Deep Learning Based Energy Spectrum Estimation for High Counting Rate Nuclear Spectrometry

Yiwei Huang, Congyu Lin, Dima Bykhovsky, Tom Trigano, Zikang Chen, Xiaoying Zheng, and Yongxin Zhu

***Abstract*—High counting rates pose a challenging problem for nuclear spectrometric systems, where a physical phenomenon known as the pile-up effect distorts direct measurements, resulting in a significant bias in spectrum estimation. In this paper, we propose an innovative neural network that combines the self-attention mechanism and convolutional neural network (CNN) architectures to address the problem of spectrum estimation. The approach was evaluated on spectral data from a small scintillator (NaI) and simulated signals from a dedicated simulator and compared with conventional methods; furthermore, our method was compared to the state-of-the-art (SOTA) time domain method on the Allpix2 simulator and was evaluated on a real-world dataset. The results demonstrate that the proposed method leads to a more accurate inference of the energy spectrum, even at high count rates. Experiments also show that the proposed method is robust with varying sources, different data scenarios, and noise intensities.**

***Index Terms*—nuclear spectroscopy, pile-up correction, convolutional neural network, self-attention, sequence segmentation**

I. Introduction

Gamma-ray spectroscopy aims to analyze and use radioactive sources to probe material properties. It is well known that the *pile-up* effect is a limiting factor in performing gamma-ray spectrometry at high count rates [1]. The phenomenon of pile-ups stems from the fact that the inter-arrival times between photons can be shorter than the durations of generated electrical pulses after amplification, thus creating clusters of pulses (i.e., pile-ups). At high count rates, pile-ups during the detection lead to energy peak drift, energy resolution deterioration, and spectral distortion.

There has been extensive research on pile-up correction. The most common approach is rejection. More specifically, we detect when pile-up occurs and collectively reject all the pulses that took part in that incident [2]. Such an approach can increase the duration of experiments, and some methods extend this approach further by compensating for the pulses that were rejected with the motivation to align the actual activity of the recorded signal with the activity of the processed, pile-up effect clean signal. Other pileup compensation strategies include pulse clipping [3], template fitting [4] [5], baseline subtraction [6], adaptive filtering [7], and time-varying trapezoidal filter algorithms [8]. These algorithms are typically most effective in systems with low pile-up probabilities and minimal pulse shape variations. More recently, it was suggested to consider the output signal as a linear combination of known shapes and to relate the pile-up effect to a sparse regression problem [9] [10]. Though efficient, these approaches are limited when the activity of the radioactive source increases too much. At a more general level, the importance of finding other methods goes beyond signal processing of the signal in the time domain was also noted in [11].

Thus, the recent effort seeks to analyze pulse collections statistically rather than to recognize individual pulses in the time domain. By formulating a nonlinear inversion problem, the authors in [12] developed a nonparametric pile-up correction method that derives an analytical relation between the probability density function (PDF) of the observed pile-ups and the PDF of the individual pulses. Subsequently, in [13], the authors proposed a fast algorithm for pile-up correction based on this inversion formula. In [14], the authors reformulated the pile-up problem as a decompounding problem, constructing a kernel-based estimator to model pulse pile-up. Additionally, a novel method is outlined in [15], where the characteristic function is used to infer the distribution of gamma photon energies from indirect measurements. In this method, the pulse shape and characteristic function must be known in advance. However, this kind of statistical approach requires long-term observations to accumulate data, resulting in delays in real-time processing.

In the latest decade, deep learning (DL) has contributed significantly to the advancement of many research areas, including gamma spectroscopy. For example, [16] developed a convolutional neural network (CNN) that attempts to separate and estimate the true pulse height of pile-up events from high count rate radiation. A DL-based technique for the detection and classification of the pulses under high count rates was proposed in [17]. In [18], a CNN-based automated isotope identification method was proposed. In [19], another DL-based activity estimation was proposed, and a DL-based spectrum correction algorithm was developed in [20]. Most of these DL-based methods tend to label only a single pulse when making training data, discarding the accumulated signals, and following a pile-up rejection paradigm, which can drastically increase the duration of the performed experiments at high counting rates.

Traditional methods, which focus on identifying and rejecting pile-ups or employing statistical corrections, often become

This research was supported by the National Natural Science Foundation of China under grant numbers 12373113 and 12475196. *(Corresponding authors: Dima Bykhovsky; Xiaoying Zheng)*

Yiwei Huang, Congyu Lin, Zikang Chen, Xiaoying Zheng, and Yongxin Zhu are with the University of Chinese Academy of Sciences and Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China (e-mail: huangyw@sari.ac.cn; lincy@sari.ac.cn; chenzk@sari.ac.cn; zhengxy@sari.ac.cn; zhuyongxin@sari.ac.cn).

Dima Bykhovsky and Tom Trigano are with the Department of Electrical and Electronics Engineering, Sami Shamoon College of Engineering, Beer-Sheva and Ashdod campuses, 8410802, Israel (e-mail: dmitrby@ac.sce.ac.il; thomast@ac.sce.ac.il).

This article has been accepted for publication in IEEE Transactions on Instrumentation and Measurement. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIM.2025.3573370

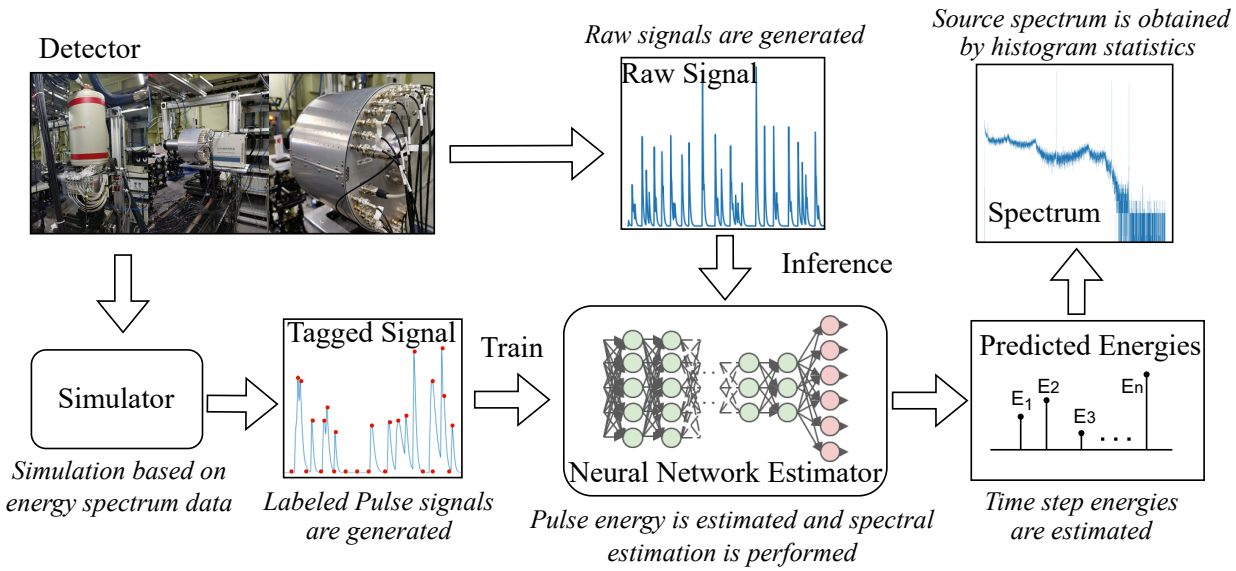


Fig. 1: Overview of the proposed DL-based spectrum estimation framework. The neural network is trained on a dataset obtained from a gamma-ray simulator. In practice, the signals from the gamma-detector are fed into the neural network to infer the pulse energy and produce an energy histogram as output.

ineffective or impractical at high counting rates due to their reliance on precise event separation and prolonged observation periods. Recent developments in deep learning (DL) might have the potential to address these limitations. This paper introduces a novel DL approach for energy spectrum estimation in gamma-ray spectrometry, which can potentially handle high counting rates where traditional methods struggle due to the pile-up effect. In contrast to other existing techniques, the input relies directly on the recorded time signal rather than the processed energy spectra, as shown in Fig. 1. In particular, we develop an *AttnUNet++* architecture combining the self-attention mechanism with UNet to address high count rate conditions with extreme pile-ups where conventional methods fail. The training dataset of signals is generated by a SOTA simulator for gamma spectroscopy [21], [22]. The reason we train a neural network using simulated data instead of the real output from a detector is that we cannot accurately label the pile-ups from a detector at very high count rates, and that up to our knowledge, no annotated dataset of the recorded time signals exists in this framework. On the other hand, all the individual electrical pulses can be accurately labeled on simulated data. It is quite common to use simulated data to train a neural network [16]. Our experiments show that under the conditions of high count rates and the failure of the conventional methods, the designed *AttnUNet++* is able to infer pulse energies and achieve accurate spectrum estimation.

The paper highlights are as follows:

- This work concentrates on time-domain signal analysis rather than the mainstream energy histogram-based analysis. The advantage of our method is that it can be directly applied to raw detector outputs rather than relying on preprocessed energy spectra.
- A novel deep-learning approach named *AttnUNet++* is introduced, which combines CNNs with self-attention mechanisms, effectively capturing both local pulse shapes and global signal dependencies, and addressing the complexities introduced by extreme pile-ups.
- Recognizing the difficulty of labeling pile-up events in real detector data, the authors generate precise training data using a SOTA open-source gamma-ray simulator, ensuring accurate labels for effective network training under realistic conditions.
- The proposed method demonstrates robustness and superior performance on various data sources, including Allpix2 simulation data and real experimental data, showing generalization ability. Our method consistently outperforms conventional algorithms, especially at high counting rates where pile-up is severe, and surpasses the SOTA method.
- The developed DL-based method significantly reduces the observation duration needed for accurate spectrum estimation, providing real-time inference capabilities beneficial for practical applications in nuclear spectroscopy.

II. Model and Data Generation

In this section, we first formally formulate the problem of estimating the energy spectrum from detectors' signals. We then describe the principle of pile-ups from the perspective of a stochastic process. Finally, we briefly introduce the simulator used to prepare training data. Table I summarizes all notations used throughout this paper.

A. Spectrum Estimation Model

We model the interaction between the gamma radiation and the detector as a sample path of a temporal point process (TPP), which is a probability distribution over variable-length sequences in some time interval: $\mathcal{H}_t = \{(t_i) : t_i < t\}$ [23].

TABLE I: Notations.

Variable name	Definition	Assumptions
$\mathcal{L}_s$	Length of time to simulate sampling (s)	-
f_s	Analog sampling rate	-
H	Energy histogram of the source	-
S	Signal generated by the simulator	-
N	Number of events	$N(t)$ is the number of events during the time t
λ_n	Intensity of Poisson process	Number of photons per second (cps) is given by $\lambda_n \times f_s$
$\mathcal{N}$	Magnitude of noise in the given unit	-
$\mathbf{A}$	Shape dictionary simulates the jitter of the signal	Dictionary size $= 20$ of double exponential shape
seed	Simulated random number seed	Randomly selected
$\mathcal{B}$	Channel number of a source	$\mathcal{B} \leq 8192$
$\mathcal{T}$	Shape length	
DC	Multiplication of arrival count rate by pulse-shape length, $DC = \lambda\mathcal{T}$	
σ	Standard deviation of Gaussian noise	
$\mathcal{P}$	Pile-up probability	$\mathcal{P} \leq 1$

Since the arrival time can be modeled as a Poisson process [24], the event times of a TPP can be described using the conditional intensity function $\lambda(t)$:

$$\lambda(t \mid \mathcal{H}_t) = \lim_{\Delta t \to 0} \frac{\mathbb{P}(N(t+\Delta t) - N(t) = 1 \mid \mathcal{H}_t)}{\Delta t}, \quad (1)$$

where the intensity $\lambda(t)$ may depend on the history $\mathcal{H}_t$, representing the activity of the source. Based on signal theory (see section II-B), each event is characterized by its arrival time and energy, which can be modeled as a marked TTP (MTPP) [25]. Here, the energies of the events serve as categorical marks of MTPP from a finite set of labels: $\mathrm{m} = \{k_1, k_2, \ldots, k|\mathrm{m}|\}$. Thus, the joint conditional intensity function is $\lambda(t, m \mid \mathcal{H}_t) = \lambda(t \mid \mathcal{H}_t) \cdot p(m \mid t, \mathcal{H}_t)$, where $p(m \mid t, \mathcal{H}_t)$ is the conditional distribution of the label m given the time t and event history $\mathcal{H}_t$.

In general, our goal is to extend a TPP $\mathcal{H}_t = \{(t_i) : t_i < t\}$ to an MTPP $\mathcal{H}_m = \{(t_i, m_i) : t_i < t\}$. From a DL perspective, given a model with parameters θ, the objective is to estimate the label for each point in a time series. We aim to find the optimal parameters θ that minimize the expected loss (e.g., cross-entropy) across all points in the series. Formally, we defined the problem of spectrum estimation as:

$$\min_{\theta} \mathbb{E}_{(X,Y)\sim p_{\text{data}}} \left[\sum_{i=1}^{N} L(\hat{m}_i, m_i) \right], \quad (2)$$

where X, Y represents the signal and the estimateed label of the dataset, L is the loss function, $\hat{m}_i, m_i$ are the estimateed label and the ground truth per event. Overall, we formulate the problem of spectrum estimation as learning an optimal model to estimate the energy mark for each photon event based on the event's arrival time and its corresponding historical data.

B. Model of the Observed Signal

Theoretically, the detector output can be modeled as a linear combination of pulse shapes. The output signal from the detector can be modeled as:

$$S(n) \triangleq \sum_{k \geq 1}^{N} \mathcal{E}_k \Phi_k (t - T_k) + \varepsilon, \quad (3)$$

where $\{T_k, 1 \leq k \leq N\}$ is the sample path of a homogeneous Poisson process with constant unknown intensity λ_n, $\{\mathcal{E}_k, 1 \leq k \leq N\}$ represents each photon's energy, which is drawn randomly from an independent and identically distributed (i.i.d.) variable population, where the energy distribution is generated based on known spectrum data $H(\cdot)$, and ε denotes the additive noise. The electrical pulse shape $\{\Phi_k, 1 \leq k \leq N\}$, is randomly chosen from a dictionary $\mathbf{A}$, as shown in Fig. 3. One of the more common models is the double exponential [26], which can be modeled as:

$$h(t) = a \left(\exp\left(-\frac{t}{\tau_2}\right) - \exp\left(-\frac{t}{\tau_1}\right) \right), \quad (4)$$

where a is the normalization factor and $\tau_1 \ll \tau_2$ are characteristic decay times. For practical purposes, any pulse shape is assumed to have a finite time length $\mathcal{T}$.

A key characteristic of the pulse shape is the peak time, t_r, also known as the rise time, which is complementary to the fall time $t_f = \mathcal{T} - t_p$. For instance, the rise time for the double exponential model is given by:

$$t_r = \frac{\tau_1 \tau_2}{\tau_2 - \tau_1} \ln \frac{\tau_2}{\tau_1}, \quad (5)$$

where $\tau_1 \ll \tau_2$ are the decay times.

Source: The signal simulator is capable of replicating the behavior of a single radioactive source, as well as modeling the interactions among a mixture of multiple elements:

$$P(E = e) = \sum_{i}^{n} \frac{w_i}{\sum_i w_i} P_{e_i}(E = e), \quad (6)$$

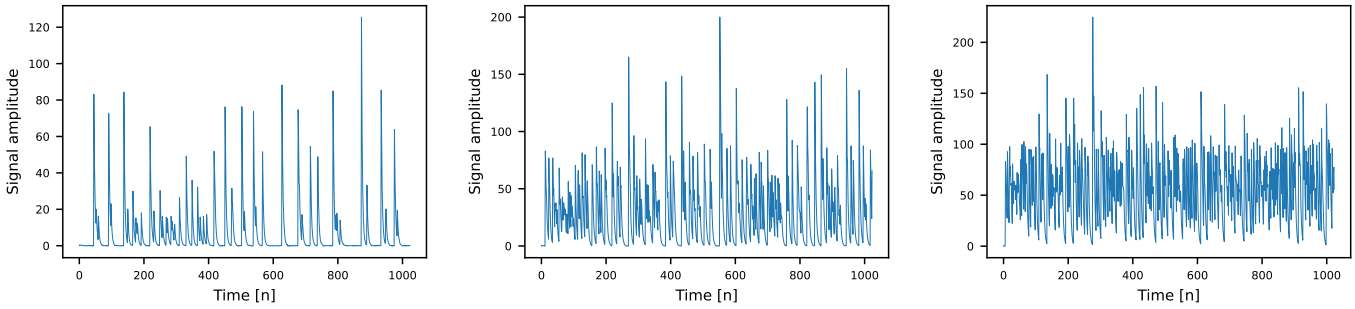
where P_{e_i} represents the probability that e_i emits energy e, and E is the signal source's spectrum. Thus, the spectrum of the mixture sources can be modeled as the following:

$$H_{mix} = \sum_{i}^{n} \frac{w_i}{\sum_i w_i} E_i. \quad (7)$$

Hence, the probability density function of the mixture histogram can be derived.

Energy-Probability: The common representation of the arrival energies is by a histogram that represents the number of signal events $b_i = N(t)/t$ per energy bin H_i (typically

This article has been accepted for publication in IEEE Transactions on Instrumentation and Measurement. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIM.2025.3573370



(a) $\lambda_n = 0.05, DC = 0.99, \mathcal{P} = 0.628$ (b) $\lambda_n = 0.2, DC = 3.96, \mathcal{P} = 0.981$ (c) $\lambda_n = 0.4, DC = 7.92, \mathcal{P} = 0.999$

Fig. 2: Example of training and testing data, generated from 33.3% ^{197m}Hg and 66.6% ^{169}Er with different random seeds. (a) $\lambda_n = 0.05$; (b) $\lambda_n = 0.2$; (c) $\lambda_n = 0.4$, representing low, high and very high source activity, respectively, with correspondingly increasing pileup probabilities.

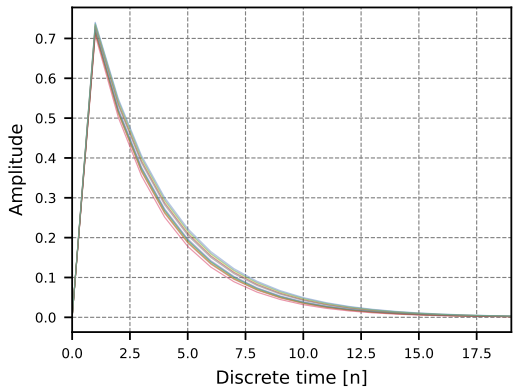


Fig. 3: Pulse shapes, which are simulated by a double exponential function dictionary; there are 20 in total, and the area under each shape is normalized to 1.

measured in keV). This energy histogram can be easily modeled by a discrete random distribution where the probability of energy is proportional to b_i:

$$\Pr(H = H_i) = \frac{b_i}{\sum_i b_i}. \tag{8}$$

The pairs of values for H_i and b_i can be determined by the radioactive energy of the mixture using the Monte Carlo method.

Arrival time: Based on the physical properties of gamma radiation, the arrival of gamma particles is modeled using a Poisson counting process with a rate of $\lambda(t)$ occurrences per unit time. For simulating arrival times, the inter-arrival times follow an exponential distribution with a mean of $1/\lambda(t)$.

Pile-up events occur when multiple pulses appear within a single trace. The duty cycle (DC) value is used as a convenient parameter to assess the probability of pile-up events. The DC is calculated by multiplying the intensity of the Poisson process by the pulse shape length:

$$DC = \lambda_n \mathcal{T}. \tag{9}$$

Based on the probability relation:

$$\Pr(N(t) = n) = \frac{(\lambda t)^n}{n!} \exp -(\lambda t), \tag{10}$$

the resulting pile-up probability is given by:

$$\mathcal{P} = \Pr(N(\mathcal{T}) > 0) = 1 - \Pr(N(\mathcal{T}) = 0) = 1 - \exp(-DC). \tag{11}$$

This study considers DC values above 0.1 to have a high pile-up probability.

C. Data Generation

Since, to the best of our knowledge, there is no annotated dataset in the time domain, for any supervised learning technique, the use of physics-reliable simulated data is mandatory to provide supervised learning with sufficient annotated events; the simulator is used to tag the signal, as can be seen in Fig. 4.

Under $\mathcal{L}_s$ seconds of signal simulation, the simulator produces time series S, energy vector $\mathcal{E}_k$, and arrival time vector T_k, and with the sampling rate f_s, the simulation contains $\mathcal{L}_s \cdot f_s$ discrete points. Thus, the dataset can be formulated by cutting S into multiple samples. Each sample is a one-dimensional time series signal with a length of L, which represents a period of pulses generated from the detector: $X_i = \{x_1, x_2, x_3, ..., x_l\}$, while the label is a vector of the same length that represents the energy of each arrival photon, as well as the idle time step:

$$Y_i = \{y_1, y_2, y_3, ..., y_L\}, y_l = \begin{cases} 0, & else \\ \mathcal{E}_k, & l = T_k \end{cases}, k \in [1, N]. \tag{12}$$

Then, the dataset is given by $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{\frac{\mathcal{L}_s \cdot f_s}{L} \times L}$, where each pulse sequence will serve as a neural network input and is labeled with an equal-length energy sequence. However, as the activity increases, multiple events may occur within a single time step. In this case, we split these events and distribute their energies over the next several time steps; this is because we do not consider multi-label classification. Training data under different counting rates are represented in Fig. 2.

This article has been accepted for publication in IEEE Transactions on Instrumentation and Measurement. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIM.2025.3573370

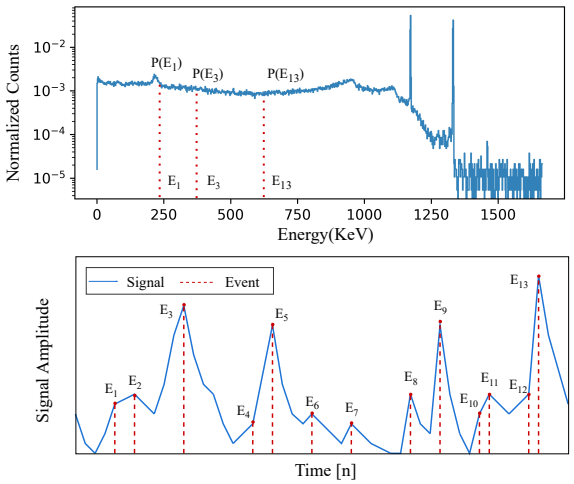


Fig. 4: The pulse data output from the simulator along with the corresponding energy tagging. Time steps without a red line indicate that no events have occurred, and their energy values are zero, which are omitted for clarity.

III. Neural Network Architecture for Energy Spectrum Estimation from Time Signals

This section describes the proposed deep neural network architecture tailored for the high precision demand in spectrum estimation. A classification model is used for energy estimation. A multi-head self-attention mechanism (MSA) enhances the classification method by capturing relevant spatial information in signal sequences.

A. Overview

We first provide some motivation and an overall description of the approach described later on. Based on the essence of pile-ups from the perspective of a stochastic process and the further signal model we established, the spectrum estimation problem can be seen as a one-dimensional signal segmentation task, which closely resembles a pixel-wise classification problem in medical image segmentation. To address our approach, each time step in the sequence is assigned a class label, and the designed DL model AttnUNet++ is trained to learn the underlying structures in a data-driven manner.

B. UNet++

Existing schemes often utilize regression methods for energy estimation. In common applications such as stock price prediction and sensor value forecasting, there are inevitably slight deviations between the predicted value and the true value obtained by the regression model. Still, in general, the prediction deviation is relatively small and the effect is satisfactory. However, when applied to spectrum estimation, even if the curve calculated by the regression model is very close to the real curve, the estimated energy will still deviate slightly from the real energy when mapped to the energy. These accumulated deviations can lead to poor energy resolution.

Therefore, we propose to use a multi-classification network. In gamma spectrometry, the spectrum is sliced into multiple channels or bins [1] evenly (for instance, 256, 512, 1024 bins for NaI detectors). When processing signals, the multi-channel analyzer (MCA) sorts the on-the-fly pulses into bins according to the pulses' energy. Intuitively, each bin in the gamma spectrum is defined as a class in the DL-based approach.

As discussed before, the training dataset is prepared by simulated data. All pulses that include pile-ups usually discarded by conventional MCA methods can be sorted into correct classes since the ground truth of the signal processes is known. Thus, the training dataset is tagged with precise labels. A one-dimensional UNet architecture is proposed to identify each pulse/pileup from the sequence of signals. Once a pulse/pile-up is identified, the multi-class tag of each individual pulse/pile-up can be determined. It is obvious that if the classifier is able to sort the pile-ups to the correct classes, the network must output an accurate energy estimate, thus ensuring the need for high energy resolution.

Fig. 5 presents the architecture of *AttnUNet++*. The UNet-like structure consists of an encoder (left side), a decoder (right side), and a bottleneck layer [27]. The encoder iteratively reduces the data length and expands feature dimensions to extract input features. At the same time, the decoder receives skip connections from the encoder and expands the tensor to generate the output. Specifically, the input to the network is represented as $\mathbf{X} \in \mathbb{R}^{\mathcal{B}\times\mathcal{L}\times C}$, where $\mathcal{B}$, $\mathcal{L}$, $\mathcal{C}$ represent the batch size, sequence length, and channel number, respectively. For each signal instance $\mathbf{x} \in \mathbb{R}^{\mathcal{L}\times C}$ of the input matrix $\mathbf{X}$, $\mathbf{x}$ contains n events, and the objective of the end-to-end deep neural network is to estimate the energy pulse event vector: $\mathbf{y} \in \mathbb{R}^{\mathcal{L}\times 1}$, where the output layer of the network maps the learned feature representations to probability matrices of all classes using *Softmax* activation: $p_k(\mathbf{x}) = \exp\left(a_k(\mathbf{x})\right) / \left(\sum_{i=1}^{C} \exp\left(a_i(\mathbf{x})\right)\right)$, where a_k is the linear projection of the last layer of the backbone network to the k-th class of the sequence. To promote the learning of semantic information at various scales, the skip pathways have been redesigned to alter the connectivity between the encoder and decoder sub-networks. In the original U-Net architecture, the decoder directly receives feature maps from the encoder. However, in UNet++ [28], feature maps undergo processing via a dense convolution block. The pyramid level determines the number of convolution layers, resulting in a complex topology characterized by densely connected skip connections between decoders. These connections facilitate dense feature propagation, allowing for flexible feature fusion at decoder nodes, where each node integrates multiscale features both horizontally and vertically. The redesigned skip pathway can be formulated as follows:

$$x^{i,j} = \begin{cases} \mathcal{H}\left(x^{i-1,j}\right), & j = 0 \\ \mathcal{H}\left(\mathcal{C}\left(x^{i},k\right)_{k=j} \oplus \mathcal{U}\left(x^{i+1,j-1}\right)\right), & 0 < j \leq \mathcal{D}, \end{cases} \tag{13}$$

where $\mathcal{D}$ represents the model depth, $\oplus$ denotes the concatenation operation, $\mathcal{C}(\cdot)$ is the continuous concatenate operation defined as: $\mathcal{C}(x,k) = x^{i,k} \oplus x^{i,k-1} \ldots x^{i,0}$, function $\mathcal{H}(\cdot)$ is a

[1]We prefer to use the term bin to avoid confusion with the concept of channel in neural networks.

This article has been accepted for publication in IEEE Transactions on Instrumentation and Measurement. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIM.2025.3573370

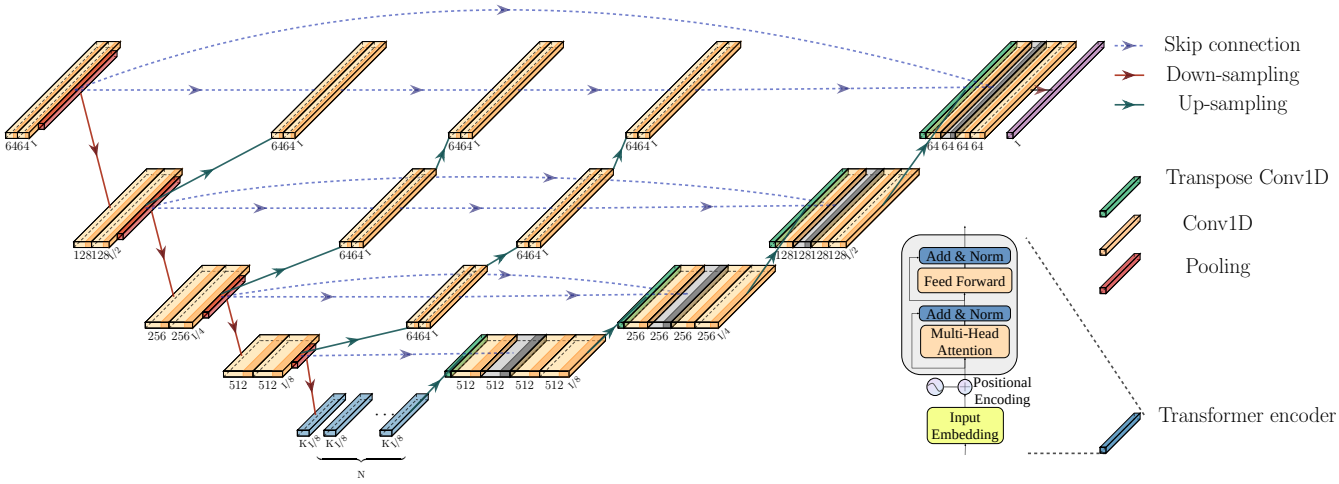


Fig. 5: Detailed architecture of *AttnUNet++*. Orange boxes denote one-dimensional convolutional (Conv1D) blocks, red downward arrows indicate down-sampling operations, green upward arrows represent up-sampling operations, dashed lines depict skip connections, and blue blocks are the transformer encoder module containing the MSA block.

convolution operation followed by an activation function, and $\mathcal{U}(\cdot)$ denotes an up-sampling layer. The deepest layer of the decoder receives a skip connection from the encoder, and as the layers ascend, the decoder incorporates more connections, enabling the fusion of multi-scale information. This multi-scale feature aggregation in UNet++ systematically refines segmentation outputs, both enhancing accuracy and increasing training convergence speed.

C. Attention Block

Rather than using a pure CNN architecture, we implemented a multi-head self-attention mechanism [29]. The motivation is to capture relevant spatial information in high-level features. MSA blocks are used to enhance the model's robustness and global information capture. This enriches feature representation and improves performance, which ensures that the model excels in both local feature extraction and global context understanding, making it highly effective for complex segmentation tasks. Specifically, the bottleneck is replaced with multiple MSA blocks to enhance the model's understanding of input features and the model's stability. The MSA block is given by:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{14}$$

where Q, K, V are the query, key, and value matrices. d_k is the hidden size of the query or key matrices.

The input tensor of the MSA block is $\mathbf{X} \in \mathbb{R}^{\mathcal{B}\times\mathcal{L}\times C}$, and $Q = QW_i^Q, K = KW_i^K, V = VW_i^V$, where $W_i^Q \in \mathbb{R}^{d\times d_q}, W_i^K \in \mathbb{R}^{d\times d_k}, W_i^V \in \mathbb{R}^{d\times d_v}$ are the linear projection matrices, and $d = C/h$, where h represents the head number. After each head's self-attention operation, the heads are concatenated and connected to a linear projection layer W: $(heads_1, heads_2, \ldots, heads_n)\, W$.

Positional encoding allows the model to carry relative or absolute position information since the attention operation is not causal:

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{\text{model}}}\right), \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{\text{model}}}\right). \end{aligned} \tag{15}$$

By stacking multiple self-attention layers, the model can leverage deeper feature representations and each layer of the MSA block is given by:

$$\begin{aligned} \mathbf{x}_0 &= \text{PE}\left(\mathbf{X}_p\right), && \ell = 1 \\ \mathbf{x}'_\ell &= \text{LN}\left(\text{MSA}\left(\mathbf{x}_{\ell-1}\right) + \mathbf{x}_{\ell-1}\right), \\ \mathbf{x}_\ell &= \text{LN}\left(\text{FFN}\left(\mathbf{x}'_\ell\right) + \mathbf{x}'_\ell\right) && 1 < \ell < \text{L}, \end{aligned} \tag{16}$$

where $\text{LN}(\cdot)$ is the layer normalization operation, L is the MSA block number, and $\text{FFN}(\cdot)$ is a single fully connected layer with *elu* activation.

D. Loss Function

The cross-entropy (CE) loss function quantifies the discrepancy between the model's estimated results and the actual labels. By minimizing CE through backpropagation, the model can learn the intricate nonlinear relationships between the data and the labels. The CE can be written as:

$$L(\mathcal{Y}, \hat{\mathcal{Y}}) = -\frac{1}{C}\sum_{i=1}^{L}\sum_{j=1}^{C}\mathcal{Y}_{i,j}\log(\hat{\mathcal{Y}}_{i,j}), \tag{17}$$

where C is the class number, in this case $C = \mathcal{B} + 1$, and $\mathcal{Y}, \hat{\mathcal{Y}}$ represent the labels Y_i after one-hot coded and estimation respectively.

IV. Evaluation

In this section, we evaluate the performance of the proposed solution. We conduct experimental evaluations from two different perspectives, including the application of common deep

TABLE II: Experimental settings.

Hyperparameters	Value
Model depth	4
Hidden size	128
Kernel size	3
Output nums	1025
Optimizer	Adam
Learning rate	0.001
Batch size	16
Num heads	8
Attention layers	6
Batch size	16
Epochs	100
Dict shape factor	τ_1: $\mu = 3 * 10^{-7}, \sigma = 10^{-8}$ τ_2: $\mu = 10^{-8}, \sigma = 10^{-10}$
f_s	10^7
SNR	70
Dict size	20
Train sample	7000
Test sample	3000
Dict type	double exponential

learning performance metrics to evaluate the classification performance of *AttnUNet++* and the application of conventional gamma spectroscopy metrics to evaluate the results of spectrum estimation.

A. Experimental Setup

As shown in Fig. 1, the workflow of experiments is as follows: first, the model is trained using the input data $\mathbf{X}_{train}, \mathbf{Y} \in \mathbb{R}^{\frac{\mathcal{L}_s \cdot f_s}{L} \times L}$; next, the trained weights are applied to the test data $\mathbf{X}_{test}$ to generate estimation $\hat{\mathbf{Y}}$; finally, the estimated results are aggregated into a histogram to produce the final spectrum estimation, $\hat{H}$.

In preparing the dataset by the simulator, the sampling rate is set to 10 MHz, with a rise time of 3.29×10^{-8}s and a fall time of 1.94×10^{-6}s, resulted in each pulse shape having a length of approximately 1.98 μs (approximately 20 samples). The Poisson process intensity λ_n varies from 0.05 to 0.4, and the corresponding counting rates range from 500,000 cps to 4,000,000 cps. The energy spectrum is split into 1024 bins, which leads to 1025 classes in the *AttnUNet++* model with an additional default class of negative samples.

All experiments were run on a single AMD Instinct MI210 GPU with 64GB memory under the TensorFlow framework. Table II summarizes the experimental settings, including hyper-parameters.

In experiments, various energy spectra are evaluated. Specifically, ^{60}Co is used to assess classification results through metrics such as the confusion matrix, precision-recall (PR) curve, receiver operating characteristic (ROC) curve, and F_1 score, and examine the training accuracy of the model under different signal-to-noise ratios (SNRs). Other energy spectra are also evaluated to verify various parameters of energy spectrum estimation.

B. Evaluation Metrics

Precision, Recall, and AP: the precision, recall and average precision (AP) are the metrics widely used in evaluating deep learning models and are defined as follows:

$$\begin{aligned} \text{Precision} &= \frac{\sum_c TP_c}{\sum_c (TP_c + FP_c)}, \\ \text{Recall} &= \frac{\sum_c TP_c}{\sum_c (TP_c + FN_c)}, \\ \text{AP} &= \sum_n (R_n - R_{n-1}) P_n, \end{aligned} \tag{18}$$

where c is the class index, n represents the number of thresholds, and TP, FP, and FN denote true positives, false positives and false negatives, respectively.

ROC: The ROC curve measures the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR). The TPR and FPR are calculated as:

$$\begin{aligned} TPR &= \frac{\sum_c TP_c}{\sum_c (TP_c + FN_c)}, \\ FPR &= \frac{\sum_c FP_c}{\sum_c (FP_c + TN_c)}, \end{aligned} \tag{19}$$

where TN represents true negatives.

Macro average $\mathbf{F_1}$: In highly imbalanced multi-class classification scenarios, micro averages are often preferred over macro averages, as they provide a balanced overall performance metric. However, to comprehensively assess the model's performance, we also employed the macro average F_1-score in addition to PR and ROC metrics. The F_1-score is the harmonic mean of precision and recall, effectively balancing these two metrics. The macro-average F_1-score is defined as:

$$\text{Macro-average } F_1 = \frac{1}{C} \sum_c \frac{2TP_c}{2TP_c + FP_c + FN_c}, \tag{20}$$

where C is the total number of classes. Using the F_1-score allows us to evaluate how well the model performs on each class individually, irrespective of the class distribution, thereby offering a more comprehensive understanding of the model's capabilities and limitations. Accuracy is defined as the percentage of the model's correct predictions out of all samples. It is used to evaluate the overall predictive accuracy of the model. Some statistical indicators are also defined to evaluate the spectral estimation.

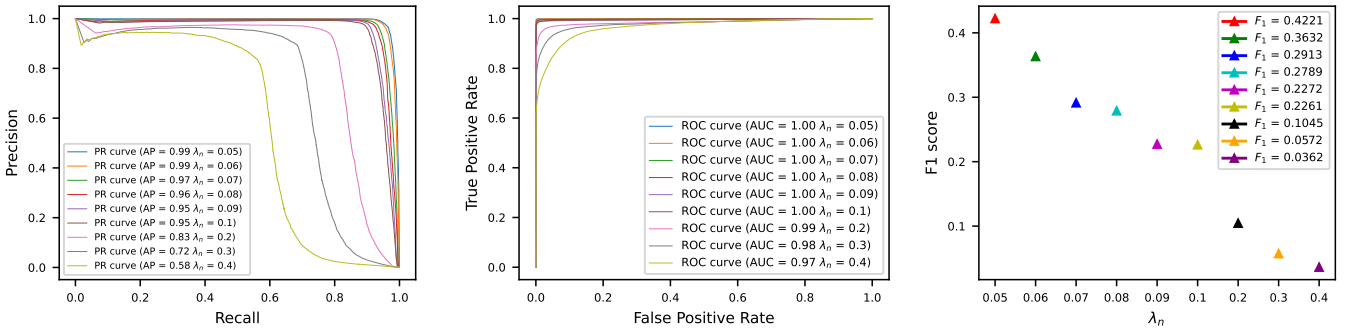
Integrated squared error (ISE): Let P be the reference spectrum probability density and Q be the estimated probability density, then the ISE is given by:

$$ISE(\mathcal{P}, \mathcal{Q}) = \sum_{i=1}^{B} (\mathcal{P}_i - \mathcal{Q}_i)^2. \tag{21}$$

Kullback–Leibler (KL) divergence: The KL divergence is used to measure the difference between two distributions. We use the symmetric KL divergence (KL_s) to evaluate the probability distribution distance between the predicted spectra and the reference spectrum:

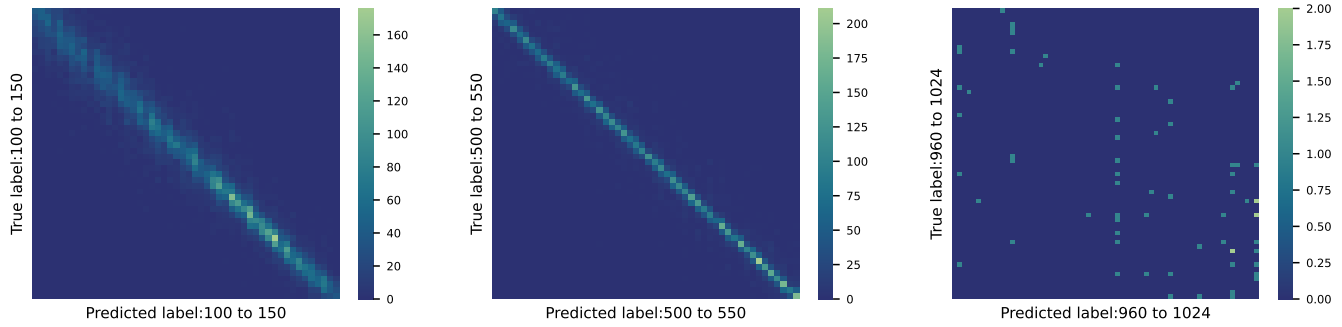
$$D_{KL}(P||Q) = \sum_{i=1}^{B} P(i) \cdot \log\left(\frac{P(i)}{Q(i)}\right), \tag{22}$$

This article has been accepted for publication in IEEE Transactions on Instrumentation and Measurement. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIM.2025.3573370



(a) Macro-average PR curve under different λ_n (b) Micro-average ROC under different λ_n (c) Macro Average F_1 score under different λ_n

Fig. 6: Classification performance by *AttnUNet++* using ^{60}Co as the radiation source with $\mathcal{B} = 1024$, and Poisson process intensity rate λ_n varying between 0.05 and 0.4.



(a) Class 100-150 (b) Class 500-550 (c) Class 960-1024

Fig. 7: A ^{60}Co source with $\mathcal{B} = 1024$ and $\lambda_n = 0.08$. The diagonal elements represent correctly predicted classes.

and

$$KL_s = \left[\frac{1}{2}\sum_{i=1}^{B}\left(\mathrm{KL}\left(\mathcal{P}_i, \|\mathcal{Q}_i\right) + \mathrm{KL}\left(\mathcal{Q}_i \|\mathcal{P}_i\right)\right)\right]. \quad (23)$$

Mean Absolute Error (MAE): This formula calculates the average of the absolute error between the predicted and true values for all samples:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \quad (24)$$

where y_i is the true value, $\hat{y}_i$ is the predicted value, and n is the number of samples.

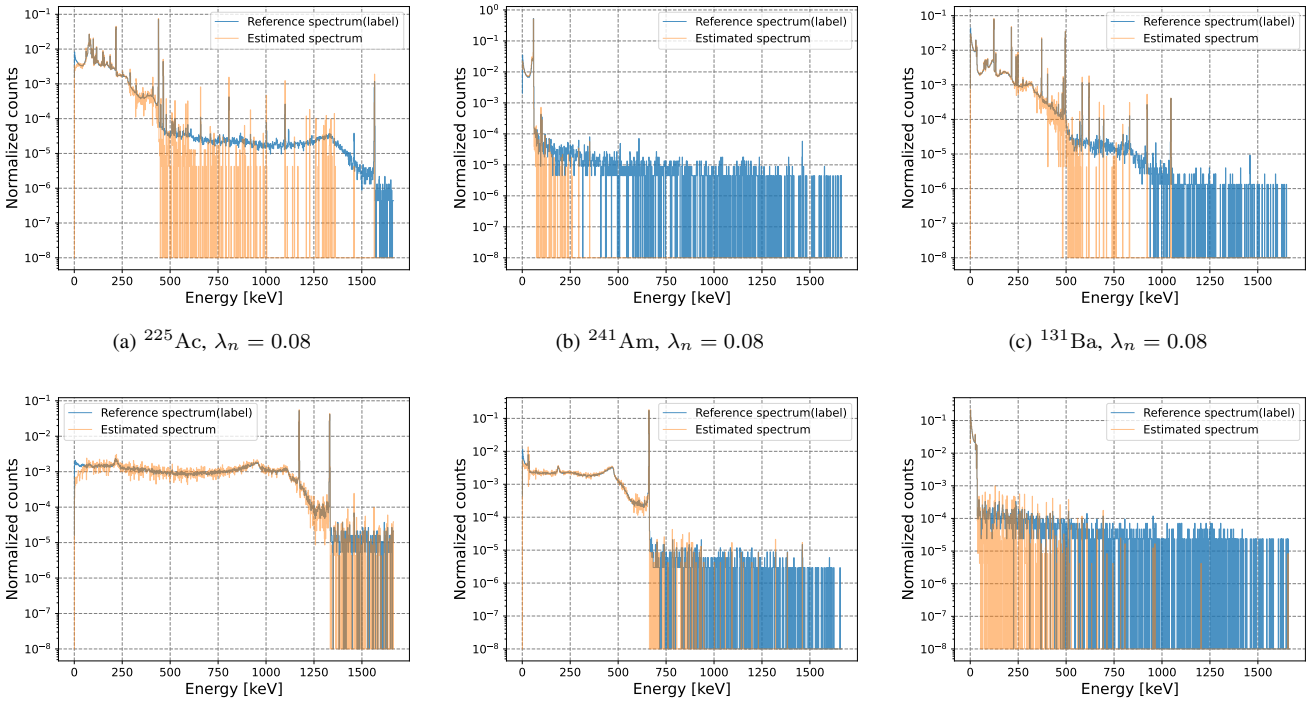
C. Data Imbalance

Before discussing the classification results, it is important to address an inherent limitation in our dataset arising from the physical properties of the signals. Due to the nature of the signal defined in Eq. (3), and the probabilistic analysis in Eqs. (9)-(11), samples that contain no pulses dominate the dataset, comprising approximately 95% to 60% of all samples as the Poisson arrival rate λ increases from 0.05 to 0.4. Moreover, among the samples that do contain pulses, the class distribution remains highly imbalanced due to the energy distribution specified in Eq. (8). Specifically, there are more samples in bins corresponding to lower energy levels and fewer samples in bins corresponding to higher energy levels. This imbalance poses challenges for accurately classifying higher energy levels, as will be evident in the results presented later. Importantly, this imbalance is intrinsic to the signals themselves, stemming from the underlying physical processes governing pulse generation and energy distribution, and cannot be effectively mitigated using traditional data balancing methods designed for tabular datasets.

D. Classification Performance of AttnUNet++

The classification performance of *AttnUNet++* in terms of PR, ROC and F_1 score is presented in Fig. 6. In Fig. 6(a), the micro-average PR decreases as λ_n increases; specifically, the PR curve achieves an AP of 0.99 at $\lambda_n = 0.05$, whereas at $\lambda_n = 0.4$, the PR decreases to 0.58. It suggests that *AttnUNet++* is able to classify all pulses/pile-ups with high precision when λ is no more than 0.1. In Fig. 6(b), the area-under-the-curve (AUC) remains high, approximately 1.00, for λ_n values ranging from 0.05 to 0.1, indicating excellent performance in distinguishing between classes. As λ_n exceeds 0.1, the AUC experiences a slight reduction but remains above 0.97, indicating robust model performance even at higher activity levels. However, when evaluating the macro-average F_1 score shown in Fig. 6(c), there is a significant drop when

This article has been accepted for publication in IEEE Transactions on Instrumentation and Measurement. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIM.2025.3573370



(a) ^{225}Ac, $\lambda_n = 0.08$ (b) ^{241}Am, $\lambda_n = 0.08$ (c) ^{131}Ba, $\lambda_n = 0.08$

(d) ^{60}Co, $\lambda_n = 0.08$ (e) ^{137}Cs, $\lambda_n = 0.08$ (f) ^{125}I, $\lambda_n = 0.08$

Fig. 8: Spectrum estimates under the experiment settings of $\lambda_n = 0.08$, and $\mathcal{B} = 1024$, for 6 different radiation sources: ^{225}Ac, ^{241}Am, ^{131}Ba, ^{60}Co, ^{137}Cs and ^{125}I. The **ISE** metrics are summarized in Table.III.

λ exceeds 0.1, which suggests that *AttnUNet++* is unable to identify the pile-ups when λ is high.

We present confusion matrices in Fig. 7 to demonstrate the *AttnUNet++*'s classification ability for different classes. In Fig. 7, if a pulse/pile-up event is sorted to the correct bins, the count on the diagonal increases; otherwise, off-diagonal counts increase. The accuracy for classes 500–550 of ^{60}Co elements at $\lambda_n = 0.08$ is significantly higher than that for classes 100–150, while the classification performance for classes 960–1024 is poor. The reason of this disparity is two-fold: first, classes 960–1024 lack positive samples; secondly, though classes 100–150 have plenty of samples, in the low-energy spectrum range, the pulse shape is small and the characteristics are few.

E. Spectrum Estimation

In Fig. 8, we present the spectrum estimates of several single sources at a specific activity intensity of $\lambda_n = 0.08$, using logarithmic scales to represent normalized counts in the plots. It can be seen that the peaks in the energy spectrum estimated by the model almost overlap with the actual nuclear peaks and can even predict very small peaks in the high-energy region. However, the model has difficulty estimating values in the low-energy region, approximately between 0 and 50 keV (i.e., classes 1-150), because these pulse signals are either obscured by noise or overlapped with higher-energy stacks, making them difficult to distinguish as we discussed in Fig. 7. In Table III, we compare ISE and KL divergence of energy spectrum estimations using *AttnUNet++* against two conventional algorithms: one without pulse correction and the Fast Correction algorithm [13]. These comparisons are made using different count rates and various sources. Our *AttnUNet++* generally outperforms the two conventional algorithms in terms of ISE and KL metrics. Although in a few cases the conventional algorithms perform adequately, they both fail when the intensity λ_n exceeds 0.3.

For the sources ^{60}Co, ^{137}Cs, ^{57}Co, ^{22}Na, ^{51}Cr, ^{241}Am, we calculated the MAE of the estimated peaks, as shown in Table IV. In terms of peak estimation, when $\lambda_n = 0.05$ and 0.06, the MAE of the Fast Correction algorithm is lower than that of *AttnUNet++*. However, as the counting rate increases, our model achieves better MAE results. Notably, when λ_n reaches 0.3 and 0.4, both conventional methods fail, whereas our method continues to perform accurately. It is worth mentioning that even at relatively low intensities like $\lambda_n = 0.05$, our method remains accurate for peak estimation in the low-energy region of the elements, while the other two algorithms fail.

F. Method Performance Comparison with Recent Research

In this part, the Pulse Height Estimation algorithm developed from [16] is used to compare with our method. To ensure the consistency of the data type and the algorithm application scenario, similar to them, we apply Allpix Squared [30] to generate X-ray pulse data. Allpix Squared is a semiconductor detector simulation framework. It provides a complete and

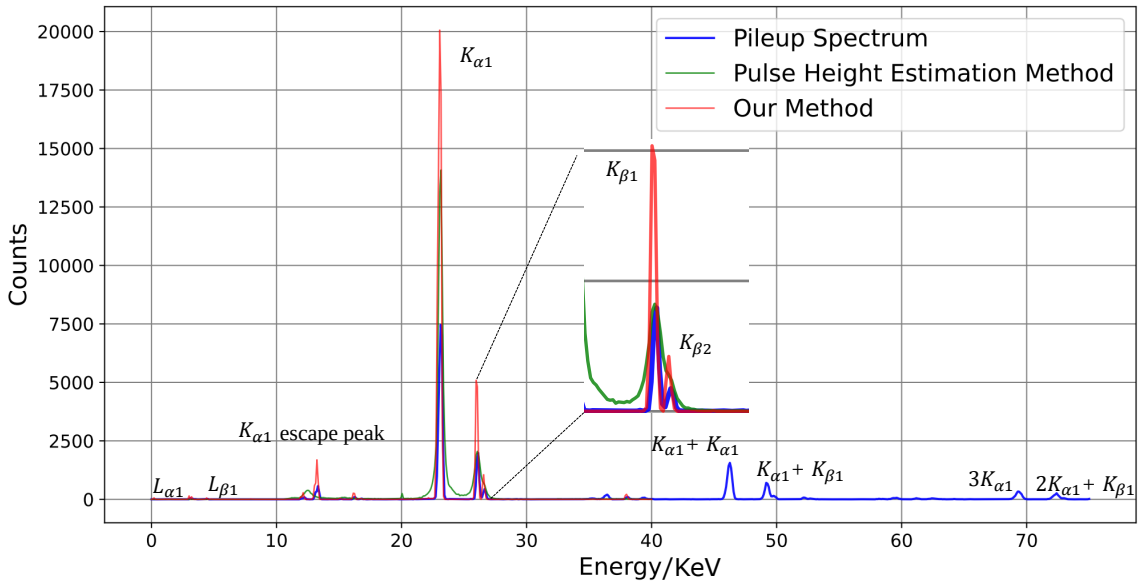


Fig. 9: Comparison of the energy spectra obtained without correction (blue), using a standard pulse height estimation method (green), and using our approach (red).

easy-to-use package for end-to-end simulations of semiconductor detector performance from incident ionizing radiation until the digitization of hits in the detector chip. In the experiment, the Cd is used as the radiation source, and the count rate is 0.27 Mcps; a detailed experiment setting can be found in [31]. We present the estimation result in Fig. 9. As shown in the figure, the pile-up peaks were effectively corrected by both algorithms, especially those peaks over 40 keV. However, the Pulse Height Estimation algorithm fails to separate the characteristic peaks:($K_{\alpha1}$, $K_{\beta1}$ and $K_{\beta2}$ peaks), as well as $K_{\alpha1}$ escape peak. Our method, however, successfully locates the individual peaks and has better resolution performance.

G. Experiment on Real Data

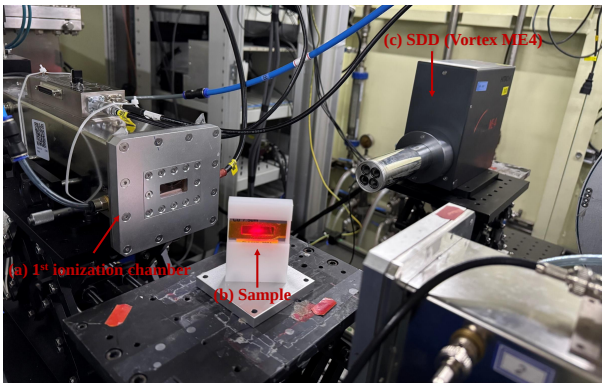


Fig. 10: Diagram of experiment setup in SSRF; (a) 1^{st} ionization chamber, (b) Sample, (c) HITACHI Vortex-ME4 silicon drift detector.

Our approach is further validated using real-world data collected from the Shanghai Synchrotron Radiation Facility (SSRF). During the X-ray detection experiments at SSRF, a HITACHI Vortex-ME4 silicon drift detector, equipped with an integrated pulsed reset charge-sensitive preamplifier, was employed, and the experimental setup is depicted in Fig. 10. Unlike conventional double-exponential pulse signals, the acquired waveform exhibits a quasi-steady step response. The experiment was conducted at a sampling rate of 250 MHz and a counting rate of 1 Mcps, using Cu as the radiation source.

To better capture the pulse characteristics, we apply a differentiation algorithm to extract transient features, followed by Gaussian filtering to suppress noise. After differentiation, the signal reveals an exponential decay behavior, as illustrated in Fig. 11:

$$U(t) = Be^{-t/\lambda}, \tag{25}$$

where B denotes the amplitude factor, and λ represents the decay constant determined by the circuit parameters. The original step-like waveform can be reconstructed via integration:

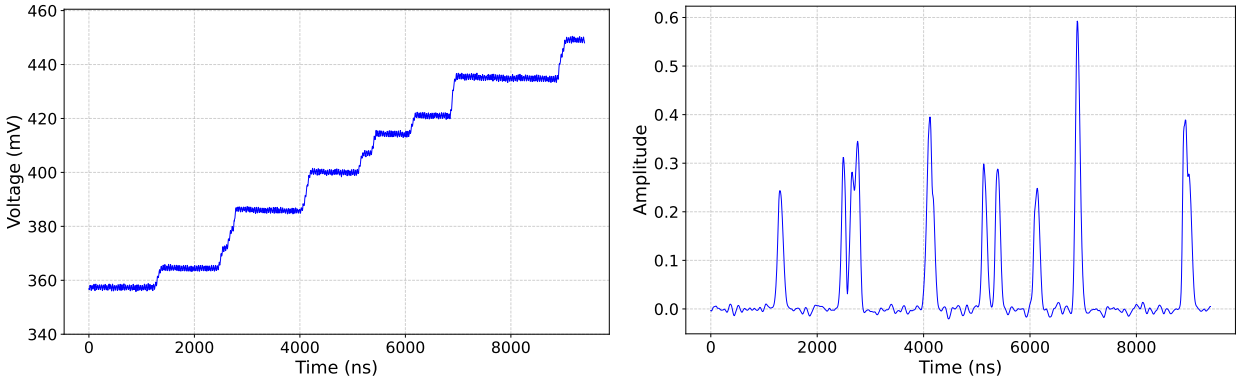
$$u(t) = -B\lambda e^{-t/\lambda} + D, \tag{26}$$

where D corresponds to the signal plateau level relative to the baseline, which can be adjusted to zero by selecting an appropriate reference. In the resulting energy spectra (Fig. 12) our algorithm effectively corrected the pile-up peaks. Further, our method successfully located the individual peaks($K_{\alpha1}$, $K_{\beta1}$).

H. Performance under Different SNRs

Experiments were carried out at different noise levels to better understand our proposed model's robustness in noisy environments. We assessed the model's test accuracy under different SNRs, from 20 dB to 100 dB, as shown in Fig. 13. The results demonstrate that the model exhibits robustness

This article has been accepted for publication in IEEE Transactions on Instrumentation and Measurement. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIM.2025.3573370



(a) Quasi-steady step signal.

(b) Double-exponential pulse signal.

Fig. 11: Raw signal from SSRF; (a) Quasi-steady step signal from a HITACHI Vortex-ME4 silicon drift detector, (b) Double exponential shape pulse signal calculated by using equations (25) and (26) from the signal in (a).

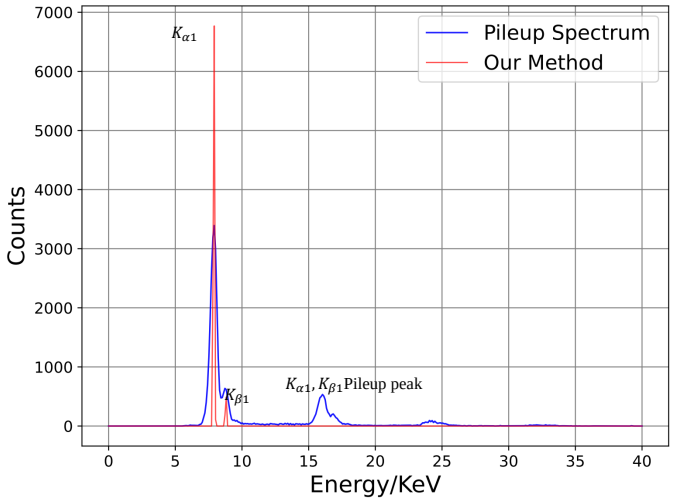


Fig. 12: Estimation result based on measurements in Fig. 11.

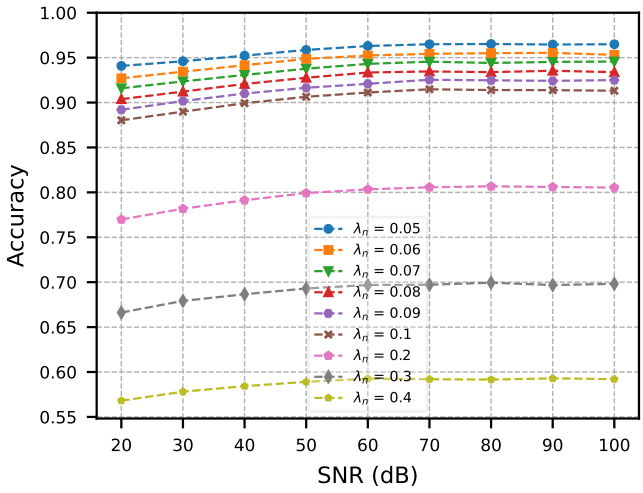


Fig. 13: Model classification accuracy as a function of SNR for different values of λ_n, ^{60}Co as the example.

against noise. However, under low SNR conditions, the classification accuracy decreases. This decline is attributed to the fact that small energy pulse features in the original signal are masked by noise, making it challenging for the model to extract relevant features effectively.

V. Conclusions

This study contributes to the corpus of research on energy spectrum estimation of high-activity measurements by developing a DL-based spectrum generation algorithm that remains effective even under high count rate conditions that result in extreme pulse pile-up. By modeling pile-up events as global recordings and formulating the problem as a point process marking problem, our proposed solution effectively handles the challenge of high-activity spectrum estimation.

Specifically, we apply the sequence segmentation paradigm of DL in this context, combining the advantages of CNNs and the multi-head self-attention mechanism to focus on both local and global features of sequences. The encoder-decoder architecture of the model is used to predict particle energies in an end-to-end manner. Our *AttnUNet++* achieves accurate results for both single and mixed sources at various count rates and demonstrates excellent performance in terms of the MAE of peak positions, as well as the ISE of the overall distribution. Consequently, subsequent elemental and peak analyses can be effectively performed by visualizing the energy spectrum. Moreover, once the model is trained, real-time inference can effectively be performed using dedicated hardware platforms, such as Nvidia Jetson. In future work, we aim to explore customized loss functions (e.g., focal loss, weighted loss) to emphasize high-energy events and attention mechanisms to enhance feature learning for rare pulses, further improving accuracy in high-energy regions. Eventually, since this approach is detector-dependent, we aim to include a pulse shape learning stage to ensure its applicability in a wide range of scenarios.

TABLE III: ISE scores calculated by three algorithms across different values of λ_n, ranging from 0.05 to 0.4, with corresponding $\mathcal{P}$ from 0.4 to 0.999. Multiple sources are mixed with a weight ratio of 1:2. In the table, a dash ('-') indicates that the algorithm failed under the specified conditions. Algorithm 1 represents the case without pile-up correction, while Algorithm 2 refers to the method from [13].

		single-source												multi-source							
Activity	Source name	Co-60		Cs-137		I-125		Ba-131		Ac-225		Am-241		Hg-197m/Er-169		Eu-152/Na-22		Pb-210/Ra-226		Re-186/Sm-153	
λ_n	metrics(/1e-5)	ISE	KL	ISE	KL	ISE	KL	ISE	KL	ISE	KL	ISE	KL	ISE	KL	ISE	KL	ISE	KL	ISE	KL
0.05	Algorithm 1	607.35	745.22	3400.17	958.12	8178.08	629.69	1997.02	1005.94	1363.55	921.72	31252.58	1776.56	1451.22	916.96	866.87	784.87	1788.28	797.72	2007.32	853.97
	Algorithm 2	811.02	819.04	5409.36	1171.70	9055.94	913.06	2328.11	1071.94	1908.95	1001.56	44688.04	2096.18	1663.81	980.56	1002.94	865.31	2081.70	911.78	2483.30	950.87
	Our method	**8.42**	**5.29**	**22.30**	**3.80**	**21.63**	**30.88**	**18.94**	**5.74**	**8.85**	**8.99**	**301.69**	**10.81**	**19.78**	**9.54**	**20.17**	**8.77**	**37.63**	**24.41**	**37.90**	**23.48**
0.06	Algorithm 1	601.90	819.80	3339.87	992.44	8306.99	687.35	1972.17	1005.33	1351.57	982.31	30048.27	1714.99	1449.31	979.13	868.25	853.52	1768.41	868.09	1927.28	948.77
	Algorithm 2	817.50	881.14	5141.76	1330.76	9393.56	1026.41	2329.12	1134.23	1966.70	1119.07	40799.40	2090.24	1663.19	1120.97	990.76	1026.78	2085.14	1006.99	2035.93	1042.92
	Our method	**7.97**	**5.52**	**21.04**	**3.38**	**19.24**	**24.04**	**26.41**	**6.28**	**9.91**	**11.98**	**140.78**	**9.75**	**15.55**	**7.22**	**39.86**	**15.21**	**44.53**	**22.35**	**18.24**	**9.58**
0.07	Algorithm 1	607.10	878.42	3305.98	1024.56	8385.88	746.38	1964.33	1044.50	1414.57	1062.31	29458.06	1658.80	1447.43	1025.93	927.30	916.57	1756.94	933.73	1917.09	969.53
	Algorithm 2	851.11	937.24	5907.48	1415.42	9576.96	1191.30	2361.00	1254.07	2159.74	1271.58	2159.74	1271.58	1588.67	1228.15	1102.60	1071.27	2049.68	1092.68	2034.71	1103.28
	Our method	**4.54**	**3.16**	**24.00**	**4.14**	**17.75**	**22.53**	**19.46**	**4.55**	**7.29**	**8.62**	**104.82**	**10.09**	**18.36**	**7.52**	**25.89**	**9.06**	**35.50**	**13.81**	**63.52**	**21.55**
0.08	Algorithm 1	613.04	930.59	3288.74	1061.60	8446.25	773.42	1969.77	1077.39	1415.30	1100.47	29169.03	1618.44	1463.02	1050.06	1012.09	997.94	1764.54	991.94	1915.58	1017.14
	Algorithm 2	896.00	988.51	6273.52	1481.45	9671.95	1384.95	2258.58	1367.38	2090.79	1342.67	36071.00	2124.13	1608.11	1294.74	1266.63	1164.84	2042.93	1157.46	2024.22	1188.14
	Our method	**10.54**	**8.25**	**15.18**	**2.92**	**14.68**	**16.23**	**18.50**	**4.61**	**11.64**	**10.62**	**57.15**	**9.57**	**11.84**	**4.64**	**26.95**	**8.69**	**24.98**	**9.79**	**35.03**	**11.88**
0.09	Algorithm 1	619.91	973.08	3300.21	1155.11	8491.85	798.08	1973.83	1109.24	1412.49	1115.31	29026.68	1587.66	1485.95	1082.34	1016.46	1055.05	1762.01	1034.57	1913.57	1065.56
	Algorithm 2	978.76	1031.16	6088.80	1560.45	9714.90	1340.82	2287.89	1462.04	2192.40	1398.90	34659.60	2178.07	1652.68	1382.71	1126.79	1218.83	1953.80	1204.76	2025.92	1253.60
	Our method	**4.11**	**2.43**	**28.27**	**5.01**	**11.90**	**13.15**	**40.36**	**6.85**	**8.17**	**8.98**	**28.49**	**7.11**	**11.17**	**4.46**	**19.76**	**5.87**	**64.27**	**21.89**	**21.13**	**8.31**
0.1	Algorithm 1	645.15	1016.63	3303.64	1259.82	8529.12	844.51	1972.49	1122.12	1411.14	1143.76	28954.00	1564.56	1507.49	1134.68	1014.32	1065.34	1765.21	1068.81	1912.84	1096.24
	Algorithm 2	1144.86	1071.95	5287.90	1692.71	9771.96	1444.83	2313.98	1548.69	2323.49	1458.58	33575.91	2277.01	1662.15	1467.78	1124.13	1215.29	1976.84	1257.01	2030.64	1307.19
	Our method	**6.67**	**4.14**	**20.33**	**3.50**	**14.40**	**14.06**	**13.32**	**5.26**	**7.63**	**8.89**	**29.42**	**7.17**	**14.23**	**5.06**	**27.95**	**6.86**	**32.13**	**14.51**	**18.96**	**7.80**
0.2	Algorithm 1	715.47	1193.06	3398.25	1278.38	8733.77	904.60	2010.89	1263.03	1422.62	1277.53	28886.60	1511.14	1545.67	1271.61	1019.51	1212.52	1868.53	1212.78	1966.84	1236.26
	Algorithm 2	1703.28	1284.20	4238.82	1592.46	-	-	2756.21	2176.53	1555.14	1539.34	77198.67	13171.61	2003.63	1820.08	1378.60	1329.11	2225.11	1335.92	2192.22	1382.10
	Our method	**12.73**	**7.99**	**73.15**	**14.98**	**9.24**	**9.01**	**20.88**	**5.73**	**40.05**	**11.49**	**72.04**	**5.56**	**28.16**	**6.48**	**276.06**	**34.07**	**53.92**	**13.64**	**115.81**	**26.45**
0.3	Algorithm 1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Algorithm 2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Our method	**36.69**	**20.10**	**176.82**	**24.38**	**20.25**	**11.11**	**150.70**	**17.14**	**272.05**	**31.31**	**138.90**	**6.15**	**227.90**	**20.81**	**947.40**	**75.97**	**640.46**	**44.04**	**393.77**	**65.75**
0.4	Algorithm 1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Algorithm 2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Our method	**68.22**	**31.35**	**336.13**	**41.10**	**74.78**	**13.32**	**507.86**	**40.66**	**1212.30**	**90.11**	**291.99**	**6.71**	**789.50**	**60.26**	-	-	-	-	**1034.17**	**147.48**

TABLE IV: MAE metrics corresponding to the results shown in Table III.

Activity	Source name	Co-60		Cs-137		Co-57		Na-22		Cr-51		Am-241
λ_n	peaks energy/keV	1173.24	1332.50	31.80	661.66	122.06	136.47	511.20	1274.54	0.17	319.69	59.54
0.05	Algorithm 1	0.02374	0.02069	-	0.06329	0.16046	0.01597	0.04675	0.01939	**0.00216**	0.10012	-
	Algorithm 2	**0.00125**	**0.00132**	-	**0.00182**	**0.06158**	**0.00170**	**0.00251**	0.00738	0.00383	**0.00522**	-
	Our method	0.00356	0.002452	**0.00127**	0.00570	0.00733	0.00078	0.00866	**0.00319**	0.01255	0.01498	**0.03581**
0.06	Algorithm 1	0.02595	0.02268	-	0.06912	0.18108	0.01765	0.05546	0.01232	0.00408	0.11983	-
	Algorithm 2	**0.00285**	0.00258	-	0.00621	0.10049	0.00253	0.01348	0.00165	**0.00248**	0.02733	-
	Our method	0.00319	**0.00247**	**0.00191**	**0.00574**	**0.00584**	**0.00057**	**0.00477**	**0.00098**	0.00986	**0.01201**	**0.02085**
0.07	Algorithm 1	0.02781	0.02426	-	0.07405	0.19518	0.01895	0.06665	0.01316	0.00572	0.13352	-
	Algorithm 2	0.00460	0.00436	-	0.01111	0.10998	0.00391	0.03555	0.00298	**0.00413**	0.05210	-
	Our method	**0.00204**	**0.00164**	**0.00317**	**0.00562**	**0.00547**	**0.00053**	**0.00362**	**0.00097**	0.00845	**0.01269**	**0.01658**
0.08	Algorithm 1	0.02935	0.02548	-	0.07797	0.20482	0.01997	0.07379	0.01393	0.00657	0.14289	-
	Algorithm 2	0.00670	0.00686	-	0.01760	0.13125	0.00580	0.06052	0.00397	**0.00491**	0.07316	-
	Our method	**0.00343**	**0.00273**	**0.00293**	**0.00440**	**0.00628**	**0.00081**	**0.00460**	**0.00154**	0.01103	**0.01164**	**0.00898**
0.09	Algorithm 1	0.03048	0.02651	-	0.08332	0.21131	0.02080	-	0.01453	0.00733	0.14927	-
	Algorithm 2	0.01010	0.00978	-	0.01378	0.14872	0.00814	-	0.00552	0.00563	0.09106	-
	Our method	**0.00229**	**0.00162**	**0.00476**	**0.00601**	**0.00587**	**0.00074**	**0.00487**	**0.00143**	**0.00513**	**0.00868**	**0.00591**
0.1	Algorithm 1	0.03245	0.02742	-	0.08889	0.21590	0.02153	-	-	0.00774	-	-
	Algorithm 2	0.01110	0.01289	-	0.01054	0.16473	0.01061	-	0.00712	**0.00593**	-	-
	Our method	**0.00251**	**0.00217**	**0.00383**	**0.00556**	**0.00568**	**0.00056**	**0.00529**	**0.00141**	0.01105	**0.01465**	**0.00677**
0.2	Algorithm 1	-	-	-	-	-	0.02624	-	-	0.00782	-	-
	Algorithm 2	-	-	-	-	-	0.02944	-	-	0.00376	-	-
	Our method	**0.003357**	**0.00298**	**0.00955**	**0.00959**	**0.00714**	**0.00115**	**0.00734**	**0.00156**	**0.00188**	**0.01595**	**0.01239**
0.3	Algorithm 1	-	-	-	-	-	-	-	-	-	-	-
	Algorithm 2	-	-	-	-	-	-	-	-	-	-	-
	Our method	**0.00654**	**0.00503**	**0.01519**	**0.01315**	**0.01035**	**0.00159**	**0.01221**	**0.00253**	**0.01862**	**0.02108**	**0.01802**
0.4	Algorithm 1	-	-	-	-	-	-	-	-	-	-	-
	Algorithm 2	-	-	-	-	-	-	-	-	-	-	-
	Our method	**0.00972**	**0.00668**	**0.01716**	**0.01680**	**0.01814**	**0.00202**	**0.01686**	**0.00318**	**0.04625**	**0.02914**	**0.02619**

REFERENCES

[1] M. Bolic, V. Drndarevic, and W. Gueaieb, "Pileup correction algorithms for very-high-count-rate gamma-ray spectrometry with nai(tl) detectors," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 1, pp. 122–130, 2010.

[2] X. Zhong, L. Chen, B. Wang, and G. Jin, "A spectrometer with baseline correction and fast pulse pile-up rejection for prompt gamma neutron activation analysis technology," *Review of Scientific Instruments*, vol. 89, p. 123504, 12 2018.

[3] M. Bolic and V. Drndarevic, "Processing architecture for high count rate spectrometry with nai(tl) detector," in *2008 IEEE Instrumentation and Measurement Technology Conference*, pp. 274–278, 2008.

[4] P. A. B. Scoullar and R. J. Evans, "Maximum likelihood estimation techniques for high rate, high throughput digital pulse processing," in *2008 IEEE Nuclear Science Symposium Conference Record*, pp. 1668–1672, 2008.

[5] M.-R. Mohammadian-Behbahani and S. Saramad, "Pile-up correction algorithm based on successive integration for high count rate medical imaging and radiation spectroscopy," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 897, pp. 1–7, 2018.

[6] M. D. Haselman, J. Pasko, S. Hauck, T. K. Lewellen, and R. S. Miyaoka, "Fpga-based pulse pile-up correction with energy and timing recovery," *IEEE Transactions on Nuclear Science*, vol. 59, no. 5, pp. 1823–1830, 2012.

[7] T. Petrovič, M. Vencelj, M. Lipoglavšek, R. Novak, and D. Savran, "Efficient reduction of piled-up events in gamma-ray spectrometry at high count rates," *IEEE Transactions on Nuclear Science*, vol. 61, no. 1, pp. 584–589, 2014.

[8] B. A. VanDevender, M. P. Dion, J. E. Fast, D. C. Rodriguez, M. S. Taubman, C. D. Wilen, L. S. Wood, and M. E. Wright, "High-purity germanium spectroscopy at rates in excess of 10^6 events/s," *IEEE Transactions on Nuclear Science*, vol. 61, no. 5, pp. 2619–2627, 2014.

[9] T. Trigano, I. Gildin, and Y. Sepulcre, "Pileup correction algorithm using an iterated sparse reconstruction method," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1392–1395, 2015.

[10] B. Liu, M. Liu, M. He, Y. Ma, and X. Tuo, "Model-based pileup events correction via kalman-filter tunnels," *IEEE Transactions on Nuclear Science*, vol. 66, no. 1, pp. 528–535, 2019.

[11] C. McLean, M. Pauley, and J. H. Manton, "Limitations of Decision Based Pile-Up Correction Algorithms," in *2018 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 693–697, June 2018.

[12] T. Trigano, A. Souloumiac, T. Montagu, F. Roueff, and E. Moulines, "Statistical pileup correction method for hpge detectors," *IEEE Transactions on Signal Processing*, vol. 55, no. 10, pp. 4871–4881, 2007.

[13] T. Trigano, E. Barat, T. Dautremer, and T. Montagu, "Fast digital filtering of spectrometric data for pile-up correction," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 973–977, 2015.

[14] C. Mclean, M. Pauley, and J. H. Manton, "Non-parametric decompounding of pulse pile-up under gaussian noise with finite data sets," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2114–2127, 2020.

[15] P. Ilhe, É. Moulines, F. Roueff, and A. Souloumiac, "Nonparametric estimation of mark's distribution of an exponential shot-noise process," *Electronic Journal of Statistics*, vol. 9, no. 2, pp. 3098 – 3123, 2015.

[16] B. Jeon, S. Lim, E. Lee, Y.-S. Hwang, K.-J. Chung, and M. Moon, "Deep learning-based pulse height estimation for separation of pile-up pulses from nai(tl) detector," *IEEE Transactions on Nuclear Science*, vol. 69, no. 6, pp. 1344–1351, 2022.

[17] I. Morad, M. Ghelman, D. Ginzburg, A. Osovizky, and N. Shlezinger, "Model-based deep learning algorithm for detection and classification at high event rates," *IEEE Transactions on Nuclear Science*, 2024.

[18] M. Kamuda, J. Zhao, and K. Huff, "A comparison of machine learning methods for automated gamma-ray spectroscopy," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 954, p. 161385, Feb. 2020.

[19] T. Trigano and D. Bykhovsky, "Deep learning-based method for activity estimation from short-duration gamma spectroscopy recordings," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–11, 2024.

[20] Y. Huang, X. Zheng, Y. Zhu, T. Trigano, D. Bykhovsky, and Z. Chen, "Deep learning based pile-up correction algorithm for spectrometric data under high-count-rate measurements," *Sensors*, vol. 25, no. 5, 2025.

[21] Z. Chen, D. Bykhovsky, X. Zheng, T. Trigano, and Y. Zhu, "Gasim:

A python class to generate simulated time signals for gamma spectroscopy," *SoftwareX*, vol. 29, p. 102037, 2025.

[22] D. Bykhovsky, Z. Chen, Y. Huang, X. Zheng, and T. Trigano, "Advanced spectroscopy time-domain signal simulator for the development of machine and deep learning algorithms," *IEEE Sensors Letters*, pp. 1–4, 2025.

[23] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media, 2007.

[24] D. F. Yu and J. A. Fessler, "Mean and variance of coincidence counting with deadtime," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 488, no. 1, pp. 362–374, 2002.

[25] M. Jacobsen and J. Gani, "Point process theory and applications: marked point and piecewise deterministic processes," 2006.

[26] G. F. Knoll, *Radiation Detection and Measurement*. Hoboken, N.J: Wiley, 4 edition ed., Aug. 2010.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[28] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," 2018.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[30] S. Spannagel, K. Wolters, D. Hynds, N. A. Tehrani, M. Benoit, D. Dannheim, N. Gauvin, A. Nürnberg, P. Schütze, and M. Vicente, "Allpix2: A modular simulation framework for silicon detectors," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 901, pp. 164–172, 2018.

[31] C. Lin, Z. Chen, C. Feng, X. Zheng, Y. Zhu, T. Trigano, and D. Bykhovsky, "An open x-ray spectrometric dataset for deep learning-based pile-up correction," in *Wireless Artificial Intelligent Computing Systems and Applications*, Springer Nature Switzerland, 2025. accepted for publication.