007 Oviedo, Spain; pietro.vischia@gmail.com
*   Correspondence: enrico.lupi@studenti.unipd.it
†   MODE Collaboration.
‡   Universal Scientific Education and Research Network, Italy.

Abstract: We simulate hadrons impinging on a homogeneous lead tungstate ($PbWO_4$) calorimeter using GEANT4 software to investigate how the resulting light yield and its temporal structure, as detected by an array of light-sensitive sensors, can be processed by a neuromorphic computing system. Our model encodes temporal photon distributions as spike trains and employs a fully connected spiking neural network to estimate the total deposited energy, as well as the position and spatial distribution of the light emissions within the sensitive material. The extracted primitives offer valuable topological information about the shower development in the material, achieved without requiring a segmentation of the active medium. A potential nanophotonic implementation using III-V semiconductor nanowires is discussed. It can be both fast and energy efficient.

Keywords: machine learning; neuromorphic computing; calorimeter; particle detector; spiking neural networks; nanowire; III-V semiconductor nanowires; nanophotonics

1. Introduction

Hadron calorimeters play a critical role in high-energy physics experiments, providing precise energy measurements of hadronic showers. A hadron calorimeter is a block of dense matter that is capable of stopping highly energetic particles through the strong interaction they withstand in hitting nuclear matter, producing a macroscopic effect that suitable electronics can record; the latter is usually constituted by the release of visible light in an amount proportional to the incident particles energy. Over the past 60 years,

the technology of hadron calorimeters evolved significantly. Still, the emphasis on the design of these instruments has stayed with the reduction in uncertainties in measuring the total incident energy of streams of hadrons. The reason for this ultimately lies within the strong interaction dynamics, which turn energetic quarks or gluons produced in high-energy processes (a scattering reaction or a decay of a heavy particle) into a collimated stream of hadrons, the individual properties of which are less important to the experiment than their collective ones.

While efforts to improve the energy measurement of hadronic jets through the individual measurement of their components dates back to the 1980s [1], the combination of a precise tracking of charged hadrons and the association with different energy deposits in finely segmented calorimeters was only proven after the turn of the century, enormously improving the overall energy measurement of the jet [2]. At the same time, it was realized that the hadronic decays of massive particles, such as *W* and *Z* bosons, top quarks, and *H* bosons, could be distinguished from the otherwise insurmountable Quantum Chromodynamics (QCDs) backgrounds if the originating particles were highly boosted. The discrimination, again, required a fine segmentation of the calorimeter and the exploitation of subtle topological features of the energy distribution within the cone of the resulting wide jets [3].

A high segmentation of the sensitive elements of a calorimeter constitutes a significant challenge due to its impact on total cost, energy consumption, and output data volume. In particular, conventional computing solutions alone may not be sufficient to cope with the computing demand of the online dimensional reduction of the resulting data in the foreseeable future. Another significant challenge lies in the temporal realm. Since particles traveling close to the speed of light (c) traverse 3 cm in just 100 ps (picoseconds), the time frame for pattern detection is condensed to sub-nanoseconds. Current detectors have not exploited such temporal information, which could help extract information about particle identity and help overall event reconstruction. Integrating an efficient online dimensional reduction and a spatio-temporal pattern recognition using parallel analog-digital neuromorphic computing architectures integrated into the detector volume might assist in overcoming these limitations.

Neuromorphic technologies and compute-in-memory architectures are ideal for processing streaming sensor data, especially when combined with event-based detectors and asynchronous distributed algorithms [4]. This is because they use the physical properties of materials and circuits to process information in parallel with high energy and latency performance [5]. Furthermore, neuromorphic architectures encode information in physical time, such as through a succession of binary events called spikes, which offer additional modalities for encoding information besides bits; see, for example, [6,7]. Spiking Neural Networks (SNNs) are used to model such hybrid analog-digital neuromorphic systems and biological neurons using differential equations to describe the dynamics. Unlike conventional artificial neural networks (ANNs), which use continuous values to represent the activation of the individual computational units/neurons, the computational units in SNNs generate discrete spikes in response to incoming stimuli. These spikes occur asynchronously at precise points, adding the temporal dimension to neural processing that enhances the model's capacity to asynchronously process time-dependent information efficiently. This spike-based approach makes SNNs particularly well-suited for event-based spatio-temporal processing, where spike dynamics and the asynchronous parallel processing of spikes are efficiently implemented using specialized algorithms, circuits, devices, and materials. A variety of mathematical models, such as the Leaky Integrate-and-Fire (LIF) model and the (adaptive) exponential integrate-and-fire model, are used to model the computational units in SNNs depending on the model capacity and level of biological

realism required. These models vary in complexity but share the core concept that information encoded in the timing of spikes is processed via the responses of nonlinear analog dynamical systems.

This work is structured as follows. In Section 2, we describe the data generation process, including the simulation setup, the calorimeter model, and the assumptions made for light production and readout. Section 3 defines the primary regression tasks and the target variables used for event characterization. In Section 4, we introduce the neuromorphic computing approach, detailing the preprocessing steps, SNN architecture, and training protocol. Section 5 presents the results of the regression models, including both single- and multi-target performance evaluations, along with an optimization study of the network architecture and hyperparameters. Section 6 explores a potential nanophotonic hardware implementation using III-V semiconductor nanowires, discussing its feasibility for high-speed and energy-efficient neuromorphic processing. Finally, in Section 7, we summarize our findings, outline the advantages of neuromorphic readout for hadron calorimetry, and discuss potential future directions, including experimental validation and further hardware developments.

2. Data Generation

To carry out this study, we used `GEANT4` [8] (version 11.3.1) simulations of the response of a homogeneous calorimeter hit by single 100-GeV charged hadrons (either p, K^+, or π^+). The data were initially produced for the work described in Ref. [9]. The same data were used for this study after they were further processed to simulate light production, propagation, and detection (see *infra*).

2.1. Calorimeter Model

The homogeneous calorimeter is made of lead tungstate ($PbWO_4$), which has a light yield equal to $LY_{PWO} = 200 \text{ ph}/MeV$ [10]. It has a total size of $300 \times 300 \times 1200 \text{ mm}^3$, which corresponds to a lateral width of $7.66\rho_M$ (Molière radii) and $5.92\lambda_I$ (interaction lengths), ensuring an average lateral containment of 100% and a longitudinal containment of approximately 87%. The detector can be logically divided into a grid of $10 \times 10 \times 10$ units called cubelets, with dimensions of $3 \times 3 \times 12 \text{ cm}^3$. Each cubelet is itself organized into a $10 \times 10 \times 10$ grid of cells, with dimensions of $3 \times 3 \times 12 \text{ mm}^3$, for a total of one million cells in the whole calorimeter.

2.2. Readout Model

The readout system is made up of a system of light-sensitive sensors, one per cubelet. They are placed on their upper face, on the xz plane, and organized in a grid of 10×10. Each sensor has the same size as a cell in the x and z directions, while its height in the y direction is considered negligible. The sensors are assumed to be 100% efficient so that they always detect all incident photons, regardless of their frequency. This simplification abstracts away from the hardware problem of manufacturing an effective readout system in order to allow us to focus on the problem of information processing. Possible hardware implementations of such a system are discussed in Section 6.

2.3. Simulation of Light Production

In order to simulate light production and diffusion inside the calorimeter, we make the following assumptions:

1. As detailed in Ref. [9], the position of each interaction is recorded only using the index of the cell where it occurred. All interactions are treated as if they occurred in the center of their respective cell.

2. All deposited energy is converted into photons with 100% efficiency.
3. No Cherenkov radiation is emitted.
4. Photons are emitted uniformly in all directions.
5. The detector material is considered completely transparent to the photons, and they travel unimpeded until they reach the face of the cubelet where they were produced. They are absorbed by the cubelet faces, and the hit is registered only if the photons reach the upper one, where the photosensitive sensors are placed.

Concerning the above assumptions, in particular, the 100% efficiency of light collection, it is important to point out that the goal of this study is not to model all detector physics in detail but rather to evaluate whether useful topological information can be extracted from the temporal structure of light emission. The assumption of perfect light yield removes hardware-dependent inefficiencies and allows for a clear focus on algorithmic performance. In a real experimental setup, the efficiency of light production is usually determined through calibration measurements, and systematic corrections can be applied to account for realistic light yield inefficiencies. Thus, this assumption does not invalidate the feasibility of the approach. Any systematic bias introduced by this assumption would be absorbed in the learned parameters of the SNN. If needed, the network can be retrained later with realistic efficiency factors.

Given the above assumptions, we simulate the light production as follows. For each interaction, we convert the energy deposited into the number of photons produced: $N_{ph}^{tot} = E_{dep} \cdot LY_{PWO}$. Given the cell in which the interaction occurred, we then calculate the solid angle subtended by each light sensor of the cubelet from the center of the cell, Ω^i, and the time it takes for the light to reach it, Δt^i. The amount of photons that reach the i-th sensor after an interaction is, thus, simply given by $N_{ph}^i = N_{ph}^{tot} \frac{\Omega^i}{4\pi}$, while the arrival time is given by $t^i = t_{interaction} + \Delta t^i$.

2.4. Discretization of Light Pulses

The arrival of the photons is recorded in a time window spanning from the start of the event up to 20 ns: by this time, all major interactions have already occurred, and more than 90% of the total energy has been deposited in the calorimeter. Time is discretized into 100 bins of 0.2 ns each, and the total number of photons received within each of them is recorded.

2.5. Primary Vertex

For this study, we chose not to include all cubelets in our analysis but to focus only on the one where the first nuclear interaction vertex occurred, which typically contains a large share of the released energy and offers the highest information content on the identity of the incident particle [9].

The primary particles were originally produced in a fixed position at a distance of 3 m from the calorimeter surface along the z direction and at its center on the xy plane. This meant that the position of the first nuclear interaction vertex showed very little variation between events, which translated into little variation in the centroid of the energy deposition, one of the variables to be regressed (see Section 3 for the exact definition).

To have a more realistic setting with a randomly distributed energy centroid, we introduce a posteriori, a random shift in the initial position of the particle gun. This is practically carried out by adding random shifts to the position of each interaction along x and y, both generated uniformly in the range $[0; 3]$ cm. This shift is kept fixed for all interactions in the same event.

2.6. Data Samples

For each simulated particle (p, π^+, k^+), a total of 5000 simulated events were analyzed, recording the number of photons detected by each light sensor at each time step in the primary vertex cubelet.

3. Task Definition

In this work, we focus on regressing the following variables of interest: the total energy deposited in the calorimeter cubelet, the spatial coordinates of the centroid of the energy depositions, and their relative dispersions. Each of these sets of variables was regressed independently in the so-called single-target regression tasks or together with other variables in multi-target regression.

The first variable of interest is the total amount of energy released inside the cubelet to demonstrate that a neuromorphic system would be able to mimic the results obtained by traditional calorimeters.

As depicted in Figure 1, the actual values of the total energy released span several orders of magnitude, from hundreds of *MeV*s to hundreds of *GeV*s. Directly regressing them would, thus, be too challenging a task. To circumvent the problem, we applied a nonlinear transformation and selected the logarithm in base 10 of the energy expressed in *MeV* as the objective of the regression.
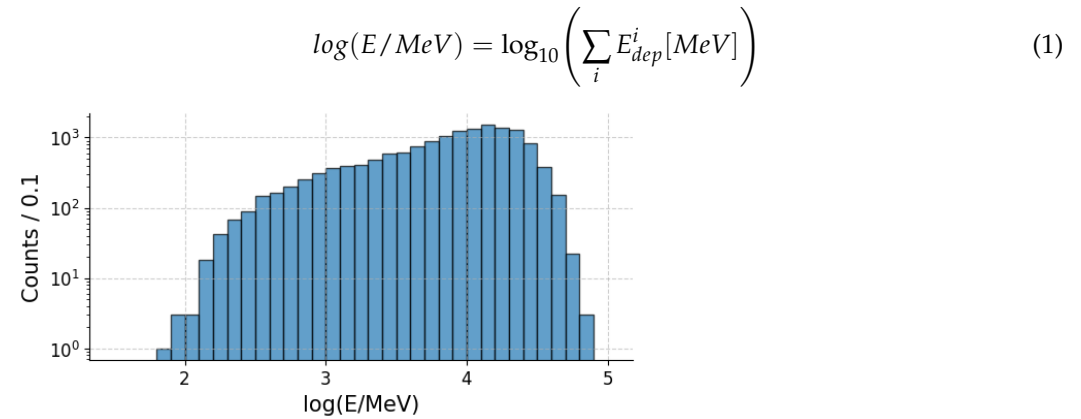
$$log(E/MeV) = \log_{10}\left(\sum_i E_{dep}^i[MeV]\right) \tag{1}$$

Figure 1. Distribution of the $log(E/MeV)$ variable across the whole dataset.

The other variables are, instead, related to the geometry of the event. Their exact definition is the following:

$$X_c = \frac{\sum_i X_i \cdot E_i}{E_{tot}}, \tag{2}$$

$$\sigma^2 = \frac{\sum_i (X_i - X_c)^2 \cdot E_i}{E_{tot}}, \tag{3}$$

where the index i runs over each interaction happening inside the cubelet, and the two variables are computed separately for each coordinate $\vec{X} = \{x, y, z\}$.

It is important to note that both variables are expressed in units of "cell lengths": the X_i variables range from 0 to 9, depending on the cell of the cubelet where the interaction occurred. As detailed in Section 2.1, a cell length corresponds to 3 cm along the *x* and *y* directions and 12 cm along *z*.

The distributions of these variables are consistent with our expectations. In particular, the distributions of x_c and y_c, seen in Figure 2, are mostly uniform due to the transformation described in Section 2.5. The peaks present are an artifact arising from the discretization of the position of each energy deposition.
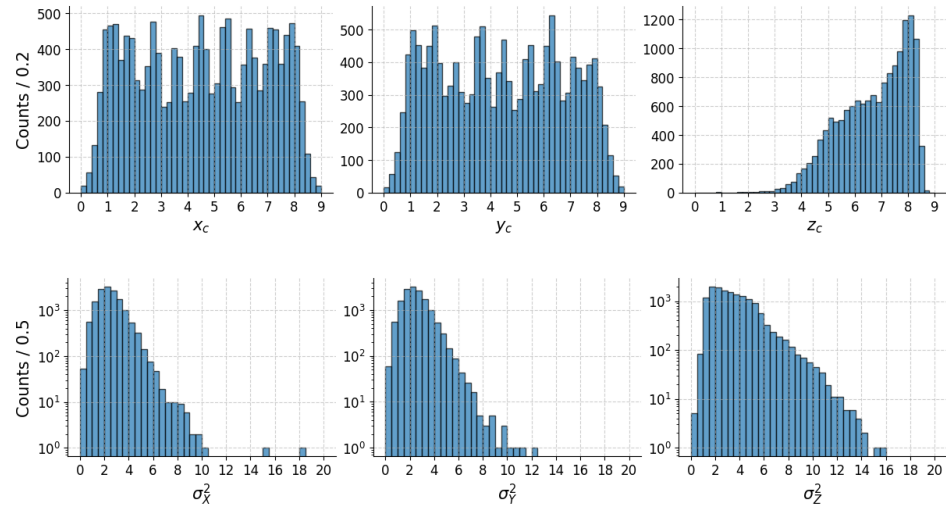
Figure 2. Distributions of the energy deposition centroid (**above**) and energy dispersion (**below**) across the whole dataset. The variables are expressed in "cell lengths".

4. Spiking Neural Network Model

4.1. Preprocessing: Encoding Procedure

Incoming data need to be converted into a series of spikes, i.e., a spike train, before being fed into the network. We adopted the following encoding scheme, graphically represented in Figure 3. For each sensor, we consider four different channels that go out of it and into the network. In each time step, these channels can either be activated and carry a spike or be inactive, depending on the amount of light received in that instant. Each channel has a different activation threshold, so spikes will carry information about both the timing and intensity of the interactions. The exact formula used is the following:

$$S^{(i)}[t] = \begin{cases} 1, & \text{if } N_{ph}[t] \geq 10^{i+2} \\ 0, & \text{otherwise} \end{cases}, \quad i = 0, 1, 2, 3 \tag{4}$$

where the index i refers to the channel, and t indicates the timestep.
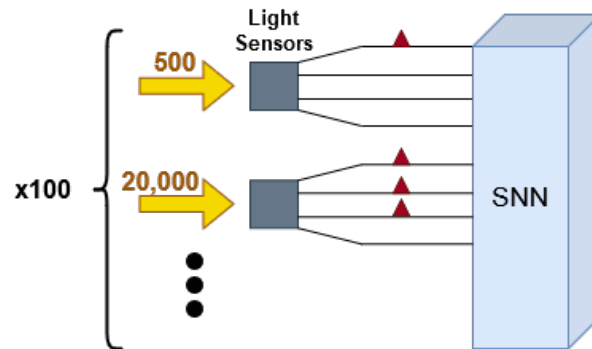

Figure 3. Graphical representation of the encoding scheme.

4.2. Network Architecture

We used a fully connected feed-forward network with 400 input afferents, two hidden layers of 120 neurons each, and an output layer of varying total size depending on the task, with populations of 20 neurons per target. We adopted a simple LIF model as the neuron model, using learnable U_{thr} and β parameters, and the *arctan* function as a surrogate gradient. The neuromorphic elements of the code were implemented using the *snnTorch* Python library [11] (version 0.9.1).

The reason why this particular architecture was adopted is discussed in Section 5.3.

4.3. Decoding Schemes

Neuromorphic systems offer many solutions to perform regression tasks, using either the output spikes or the membrane potential of the last layer as quantities to be analyzed. The latter is a continuous value, thus offering higher precision at the cost of higher latency and computational complexity. The former, on the other hand, needs more refined decoding schemes to handle its binary nature (rate-based, latency-based, spiking interval codes) but usually provides better performance.

A possible decoding scheme using the membrane potential value consists of simply equating its value at the last time step to the variable to regress. In order to make this scheme more stable, we can consider a population of output neurons and take the average of their membrane potential:

$$\hat{X} = \frac{1}{N_{pop}} \sum_{i=1}^{N_{pop}} U^{(i)}[T] \tag{5}$$

In this equation, $\hat{X}$ is a general variable that we want to regress, N_{pop} indicates the number of neurons inside that population, and $U^{(i)}[T]$ refers to the value of the membrane potential of the i-th neuron at the last timestep.

In contrast, a promising scheme that uses the output spikes is the rate-based encoding, which links the variable to regress to the number of spikes produced. By using, once again, a population of neurons to stabilize the results, we obtain the following expression:

$$\hat{X} = \frac{1}{N_{pop}} \sum_{i=1}^{N_{pop}} \sum_{t=1}^{T} S^{(i)}[t], \tag{6}$$

where $S^{(i)}[t]$ is a binary variable that is equal to 1 if the i-th neuron has emitted an output spike at timestep t; it is 0 otherwise.

In this paper, we will show the results obtained using the second decoding scheme outlined. The full data processing pipeline is illustrated in Figure 4.
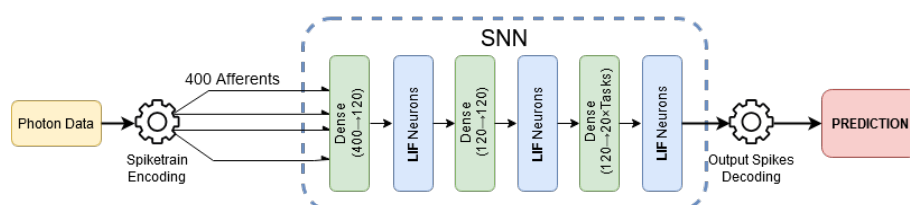

Figure 4. Data processing pipeline and network architecture.

4.4. Training

The complete dataset is divided into training, validation, and test datasets with a 0.7-0.15-0.15 split. The validation dataset was used for hyperparameter optimization (Section 5.3) and to monitor the loss behavior during training.

Different loss functions were adopted depending on the distributions of the target variable. For $log(E/MeV)$, the centroid z coordinate, and the energy dispersion, we use the Mean Absolute Error (MAE) loss (or L1 loss), given the small dynamical range of the outputs and their asymmetrical distribution, as shown in Figures 1 and 2. For the energy centroid x and y coordinates, instead, we adopt the Mean Squared Error (MSE) loss. This is because optimizing the MSE loss corresponds to calculating the batch distribution's mean, and the L1 loss corresponds to the median of the batch distribution, which is a better estimate for a skewed distribution.

In the case of multi-target regression tasks, we compute the losses for each target individually and then sum them to obtain the final loss estimate.

We used Adam [12] as the optimizer, with a learning rate η and a weight regularization λ of 0.001, and parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$. Together with it, we used a learning rate scheduler, which decreases η each epoch of a factor $\gamma = 0.7$.

The networks were trained using a batch size of 50 samples for a total of 5 epochs (each processing 210 batches). This is sufficient for the loss to converge to a minimum, as illustrated by the plots in Appendix A.

5. Results

In this section, we will cover the results obtained by the neuromorphic system on the test dataset for both single and multi-target regression and the procedure followed to optimize the architecture hyperparameters.

When reporting the error, we used the average of the l_1 distance between prediction and target across all of the test dataset, eventually weighted by the target itself.

$$\epsilon_{rel} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left| \frac{target_i - prediction_i}{target_i} \right| \tag{7}$$

$$\epsilon_{abs} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} |target_i - prediction_i|, \tag{8}$$

where N_{test} refers to the total size of the test dataset.

5.1. Single-Target Regression

Table 1 shows the results obtained for single-target regression tasks, while Figures 5–7 show the comparison between targets and predicted values and the respective residuals. We notice that $log(E/MeV)$ can be regressed with very good accuracy; conversely, the absolute error on the true value of the energy is very large. This is due to the large dynamical range of the energy depositions (see Figure 1): small variations in $log(E/MeV)$ lead to very large differences in E on the higher tail of the distribution. The same phenomenon is also responsible for the large tails on the left of the residual distribution: the model cannot correctly take into account higher energy deposits and tends to over-estimate them. We also observe that the position of the centroid of the energy distribution can be estimated by the SNN with good accuracy (with an uncertainty of less than one cell in all coordinates). For the dispersions of the energy distribution in the cubelets, we obtain good results for the z component, while the x and y dispersions are less easy to evaluate.
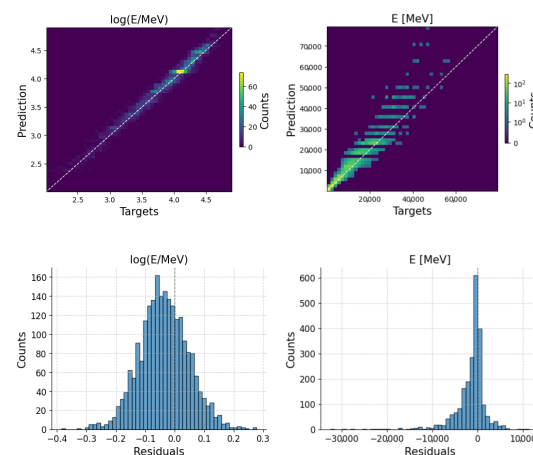

Figure 5. Output of the $log(E/MeV)$ network and the respective results for the true value of the energy (in *MeV*). The plots show the correlation between targets and predictions (**above**) and the residuals (**below**).
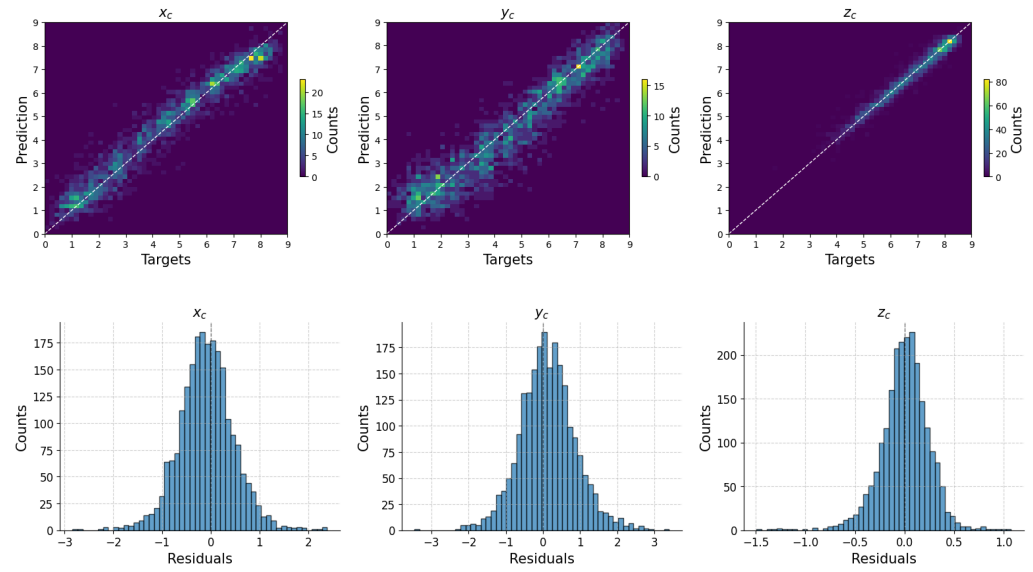
Figure 6. Output of the energy centroid network for the x, y, and z coordinates. The plots show the correlation between targets and predictions (**above**) and the residuals (**below**).
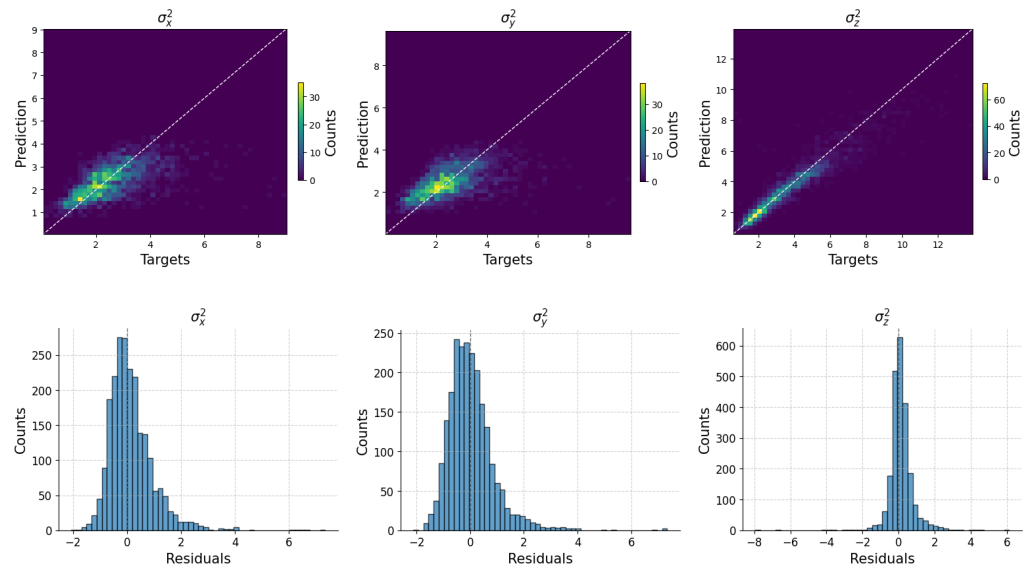

Figure 7. Output of the energy dispersion network for the x, y, and z coordinates. The plots show the correlation between targets and predictions (**above**) and the residuals (**below**).

Table 1. Single-target regression errors. Position and variance estimates are expressed in cell unit lengths.

	log(E/MeV)	E	x_c	y_c	z_c	σ_x^2	σ_y^2	σ_z^2
ϵ_{rel} (%)	1.975	18.37	18.13	24.74	2.85	26.07	31.39	12.04
ϵ_{abs}	0.073	2039 MeV	0.44	0.58	0.18	0.58	0.62	0.41

5.2. Multi-Target Regression

We trained two different networks: one regressing $log(E/MeV)$ and the position of the centroid, the other regressing $log(E/MeV)$ and the energy dispersion in all three directions. Table 2 and Figure 8 show the results for the former network, whereas Table 3 and Figure 9 concern the latter.
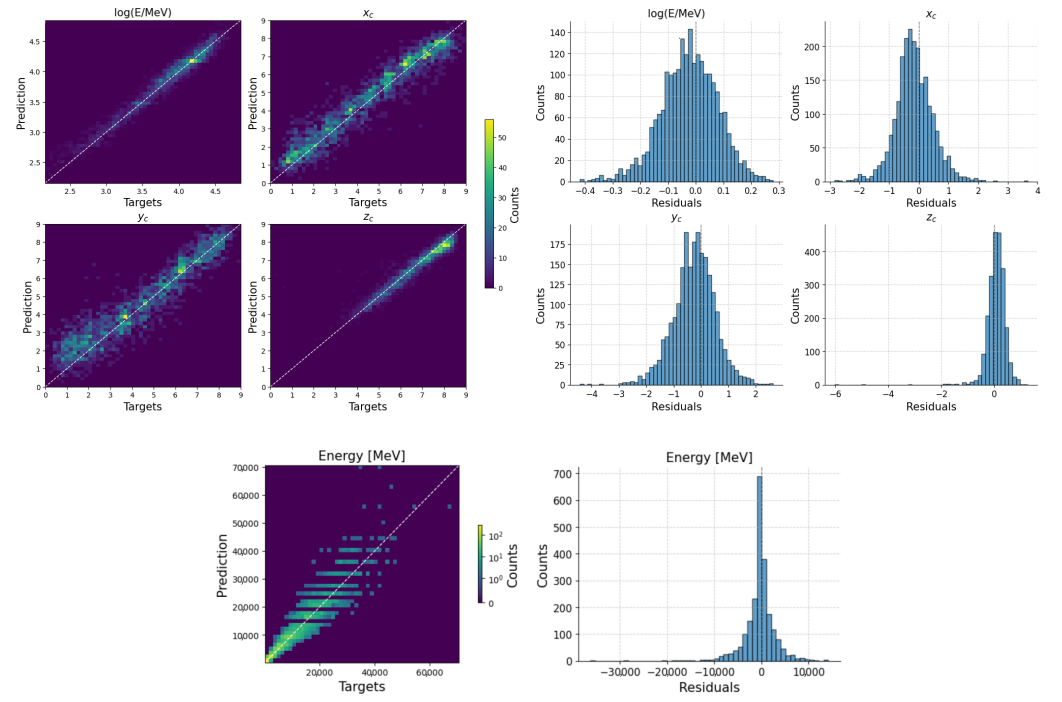
Figure 8. Output of the energy deposition and centroid network. The plots show the correlation between targets and predictions (**left**) and the residuals (**right**).
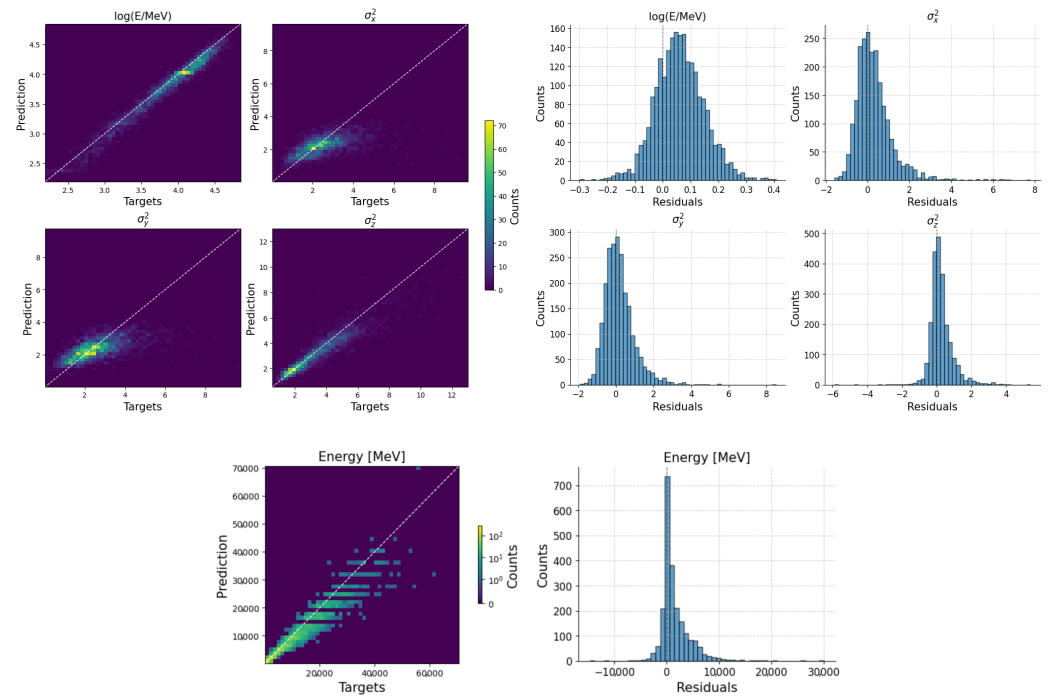

Figure 9. Output of the energy deposition and dispersion network. The plots show the correlation between targets and predictions (**left**) and the residuals (**right**).

Table 2. Multi-target regression errors for deposited energy and energy centroid.

	log(E/MeV)	E	x_c	y_c	z_c
ϵ_{rel} (%)	2.262	20.59	20.66	32.66	5.33
ϵ_{abs}	0.082	2001 MeV	0.51	0.62	0.25

Table 3. Multi-target regression errors for deposited energy and energy dispersion.

	log(E/MeV)	E	σ_x^2	σ_y^2	σ_z^2
ϵ_{rel} (%)	2.252	17.21	27.71	26.40	12.76
ϵ_{abs}	0.084	1967 MeV	0.63	0.59	0.47

The results of multi-target regressions are consistent with those shown *supra* for single-target regression, indicating that the model is capable of handling multiple tasks at once without huge losses in performance. This is useful, as it reduces the complexity of the hardware required to implement the model.

5.3. Optimization of Architecture and Hyperparameters

To estimate the best network architecture, a Bayesian optimization algorithm was implemented and applied to both the multi-target regressions involving energy together with centroids and the one involving energy and dispersions. The scheme is based on a Gaussian surrogate model of the objective "black box" objective function. This model acts as prior on the function space, which is updated by new observations. The latter are generated to maximize a so-called acquisition function that, in our case, is the Expected Improvement. This method directs the sampling towards the maximum of the surrogate Gaussian model, allowing us to obtain a faster optimization due to conditioning on sampling towards maxima, and it also gives a posterior distribution of the performance over the parameters, which is useful to obtain further insights on the total parameters with space considered. Since the optimization aims to sample towards maxima, we defined a measure of performance that quantifies how well a certain architecture or the parameters are doing. Performance P is defined as the mean of relative error reciprocals:

$$P = \frac{1}{N_{targets}} \sum_{i}^{N_{targets}} \frac{1}{\epsilon_i} \qquad (9)$$

where $N_{targets}$ is the total number of targets being regressed together, and ϵ_i is the relative error on target "i" regression. The errors are computed on the validation set.

The resulting posterior distributions over the parameters space are shown in Figures 10–11.
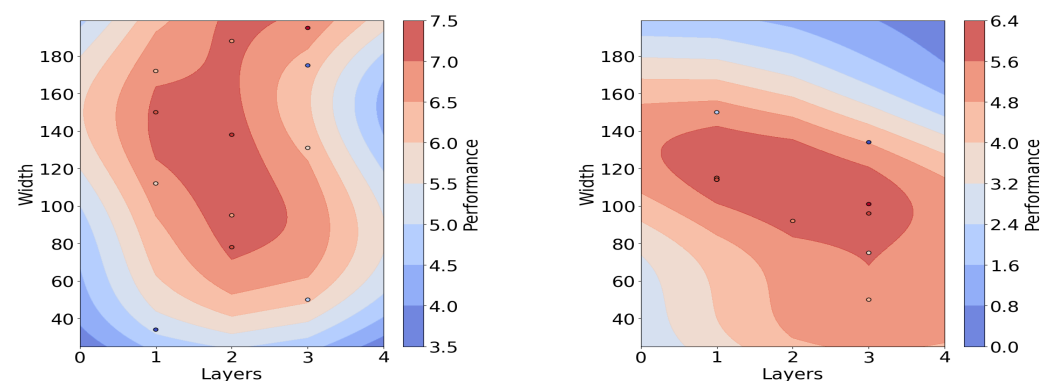

Figure 10. Posterior distribution of network performance on *log(E/MeV)* and centroid (**left**) and *log(E/MeV)* and dispersion (**right**) for multi-target regression as a function of the layers and width per layer, having set $\eta = 10^{-4}$ and $\lambda = 0$.
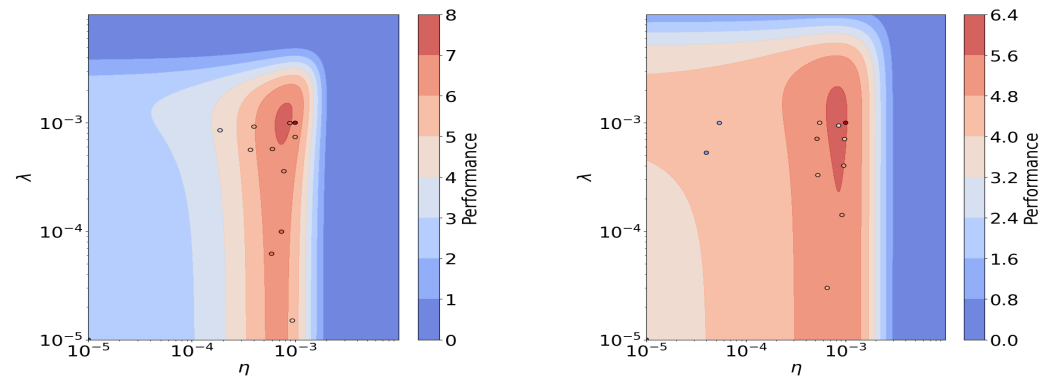
Figure 11. Posterior distribution of network performance on *log(E/MeV)* and centroid (**left**) and *log(E/MeV)* and dispersion (**right**) for multi-target regression learning rate η and weight regularization constant λ, having set the number of layers to $N = 2$ and the network width to $W = 120$, as obtained from the architecture optimization above.

6. Nanophotonic Hardware Implementation

As described in the introduction, the speed of the processes to be sampled and analyzed needs sub-nanosecond resolution, putting severe constraints on the hardware solutions. As an additional parameter, energy consumption should be kept low so as to not increase the overall energy budget of the experiment too much. Most neuromorphic systems operate at up to 200 MHz [13], which is equivalent to 5 ns of cycle time, not including the light conversion step.

We propose a nanophotonic implementation for computation, with all electrical and optical components integrated into a single chip stack, including optical detectors. III-V semiconductors, with their excellent photon/electron conversion efficiencies, are aptly suited for this. Nanowire (NW) devices made from III-Vs can detect light, emit light, and also operate as electronic components [14]. III-V NWs have the additional advantage of having excellent cross-sections for light absorption, and due to their small size, they can operate at very fast timescales with very low energy consumption [15,16]. By combining these functions in InP NWs, neural networks using light for communication have recently been proposed [15,17], and light communication between individual NW components has been demonstrated [18].

These components and the developments thereof can also be used in the present proposal, but a range of other III-V NWs with a direct bandgap are also relevant [14]. While individual NW devices were employed for neural network implementation [18], arrays of parallel NWs can also be fabricated, delivering similar performance in terms of sensitivity and operational times; however, they use larger currents [19]. The higher current leads to a more robust circuit while also having a larger area of detection for each pixel. A photodetector circuit that can also act as a sigmoid neuron then consists of two different types of photodiodes that supply or remove charge at the gate of an InAs NW field effect transistor (FET), which then controls an InP NW LED emitter [17]. Operated at suitable voltages, inputs on the two photodiodes summarize as exciting/inhibiting signals, respectively, which are treated via a sigmoid function to result in (possible) LED light emission. Because the gate turns on when a sufficient charge has been accumulated, pulses do not need to be strictly timed, and in principle, it can work in a spiking mode. Spiking NW neurons, including emitters, have been developed [20], and spiking photonic neurons with picoseconds have been suggested [21].

For the present detector and analyzing systems, using III-V NWs appears to be an interesting solution, as they can act both as detectors of light and artificial neurons, including LIF units. Further, using light signals for network connections is highly relevant, as this

will give the fastest possible connectivity. The 450-550 nm bandwidth light emission of lead tungstate can be detected by the numerous III-V materials with bandgaps below 2.25 eV, including GaAs, InP, and various ternaries of these two compounds.

A layered system of arrays of NW components organized in a suitable geometry, where the bottom layer absorbs the incident scintillator light and the top layer serves as the output of the processed information, can be considered a trainable neuromorphic system. Referring to such a system in Figure 12a, we first turn to the light detectors. Here, four detectors in each pixel should be constructed that sequentially have an order of magnitude different sensitivity, as described in Equation (4). A suitable NW neuron architecture has one phototransistor for excitation coupled with an NW FET and NW LED emitter [17]. When sufficient light hits the phototransistor, the LED starts to emit. A simple way to achieve different thresholds on the four different phototransistors is to place scintillator-light absorbing GaInP layers of varying thickness between the scintillator and respective NW detectors. As 1 μm thick GaInP will be enough to completely absorb the scintillator light output and absorption decreases exponentially, varying thickness can be directly used to obtain the required orders of magnitude of difference. A representative design is shown in Figure 12b. Operating the NW LED emitter at lower energies (such as by using InP) can ensure that the signal output from the detector neuron will not be absorbed by the GaInP and reach the SNN. The operating timescales will be determined by the sensing and emission components of the neuron. The signal transmission travels at the speed of light and can be assumed to be instantaneous at these length scales (light travels 1 μm in ∼3 fs). Although the phototransistor risetime is ∼10 ps, the longer NW LED risetime of ∼100 ps sets the speed of the system. This response time is sufficient for the time window required in the present study.
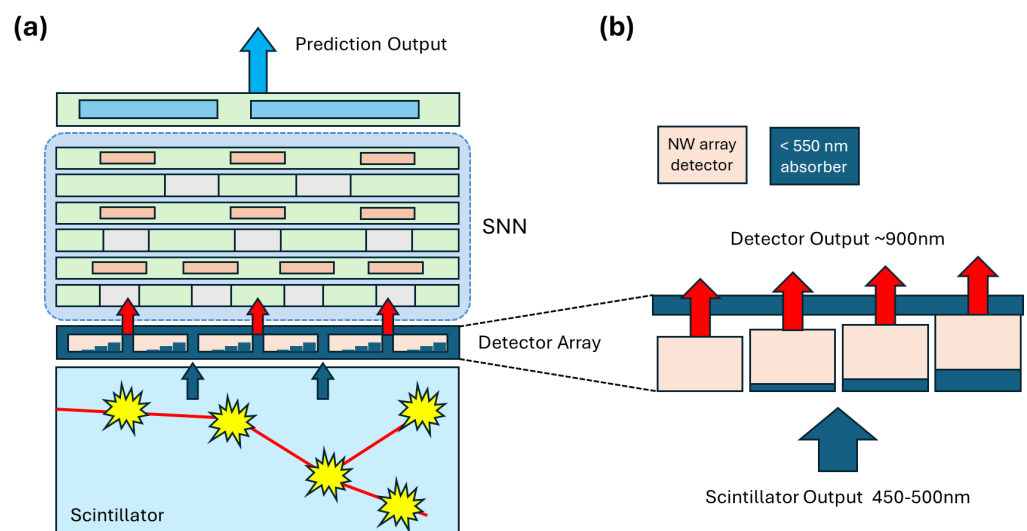

Figure 12. (**a**) Possible implementation of a readout system for a calorimeter block (bottom left)—light pulses are received by the layer of detector arrays and passed onto the SNN, which comprises multiple layers of metalens/waveguide broadcasting and LIF neurons. Finally, the prediction output from the SNN is decoded and read out. (**b**) Four-channel detector superpixel, with individual detectors receiving progressively attenuated signals due to their respective thicker absorbers.

The light signals from the four-detector superpixel can be guided further upward toward the SNN layer using waveguiding structures of a dielectric or plasmonic nature. For the next layer, an integrated LIF NW array neuron can be placed to act as the first layer in the SNN. Here, either a rate-dependent or spiking scheme can be used, as has been developed using various III-V materials [17,20]. To create the weight between different

layers of NW neurons, light signals should be manipulated to achieve varying intensity in different directions, using several wavelengths or polarization. NWs are sensitive to all of these, depending on the materials and geometric structure with sub-wavelength sensitivity possible [14,22]. Metalenses can also be used for inter-connectivity and to assign weights [23]. One can work with either for pre-training during the design phase and use fixed weights. Alternatively, absorbing molecules can be introduced that can be used as artificial synapses in the connections between the NWs [24].

7. Conclusions

While high-granularity calorimeters today appear to meet the dire necessity for cutting-edge future High Energy Physics (HEPs) applications and studies, at the high-energy and high-intensity frontiers, the fast readout and creation of suitable trigger primitives, as well as the size of the generated data volume and energy consumptions, pose problems for which the solution lays beyond the state of the art of the relative technologies. Processing the vast amounts of spatio-temporal data generated by light emission within sub-nanosecond time frames is challenging. Traditional machine learning approaches suffer from high latency and energy dissipation, and FPGA-based solutions are often limited by fixed logic constraints.

In this article, we propose a way forward by using neuromorphic computing for the extraction of fast topological primitives from the pattern of energy releases in a scintillating medium. The absence of incoming light amplification and the intrinsic capability of the considered system to work with photonic information processing make it a promising solution to the aforementioned bottlenecks.

Our studies indicate that an array of neurons exchanging spikes generated by photon intensities and arrival times on an array of input sensors can extract useful information on the total energy release, its centroid, and its dispersion, opening the way to the creation of high-level features that may extract (from the calorimeter block) the same amount of information that would be harvested by a highly granular instrument, thus bypassing the related readout and technology challenges.

The neuromorphic approach, particularly when implemented in nanophotonic hardware, is well-suited for this problem. By leveraging event-driven computation and sub-nanosecond neurosynaptic dynamics, neuromorphic systems can efficiently extract relevant features from the evolving light patterns in a homogeneous calorimeter without requiring high segmentation. The proposed III-V NW-based implementation further enhances performance by enabling ultra-fast and energy-efficient spike-based processing, paving the way for real-time, low-power event reconstruction in high-energy physics detectors.

Appendix A. Loss Profiles

Here, we report the loss profiles for the training of the various models. The training loss is computed for each batch (referred to as iteration in the plots), while the validation is computed after each full epoch.
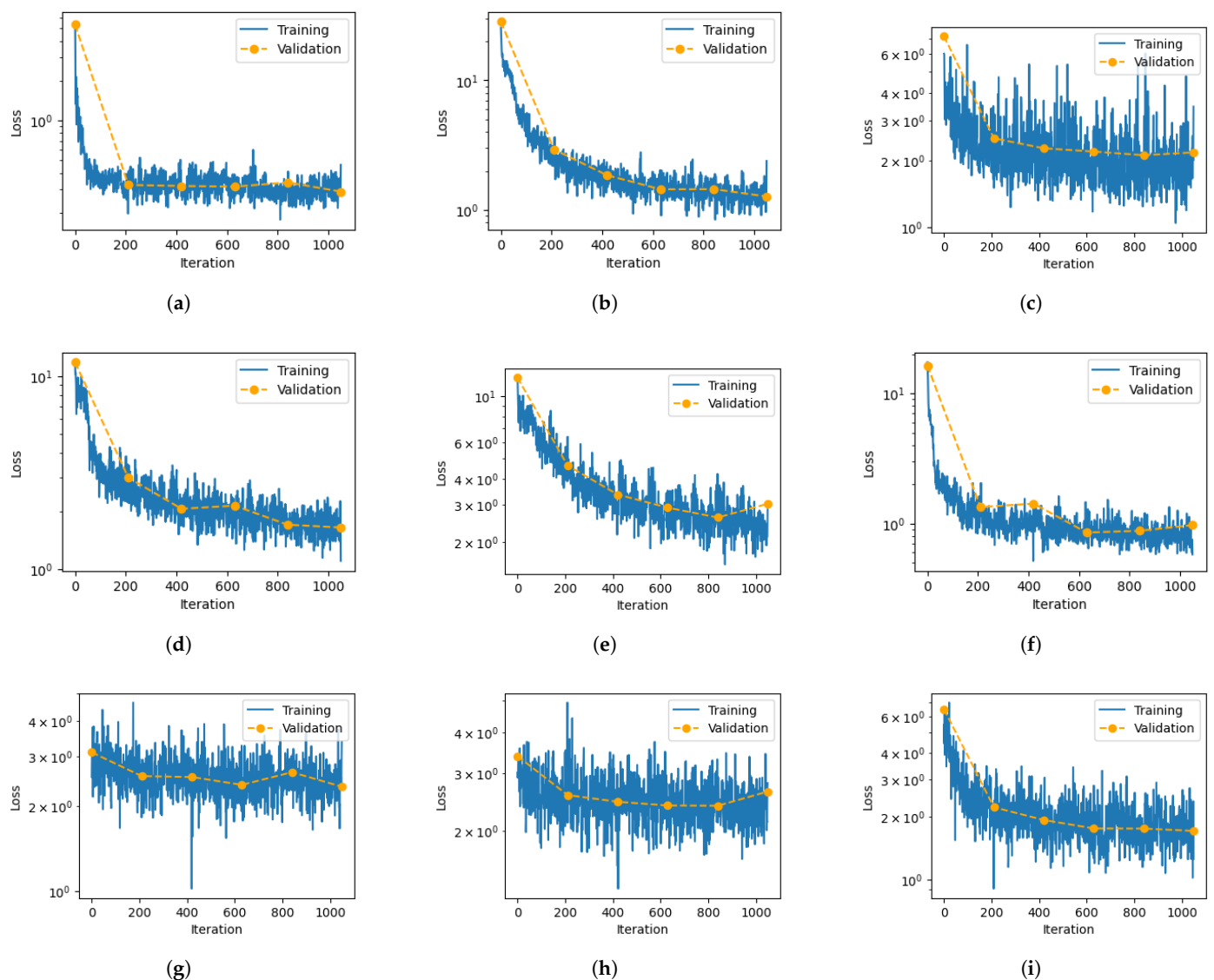

Figure A1. Loss profiles during training: (**a**) Single-target energy regression. (**b**) Multi-target energy and centroids regression. (**c**) Multi-target energy and dispersion regression. (**d**) Single-target x_c regression. (**e**) Single-target y_c regression. (**f**) Single-target z_c regression. (**g**) Single-target σ_x^2 regression. (**h**) Single-target σ_y^2 regression. (**i**) Single-target σ_z^2 regression.

References

1. Wing, M. Precise measurement of jet energies with the ZEUS detector. *Frascati Phys. Ser.* **2001**, *21*, 617–623.
2. The CMS Collaboration. Particle-flow reconstruction and global event description with the CMS detector. *J. Instrum.* **2017**, *12*, P10003. [CrossRef]
3. Kasieczka, G. Boosted Top Tagging Method Overview. *arXiv* **2018**, arXiv:1801.04180.
4. Kudithipudi, D.; Schuman, C.; Vineyard, C.M.; Pandit, T.; Merkel, C.; Kubendran, R.; Aimone, J.B.; Orchard, G.; Mayr, C.; Benosman, R.; et al. Neuromorphic computing at scale. *Nature* **2025**, *637*, 801. [CrossRef]
5. Mehonic, A.; Ielmini, D.; Roy, K.; Mutlu, O.; Kvatinsky, S.; Serrano-Gotarredona, T.; Linares-Barranco, B.; Spiga, S.; Savel'ev, S.; Balanov, A.G.; et al. Roadmap to neuromorphic computing with emerging technologies. *APL Mater.* **2024**, *12*, 109201,
6. Nilsson, M.; Schelén, O.; Lindgren, A.; Bodin, U.; Paniagua, C.; Delsing, J.; Sandin, F. Integration of neuromorphic AI in event-driven distributed digitized systems: Concepts and research directions. *Front. Neurosci.* **2023**, *17*, 1074439. [CrossRef]
7. Nilsson, M.; Pina, T.J.; Khacef, L.; Liwicki, F.; Chicca, E.; Sandin, F. A Comparison of Temporal Encoders for Neuromorphic Keyword Spotting with Few Neurons. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023; p. 1. [CrossRef]
8. Allison, J.; Amako, K.; Apostolakis, J.; Arce, P.; Asai, M.; Aso, T.; Bagli, E.; Bagulya, A.; Banerjee, S.; Barrand, G.; et al. Recent developments in Geant4. *Nucl. Instrum. Meth. A* **2016**, *835*, 186. [CrossRef]
9. De Vita, A.; Abhishek; Aehle, M.; Awais, M.; Breccia, A.; Carroccio, R.; Chen, L.; Dorigo, T.; Gauger, N.R.; Keidel, R.; et al. Hadron Identification Prospects with Granular Calorimeters. *arXiv* **2025**, arXiv:2502.10817.
10. Annenkov, A.; Korzhik, M.; Lecoq, P. Lead tungstate scintillation material. *Nucl. Instrum. Meth. A* **2002**, *490*, 30. [CrossRef]
11. Eshraghian, J.K.; Ward, M.; Neftci, E.O.; Wang, X.; Lenz, G.; Dwivedi, G. Training Spiking Neural Networks Using Lessons from Deep Learning. *Proc. IEEE* **2023**, *111*, 1016. [CrossRef]
12. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015. [CrossRef]
13. Javanshir, A.; Nguyen, T.T.; Mahmud, M.A.P.; Kouzani, A.Z. Advancements in Algorithms and Neuromorphic Hardware for Spiking Neural Networks. *Neural Comput.* **2022**, *34*, 1289. [CrossRef] [PubMed]
14. Barrigón, E.; Heurlin, M.; Bi, Z.; Monemar, B.; Samuelson, L. Synthesis and Applications of III–V Nanowires. *Chem. Rev.* **2019**, *119*, 9170–9220.
15. Winge, D.O.; Limpert, S.; Linke, H.; Borgström, M.T.; Webb, B.; Heinze, S.; Mikkelsen, A. Implementing an Insect Brain Computational Circuit Using III–V Nanowire Components in a Single Shared Waveguide Optical Network. *ACS Photonics* **2020**, *7*, 2787–2798. [CrossRef] [PubMed]
16. Wittenbecher, L.; Boström, E.V.; Vogelsang, J.; Lehman, S.; Dick, K.A.; Verdozzi, C.; Zigmantas, D.; Mikkelsen, A. Unraveling the Ultrafast Hot Electron Dynamics in Semiconductor Nanowires. *ACS Nano* **2021**, *15*, 1133–1144. [CrossRef] [PubMed]
17. Winge, D.; Borgström, M.; Lind, E.; Mikkelsen, A. Artificial nanophotonic neuron with internal memory for biologically inspired and reservoir network computing. *Neuromorphic Comput. Eng.* **2023**, *3*, 034011. [CrossRef]
18. Flodgren, V.; Das, A.; Sestoft, J.E.; Alcer, D.; Jensen, T.K.; Jeddi, H.; Pettersson, H.; Nygård, J.; Borgström, M.T.; Linke, H.; et al. Direct on-Chip Optical Communication between Nano Optoelectronic Devices. *ACS Photonics* **2025**, *12*, 655–665. [CrossRef]
19. Alcer, D.; Hrachowina, L.; Hessman, D.; Borgström, M. Processing and characterization of large area InP nanowire photovoltaic devices. *Nanotechnology* **2023**, *34*, 295402. [CrossRef]
20. Romeira, B.; Javaloyes, J.; Ironside, C.N.; Figueiredo, J.M.L.; Balle, S.; Piro, O. Excitability and optical pulse generation in semiconductor lasers driven by resonant tunneling diode photo-detectors. *Opt. Express* **2013**, *21*, 20931–20940. [CrossRef]
21. Shastri, B.J.; Nahmias, M.A.; Tait, A.N.; Wu, B.; Prucnal, P.R. SIMPEL: Circuit model for photonic spike processing laser neurons. *Opt. Express* **2015**, *23*, 8029–8044. [CrossRef]
22. Mårsell, E.; Boström, E.; Harth, A.; Losquin, A.; Guo, C.; Cheng, Yu.; Lorek, E.; Lehmann, S.; Nylund, G.; Stankovski, M.; et al. Spatial Control of Multiphoton Electron Excitations in InAs Nanowires by Varying Crystal Phase and Light Polarization. *Nano Lett.* **2018**, *18*, 907–915. [CrossRef]
23. Meng, Y.; Chen, Y.; Lu, L.; Ding, Y.; Cusano, A.; Fan, J.A.; Hu, Q.; Wang, K.; Xie, Z.; Liu, Z.; et al. Optical meta-waveguides for integrated photonics and beyond. *Light. Sci. Appl.* **2021**, *10*, 235. [CrossRef] [PubMed]
24. Alcer, D.; Zaiats, N.; Jensen, T.K.; Philip, A.M.; Gkanias, E.; Ceberg, N.; Das, A.; Flodgren, V.; Heinze, S.; Borgström, M.T.; et al. Integrating molecular photoswitch memory with nanoscale optoelectronics for neuromorphic computing. *Commun. Mater.* **2025**, *6*, 11. [CrossRef]