scientific reports

Check for updates

OPEN

# Light-powered end-to-end neutron detection and imaging with an edge-deployed optical AI chip

Shanny Lin[1,2], Hanqing Zhu[2], Steven Clayton[1], C. L. Morris[1], David Z. Pan[2], Zhaowen Tang[1], Ray T. Chen[2,3,4]✉ & Zhehui Wang[1]✉

**Neutron detection is widely used in many applications including nuclear physics, nuclear energy, nuclear technologies and nuclear safeguards. Developing an end-to-end neutron detection and imaging workflow paves way towards fully automated processes for many applications. We implemented an automated workflow for neutron detection experiments which use a solid state image sensor to capture neutron hits as a digital image. We deploy the workflow to an edge-based optical neural network (ONN) to increase the radiation-hardness and lifetime of neutron detection instruments. We present a two-stage neural network framework for detection of neutrons at sub-pixel resolution. The first stage uses a region proposal network to efficiently detect and extract neutron hits from the input camera image. The second stage feeds the extracted hits into a fully connected neural network to predict the sub-pixel hit position. The performance of the two-stage framework is evaluated using the edge-based ONN. The results show that we can achieve above 96% neutron detection accuracy as well as sub-pixel and sub-micron position resolution, while enjoying the advantages of the ONN hardware including radiation-hardness, low energy consumption and high computing speed for integrated edge camera and hardware deployment, when compared with electronic counterparts.**

**Keywords** Ultracold neutrons, Deep learning, Edge computing, Optical neural networks

In recent years, neutron detection and imaging techniques have been used in a wide variety of fields and applications including but not limited to nondestructive testing, material science, national security and nuclear safeguards, nuclear reactor monitoring and safety, and scientific research such as fusion monitoring and particle physics[1,2]. For nondestructive testing, many applications need to explore the internal structures of materials without damage or dismantling. Similar to X-ray radiography, the attenuation of the neutron beam intensity varies with different material composition, density and thickness. However, neutron radiography offers a unique perspective than X-rays as neutrons can detect and identify light elements such as hydrogen and lithium for visualization as well as penetrate deeper into dense materials and elements such as iron and lead[3,4]. This unique property allows for clear imaging of the light elements in material science research including the studying and imaging of lithium batteries[5] and hydrogen-rich rocks and soils[6]. In addition, neutron detection is used for national security and nuclear safeguards to prevent illicit trafficking and transportation of nuclear materials and radioactive sources including uranium and plutonium[7]. Furthermore, it is used to monitor nuclear reactor cores to verify the nuclear fuel cycle and to monitor reactor conditions such as neutron flux and power levels[8]. This type of neutron monitoring also used for fusion monitoring to measure the neutron flux to estimate the total fusion power and energy[9], as well as in particle physics to probe the fundamental interactions[10]. Recently, a passive neutron coincidence collar has been developed to use neutron detection as a nondestructive method to assay the fissile concentration within Light Water Reactor fresh fuel assemblies[11]. Nonetheless, all these neutron imaging and detection applications require high spatial resolution to produce high resolution images and to localize individual neutron events. However, achieving high resolution is a challenge as traditional methods are often not an automated process, and thus require offline development and data analysis. In this work, we implement an end-to-end automated workflow for neutron detection experiments which use solid state image sensors to capture neutron hits as a digital image. Specifically, we develop the automated workflow for ultracold neutron (UCN) direct detection experiments.

[1]Los Alamos National Laboratory, Los Alamos, NM 87545, USA. [2]Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78705, USA. [3]The University Texas at Austin, Microelectronics Research Center, Austin, TX 78758, USA. [4]Omega Optics, Inc., Austin, TX 78757, USA. ✉email: chenrt@austin.utexas.edu; zwang@lanl.gov

UCNs have a kinetic energy below 400 neV and are the coldest free neutrons produced in laboratories. Their very low kinetic energy, and correspondingly very long de Broglie wavelength, allow UCNs to be stored in material bottles as they are totally reflected from the surfaces of any material with large enough Fermi potential[10,12]. This unique property enables UCNs to be a powerful tool in studying the fundamental sciences in nature such as the neutron lifetime[13], the neutron beta decay asymmetries[14,15] and the neutron electric dipole moment[16,17]. In addition, UCNs are a perfect tool to study the quantum gravitational states of bouncing neutrons[10,18]. These experiments require position measurements with a spatial resolution of 1 $\mu$m or less to resolve individual quantum states. Other position-sensitive UCN studies include but are not limited to UCN spectroscopy and materials research[19]. Accurate real-time UCN position resolution will allow for precise determination of UCN position, energies and kinetic information to further scientific research with UCNs.

Position-sensitive measurements of UCNs have been reported using either indirect or direct detection methods. As neutrons lack an electrical charge, both detection methods require a nuclear reaction to generate ionizing charged particles[2,20]. In our experiments, we utilize the nuclear reaction of a neutron with isotropically purified Boron-10 ( $^{10}$B) thin films, which is deposited onto the detector using an electron-evaporation deposition process[21]. The resulting neutron capture reaction creates an excited and unstable Boron-11 ( $^{11}$B$*$) isotope that immediately undergoes fission to generate[1,2]:

$$\begin{aligned} n +^{10}\mathrm{B} \quad &\rightarrow^{11} \mathrm{B}* \rightarrow^{4} \mathrm{He}(1.48MeV) +^{7} \mathrm{Li}(0.84MeV) + \gamma(0.48MeV) \qquad (94\%) \\ n +^{10}\mathrm{B} \quad &\rightarrow^{11} \mathrm{B}* \rightarrow^{4} \mathrm{He}(1.78MeV) +^{7} \mathrm{Li}(1.02MeV) \qquad (6\%) \end{aligned} \tag{1}$$

The indirect detection method uses scintillators to convert the ionizing charged particles to light photons which are detected by photomultiplier tubes (PMTs). The UCN position is then determined by analyzing the light intensity distribution of the detected photons. However, the spatial resolution of the captured UCNs are limited by factors such as PMT size[21,22] and light yield and optics of the imaging camera[19]. Meanwhile, the direct detection method uses solid state imaging sensors such as charge-couple devices (CCDs) and active-pixel sensors, also known as complementary metal-oxide semiconductor (CMOS) sensors, to capture UCN hits as digital images. The ionizing charged particles are stopped within the sensor's active silicon layer and the deposited energy creates electron-hole pairs. The internal electric field propagates one charge carrier type to the sensor's potential wells and the total charge collected per pixel is output as a digital UCN hit image. The position resolution for the direct detection method is naturally enhanced by selecting image sensors with smaller pixel sizes. The spatial resolution is determined by analyzing the pixel intensity distribution, resulting from charge spreading into neighboring pixels, by fitting a 2D Gaussian fit, where the fitted centroid is chosen as the hit position[23]. Nonetheless, the Gaussian fitting method does not take into account sensor physics and the ground-truth hit position information is unknown as the position resolution of silicon detectors are limited by factors such as pixel size, charge spreading and diffusion effects.

To address the real-time detection and super position resolution challenge, we introduce a two-stage deep learning framework which takes as input UCN hit images and outputs the hit position for each detected UCN. The first stage performs UCN detection using a region proposal network (RPN) that generates bounding boxes around detected UCN hits. The RPN is inspired by the You Only Look Once v3 (YOLOv3) network[24], which is popularly used for its fast and efficient feature extraction properties for object detection by simultaneously performing bounding box regression and classification. The second stage inputs the detected UCN hits into a fully connected neural network (FCNN) to predict the sub-pixel hit position following from our previous work[25]. The hidden layers of the FCNN aims to model the underlying detector physics that produces the pixel intensity distribution of the UCN hits.

For real-time operation, the two-stage framework needs to be deployed on the edge device with the image sensor. There are many available options of electronic-based embedded hardware for artificial intelligence (AI) in the hardware industry[26]. However, UCN experiments and the broader scope of radiation experiments, such as X-ray and proton radiography, will subject the electronic hardware to radiation environments proliferated with electromagnetic interference (EMI) and electromagnetic pulse (EMP) effects. Such environments can have significant adverse effects on electronic hardware ranging from performance degradation to device failure. A few examples of damaging sources include, but are not limited to, total ionization dose, displacement damage and single-event effects[27,28]. Meanwhile, optical neural networks (ONNs), a promising hardware platform for next generation neurocomputing, offers many advantages over electronic-based hardware under exposure to radiation. Optical systems are naturally more resistant to radiation due to the transmission of photons as opposed to electrons and optical materials such as optical fibers are less sensitive to radiation-induced material degradation. In addition, ONNs enjoy additional advantages over electronic systems such as lower power consumption, high parallel processing capability, higher bandwidth, resistance to radiation induced noise, and lower latency[29–31].

In this work, we introduce the two-stage framework for UCN detection and sub-micron position resolution given an input hit image and demonstrate the network performance on our edge-deployed Optical Segmented Neural Network (OSENN). The rest of this paper is summarized as follows. In Section "Proposed Methods", we discuss the data generation process and the network details of the RPN and FCNN networks of the two-stage framework. In addition, we also discuss the network training strategy. In Section "Results and Discussion", we report the detection and position resolution performance of the two-stage framework using a laptop graphic processing unit (GPU) and the OSENN. Lastly, Section "Conclusion" concludes this research work.

## Proposed method

In this section, we describe the synthetic data generation process and the two-stage deep learning architecture and algorithm. To obtain the goal of sub-micron position resolution, we need to generate synthetic UCN hit data with ground-truth position information as it is not available experimentally. The data generation process then follows the label requirements needed to train the detection and super position networks.

## UCN data generation

An overview of the data generation process and where the data is used during the deep learning training process is illustrated in Fig. 1. We use Allpix Squared[32], an open-source silicon detector simulation tool, to generate synthetic UCN data consisting of UCN hit images and their corresponding ground-truth hit position labels. Allpix Squared implements end-to-end Monte Carlo simulations from the incident particle detection to the digitized detector output image. The simulation process is divided into several stages to model different detector processes such as energy deposition, charge transport, digitization and other front-end detector processes. In addition, Allpix Squared can save the entire simulation history and the ground-truth information for each stage. Therefore, we can generate a database of UCN hit images and their corresponding ground-truth position information. In our previous work[25], we used this process to generate a large database of single UCN hit images of size $14 \times 14$ pixels with position labels to train a super position network. The single UCN hit database is generated by first randomly sampling the particle type and their corresponding energy following from equation (1). Specifically, the four different combinations of particles and energies, $^{4}$He(1.48MeV), $^{7}$Li(0.84MeV), $^{4}$He(1.78MeV), and $^{7}$Li(1.02MeV), were randomly sampled with associated probabilities 0.47, 0.47, 0.03, and 0.03, respectively. The particles and their associated energy are the input to the Allpix Squared simulation to obtain its simulated hit image and corresponding ground-truth hit position. However, in this work, we need to consider larger images with multiple UCN hits and bounding box labels in order to train the detection network. Lastly, we note that our synthetic database of UCN hits is shown to be very similar to the experimental in both hit images and energy spectrum analysis, where the energy spectrum between synthetic and experimental are more closely matched if the image sensor is fully characterized[25].

As the detection network design is inspired by YOLOv3[24], the training data consists of a collection of images of size $416 \times 416$ and corresponding text files consisting of bounding box labels $[x_c, y_c, w, h]$ of all objects within an image. The bounding box labels denote the center coordinates of the bounding box $(x_c, y_c)$, the width $w$ and the height $h$, such that the created box captures the object of interest. We generate the detection dataset by randomly selecting a number of single hits from the single UCN hit database and then placing them in uniformly random locations within the area of a $416 \times 416$ black image. The center coordinate $(x_c, y_c)$ of the bounding box label is created by referencing the position of the center of the single hit image to its coordinate on the $416 \times 416$ image. Next, we set the width and height labels directly to 8 $(w = h = 8)$ to create bounding boxes of size $8 \times 8$ pixels. For simplicity, we shall refer to the coordinates of the $416 \times 416$ image as the *global* coordinates and the single hit images as *local* coordinates. Note that we do not include a class label that is used by YOLOv3 as we are only detecting one class that is UCNs. Lastly, we add camera noise by adding random Gaussian noise to the generated image. Note that other noises such as fixed pattern noise and cold/hot pixels can also be added, however, they can be corrected through background subtraction. Therefore, we do not consider them in this work. We note that the camera active imaging area is typically much greater than $416 \times 416$ pixels. Our current experiments use a CMOS image sensor with an active area size of $3872 \times 2764$ pixels. However, the computational cost to process such a large image as input to a deep neural network is significantly greater as it would require more memory and high-performance processing power for both training and inference. To reduce the computational burden for edge computing, we propose to process the image frame in patches of $416 \times 416$ pixels.

Lastly, we generate the dataset to train the FCNN super position network which takes single UCN hit images as input and outputs the hit position. This step uses the trained detection network to extract detected UCNs

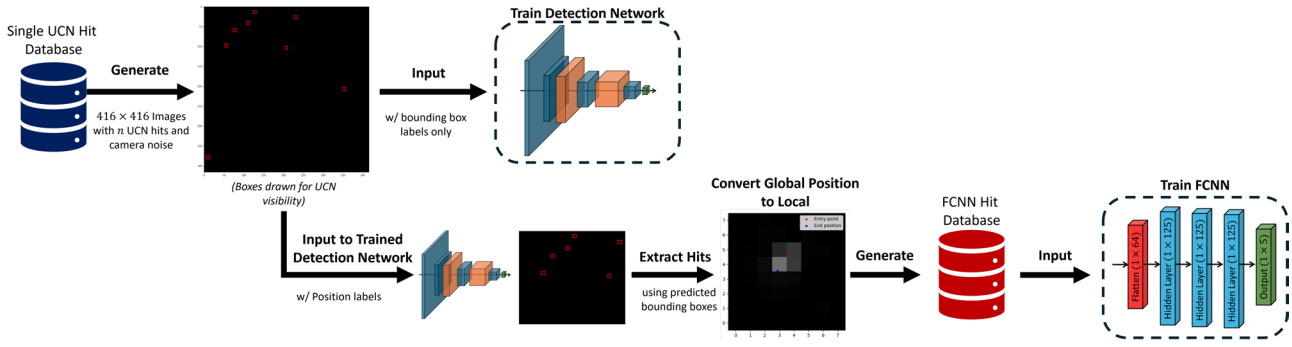


**Fig. 1**. An overview of the data generation process. The detection network data is generated by randomly placing a random number of single UCN hits on a black $416 \times 416$ image and creating a corresponding text file consisting of the bounding box labels for each UCN hit. The FCNN database is generated by using the trained detection network to extract the detected UCN hits as $8 \times 8$ images. The position labels are created by converting the global position to their local positions.

captured by the predicted $8 \times 8$ bounding boxes to generate the single hit FCNN database with hit position labels. Note that we do not use the local position data used to generate the detection network data to train the FCNN as the predicted bounding box (cyan) does not exactly match the ground-truth bounding box (red), shown in Fig. 4a. The shift in bounding box position will result in a shift in the local hit position and thus, the hit position labels must be relative to the predicted bounding box. In summary, the FCNN dataset is generated by extracting single $8 \times 8$ UCN hits predicted by the detection network as the input image and converting the global hit coordinates to the relative local coordinates as the ground-truth position labels.

## Two-stage network framework

The overview of the two-stage framework is illustrated in Fig. 2 where the detection and super position stages use a RPN and FCNN, respectively.

**RPN for detection** Our RPN architecture design and algorithm is inspired by YOLOv3 for its fast and efficient object detection capabilities compared to other popular networks such as Faster Region-based Convolutional Neural Network (Faster R-CNN)[33], Single Shot Detector (SSD)[34], Region-based Fully Convolutional Network (R-FCN)[35], and RetinaNet[36]. It is shown in the technical report[24] that YOLOv3 results in significantly faster inference times with higher or comparable classification accuracy performance. While YOLOv3 is well known for both object detection and classification, we mainly focus on its detection capabilities as the task of UCN detection can be considered as only one class. However, we note that the classification capability can be further expanded to classify the UCN hits into the two charged particle products ( $^{4}$He and $^{7}$Li) with two different energies each as denoted in (1), which will be addressed in other publications.

The RPN architecture is designed with 10 convolutional layers which can be broken down into two main components, namely the convolutional and residual blocks. The overview of the RPN architecture is shown in Fig. 2a. The convolutional blocks consists of a convolutional layer followed by a batch normalization and a Leaky ReLU activation function to extract image features. The residual block consists of two convolutional blocks with the purpose of mitigating the vanishing gradient issue commonly encountered when training deep neural networks. Lastly, the output block consists of one convolutional block followed by one convolutional layer. This final convolutional layer divides the input image into feature maps of size $52 \times 52$ pixels to naturally support predicting bounding boxes of size $8 \times 8$ with unit anchor. For our predictions, we end up with 5 output values: the confidence score of the predicted bounding box and the 4 coordinates of the bounding box which are $[x_c, y_c, w, h]$. As a result, the RPN predicts $(52 \times 52) = 2704$ bounding boxes for each image which is then reduced to the actual number of detected UCN hits through non-maximum suppression (NMS).

**FCNN for super position** We design the FCNN architecture following from our previous work[25] and an overview of the architecture is shown in Fig. 2b. It takes as input the extracted $8 \times 8$ UCN hit images and flattens the image into a vector of size $1 \times 64$. The flattened layer is followed by 3 hidden layers with 125 neurons each. Lastly, the output layer consists of 5 labels, namely the entry and exit coordinates as well as the incident angle $\theta$, the angle at which the charged particle enters the detector. The FCNN aims to model the underlying detector physics by learning a mapping between the input hit image and the ground-truth labels. Note that dropout layers with a dropout rate of 0.2 are implemented after each hidden layer to mitigate model overfitting and to enable uncertainty quantification during testing, but are not shown in the architecture overview.

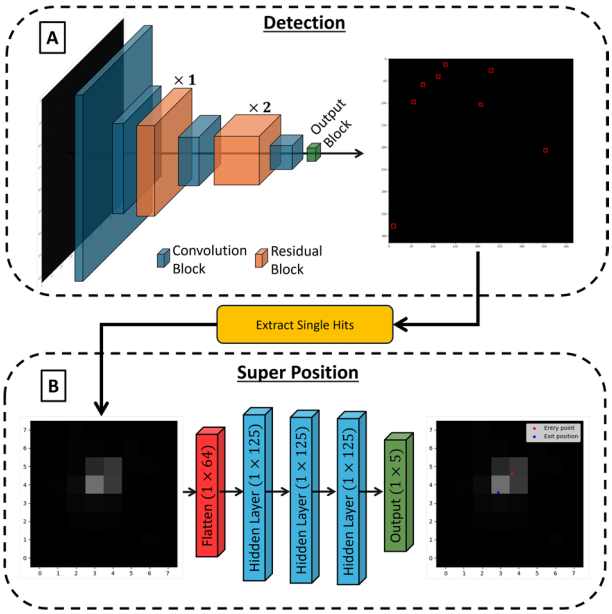


**Fig. 2**. An overview of the two-stage network framework. (**a**) The detection stage uses a RPN that takes as input a camera image of size $416 \times 416$ and predicts bounding boxes (red) of detected UCN hits. Next, the detected hits of image size $8 \times 8$ are extracted and inputted into (**b**) FCNN for super position resolution.

## Optical Segmented Neural Network (OSENN)

In recent years, the deep learning training computation, measured in floating operations per second (FLOPS), has been approximately quadrupling every year[37] and thus results in increasing computing costs. The ONN is an emerging next-generation neurocomputing platform for deep learning that features highly parallel, high speed, wide bandwidth, low-latency and near-zero energy computation[29,38–41]. ONNs are built using a series of photonic tensor cores that are designed to enhance and realize matrix vector multiplication with light-speed computation. We propose to use ONNs as a computation platform not only for its high speed and energy efficient computing, but also for its potential as a *radiation-hard* computation platform as photons are immune to radiation-induced degradation. We specifically propose to use our group's optical segmented neural network (OSENN), shown in Fig. 3a, a special design of ONN that implements segmented neural networks to optimize the scalability and efficiency of OSENNs at the hardware level.

Figure 3b illustrates the process of mapping a weight matrix W to the OSENN chip. First, the OSENN partitions the weight matrix $\mathrm{W}^l \in \mathbb{C}^{m\times n}$ of each layer $l$ of the neural network into submatrices of size $k \times k$, where $k = 4n$ and $n = 1, 2, 3, \cdots$. This results in $\lceil\!\lceil \frac{m}{k} \rceil\!\rceil \times \lceil\!\lceil \frac{n}{k} \rceil\!\rceil$ submatrices $\mathrm{W}^l_{i,j} \in \mathbb{C}^{k\times k}$, where $i \in \left[1, \lceil\!\lceil \frac{m}{k} \rceil\!\rceil\right]$ and $j \in \left[1, \lceil\!\lceil \frac{n}{k} \rceil\!\rceil\right]$. Similarly, the input vector $\mathrm{x}^l_{in} \in \mathbb{R}^n$ is also partitioned into subvectors $\mathrm{x}^l_{in,j} \in \mathbb{R}^k$. For notational simplicity, we drop the variable $l$ and define $m^* = \lceil\!\lceil \frac{m}{k} \rceil\!\rceil$ and $n^* = \lceil\!\lceil \frac{n}{k} \rceil\!\rceil$. The matrix vector multiplication per layer is given as

$$\mathrm{x}_{out} = \mathrm{W}\mathrm{x}_{in} = \begin{bmatrix} \mathrm{W}_{1,1} & \cdots & \mathrm{W}_{1,n^*} \\ \vdots & \ddots & \vdots \\ \mathrm{W}_{m^*,1} & \cdots & \mathrm{W}_{m^*,n^*} \end{bmatrix} \begin{bmatrix} \mathrm{x}_{in,1} \\ \vdots \\ \mathrm{x}_{in,n^*} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{n^*} \mathrm{W}_{1,j}\mathrm{x}_{in,j} \\ \vdots \\ \sum_{j=1}^{n^*} \mathrm{W}_{m^*,j}\mathrm{x}_{in,j} \end{bmatrix}. \tag{2}$$

Next, we represent each weight submatrix $\mathrm{W}_{i,j}$ in a subspace form to reduce the number of trainable parameters. Each weight submatrix is decomposed into the form $\{\mathrm{W}_{i,j} = \mathrm{B}\mathbf{\Sigma}_{i,j}\mathrm{P}\}_{i\in[1,m^*],j\in[1,n^*]}$, where B and P are $k \times k$ unitary matrices and $\mathbf{\Sigma}_{i,j}$ is a $k \times k$ diagonal matrix. Note that this decomposition looks similar to singular value decomposition. However, the main difference is that all the submatrices share the same unitary matrices B and P. As a result, OSENN reduces the number of trainable parameters to the values of the diagonal matrix $\mathbf{\Sigma}_{i,j}$ and equation (2) becomes

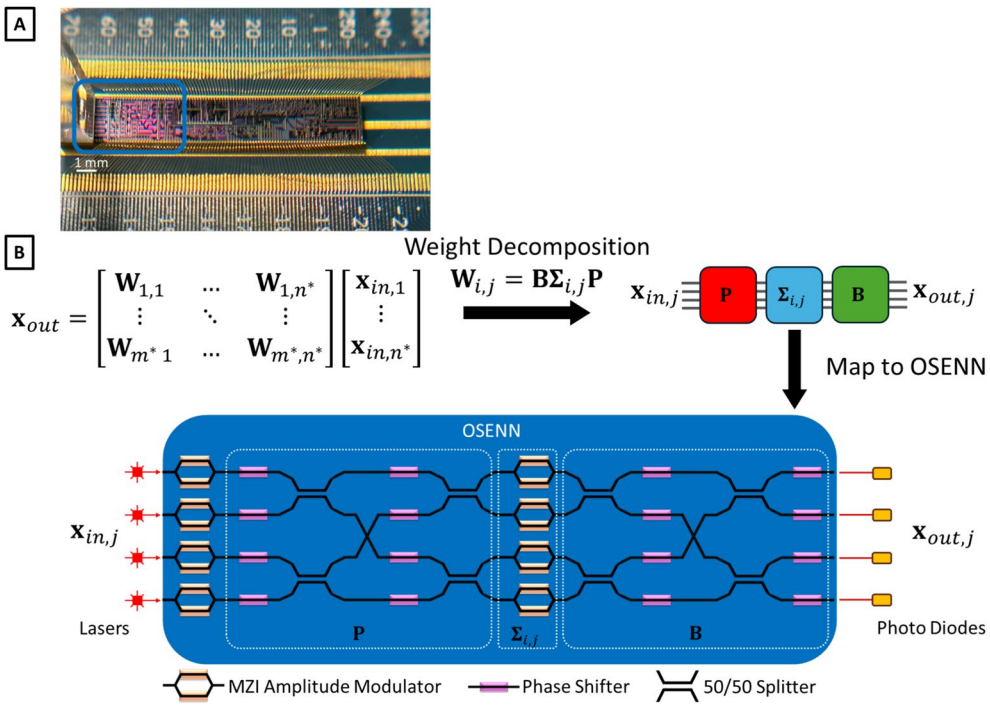


**Fig. 3**. (**a**) A photograph of the edge-based OSENN optical AI chip. (**b**) An overview of the process of mapping the weight matrix W to a $4 \times 4$ OSENN chip and their corresponding optical devices. The unitary matrices P and B are implemented using phase shifters and 50/50 splitters. The diagonal matrix $\mathbf{\Sigma}_{i,j}$ is implemented using Mach-Zehnder interferometer (MZI) amplitude modulators. MZIs are used to adjust the input laser amplitudes to represent $\mathrm{x}_{in,j}$ and photo diodes are used to readout $\mathrm{x}_{out,j}$ values.

$$\mathrm{x}_{out} = \begin{bmatrix} \mathrm{B}\sum_{j=1}^{n*} \mathbf{\Sigma}_{1,j}\mathrm{Px}_{in,j} \\ \vdots \\ \mathrm{B}\sum_{j=1}^{n*} \mathbf{\Sigma}_{m^*,j}\mathrm{Px}_{in,j} \end{bmatrix} \tag{3}$$

such that this matrix segmentation method reduces the number of trainable neural network parameters by up to four times. For example, the 904,902 and 40,255 trainable parameters for the electronic versions of the RPN and FCNN networks, respectively, would be reduced down to approximately 226,226 and 10,064 trainable parameters for the OSENN. Lastly, the $\mathbf{\Sigma}_{i,j}$ parameter values are restricted to a 3 bit weight resolution during training and operation. This weight resolution restriction allows for the OSENN to be designed with further reduced number of optical components and chip area.

## Training strategy

**Deep learning** We use the PyTorch library[42] to build, train and test the electronic neural network models. We use the Pytorch-ONN library[43] to build, train and test the OSENN version of the models. The neural networks are trained on a NVIDIA RTX A3000 GPU.

**Region proposal network (RPN)** Recall that the RPN is designed to detect UCN hits in the input camera image by predicting $8 \times 8$ bounding boxes around detected hits. We use the data generation method discussed previously to generate a dataset of 6500 images of size $416 \times 416$ and the bounding box label files for each image. The dataset is divided into 80% for training, 10% for validation and 10% for testing. During the training phase, the key training parameters are set to 19 epochs, a batch size of 16 and a learning rate of 0.0001 using the Adam optimizer for the electronic network. We define the RPN loss $\mathscr{L}_{RPN}$ as the summation of the localization loss $\mathscr{L}_{loc}$ and confidence loss $\mathscr{L}_{CL}$ as:

$$\begin{aligned} \mathscr{L}_{RPN} &= \mathscr{L}_{loc} + \mathscr{L}_{CL} \\ \mathscr{L}_{loc} &= \sum_{i=1}^{S^2} \mathbb{1}_i^{obj} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + \left(h_i - \hat{h}_i\right)^2 \right] \\ \mathscr{L}_{CL} &= \sum_{i=1}^{S^2} \mathbb{1}_i^{obj} \left(C_i - \hat{C}_i\right)^2 + \sum_{i=1}^{S^2} \mathbb{1}_i^{noobj} \left(C_i - \hat{C}_i\right)^2 \end{aligned} \tag{4}$$

where $[x, y, w, h]$ are the bounding box labels, $\mathbb{1}_i^{obj}$ denotes the indicator function such that $\mathbb{1}_i^{obj} = 1$ if cell *i* is responsible for detecting the UCN and 0 otherwise, $\mathbb{1}_i^{noobj}$ is the complement of $\mathbb{1}_i^{obj}$, $\hat{}$ the predicted value, *S* the grid size ($S = 52$), and $C_i$ is the box confidence score for cell *i*. The localization loss $\mathscr{L}_{loc}$ measures the error of the predicted bounding box if cell *i* contains a UCN hit. The confidence loss $\mathscr{L}_{CL}$ measures the model's confidence in detecting objects in the bounding box.

**Fully connected neural network (FCNN)** Next, we want to predict the subpixel and sub-micron hit position of the detected UCN hit. We generate the FCNN dataset by using the trained RPN network to detect UCNs and extract the $8 \times 8$ single UCN hits. The hit position labels are generated by converting the global hit coordinates to their local ones. We generate approximately 50,000 UCN hit images of size $8 \times 8$ and their corresponding ground-truth labels. The dataset is divided into 60% for training, 20% for validation, and 20% for testing. We use the mean squared error (MSE) loss function to train the FCNN. The key training parameters are set to 80 epochs, a batch size of 175 and a learning rate of 0.001 using the Adam optimizer.

**Optical segmented neural network (OSENN)** We partition the weight matrix into submatrices of size $4 \times 4 (k = 4)$. We also convert the PyTorch neural network model into their Pytorch-ONN versions. The key training parameters used to train the OSENN code is the set to the same values used to train the electronic version except for the number of epochs, which are set to 50 and 120 for the RPN and FCNN, respectively.

# Results and discussion

## Performance metrics

*UCN detection accuracy*

We evaluate the UCN detection accuracy using the Precision, Recall and F1-score performance metrics defined respectively as

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \times 100\% \\ Recall &= \frac{TP}{TP + FN} \times 100\% \\ F1\ score &= \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \end{aligned} \tag{5}$$

where the acronyms refer to true positive (TP), false positive (FP) and false negative (FN). In this work, TP denotes the instances that the RPN correctly detected UCN hits. FP denotes the instances where the RPN predicted a bounding box location that contains no UCN hit. FN denotes the instances where the RPN did not detect a true UCN hit. Note that true negative (TN) does not apply for the task of object detection.

*Spatial resolution accuracy*

During the testing phase, we enable the dropout layers in the FCNN to allow for uncertainty quantification in the label prediction by simulating Monte Carlo runs. We feed each test image $n$ into the trained FCNN 500 times to obtain a mean $\hat{y}_n$ and standard deviation $\hat{\sigma}_n$ for each output label. Therefore, for each test image, we can plot a kernel density estimate (KDE) for the entry and exit positions as shown in Fig. 4b, d. Next, we summarize the performance of the FCNN on the full test dataset consisting of $N$ images and corresponding ground-truth labels by reporting the mean absolute error (MAE), mean absolute percent error (MAPE), mean squared error (MSE), and root mean squared error (RMSE) for each output label. The MAE, MAPE, MSE and RMSE metrics are computed as

$$\begin{aligned} MAE &= \frac{1}{N}\sum_{n=1}^{N}|y_n - \hat{y}_n| \\ MAPE &= \frac{1}{N}\sum_{n=1}^{N}\left|\frac{y_n - \hat{y}_n}{y_n}\right| \times 100\% \\ MSE &= \frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2 \\ RMSE &= \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}_n)^2} \end{aligned} \tag{6}$$

where $y_n$ and $\hat{y}_n$ denotes the ground-truth and predicted value for image $n$, respectively.

## Comparative analysis of hardware performance: Electronic vs. ONN

We generate the training, validation and test datasets following the process described previously in the Data Generation section and train the RPN and FCNN networks following the Training Strategy section. We evaluate the performance of the trained RPN network on the test dataset which contains 650 images of size $416 \times 416$ and a total of 3628 true UCN hits. Similarly, we evaluate the performance of the trained FCNN on the test dataset consisting of 10,000 extracted hit images of size $8 \times 8$ and their corresponding local ground-truth labels. We compare the detection and super position performance between the NVIDIA RTX A3000 GPU and the edge-

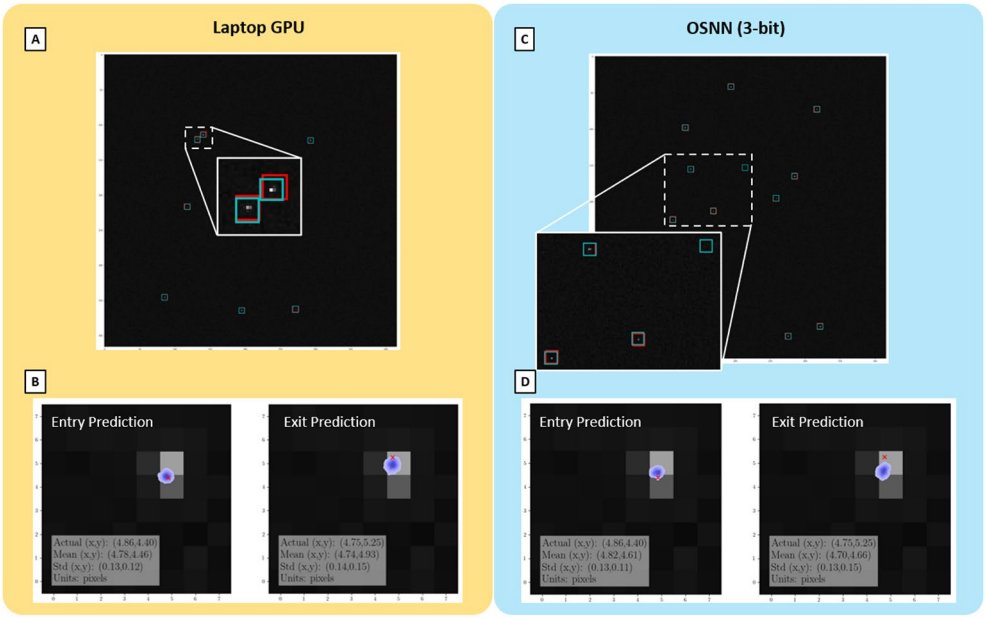


**Fig. 4**. Example prediction results of the RPN and FCNN using a NVIDIA RTX A3000 GPU and the chip-based OSENN with 3-bit weight. (**a**) and (**c**) show the RPN detection results for the NVIDIA RTX A3000 GPU and OSENN, respectively. The ground-truth bounding box is drawn in red and the predicted box is drawn in cyan. In (c), the zoomed-in section shows an example of a false positive prediction in the top right corner. (**b**) and (**d**) show the FCNN entry and exit predictions by the GPU and OSENN, respectively, for the same input hit image. The ground-truth position is denoted by the red 'x' and the Monte Carlo predictions are shown as the blue KDE plot. The prediction for this example shows sub-micron position resolution by both hardware types.

| Hardware | Precision | Recall | F1 Score | FPS |
|---|---|---|---|---|
| NVIDIA RTX A3000 GPU | **99.94%** | 98.57% | **99.25%** | **46.43** |
| OSENN (3-bit) | 96.26% | **99.59%** | 97.90% | 13.12 |

**Table 1**. Summary of UCN detection performance on the test dataset using the RPN. Significant values are in bold.

| Hardware/Method | Output label | MAE | MAPE | MSE | RMSE | Computation time |
|---|---|---|---|---|---|---|
| NVIDIA RTX A3000 GPU | Entry x (pixel) | 0.2132 | 4.994% | 0.0708 | 0.2661 | 5.2315e-5 seconds |
| | Entry y (pixel) | 0.2170 | 4.990% | 0.0716 | 0.2676 | |
| | Exit x (pixel) | 0.2877 | 6.823% | 0.1228 | 0.3504 | |
| | Exit y (pixel) | 0.2837 | 6.594% | 0.1198 | 0.3461 | |
| | $\theta$(degrees) | 5.5989 | 26.932% | 94.911 | 9.7422 | |
| OSENN (3-bit) | Entry x (pixel) | 0.2237 | 5.068% | 0.0754 | 0.2746 | 26.24e-5 seconds |
| | Entry y (pixel) | 0.2196 | 5.318% | 0.0721 | 0.2686 | |
| | Exit x (pixel) | 0.2956 | 6.639% | 0.1261 | 0.3551 | |
| | Exit y (pixel) | 0.3000 | 7.282% | 0.1301 | 0.3607 | |
| | $\theta$(degrees) | 6.3616 | 28.858% | 99.19 | 9.9594 | |
| 2D Gaussian fitting | Entry x (pixel) | 0.2287 | 5.394% | 0.0800 | 0.2829 | 0.0296 seconds |
| | Entry y (pixel) | 0.2312 | 5.335% | 0.0815 | 0.2855 | |
| Center of gravity (COG) | Entry x (pixel) | 0.7101 | 15.44% | 0.7162 | 0.8463 | 1.0659e-5 seconds |
| | Entry y (pixel) | 0.7718 | 16.53% | 0.8101 | 0.9001 | |

**Table 2**. Summary of UCN super position resolution performance using the FCNN and its comparison with traditional methods. The computation time represents the average time to obtain prediction results for one UCN hit image. Note that 1 pixel = 1.67 $\mu$m.

deployed OSENN. The numerical performance results for detection and position resolution are compared in Tables 1 and 2, respectively.

For the task of UCN detection, both the electronic and OSENN achieves very high performance with over 96% for precision, 98% for recall and 97% for F1 score. Recall that the OSENN uses 3-bit weight resolution, but it can achieve comparable performance to the NVIDIA RTX A3000 GPU in precision and better performance for recall (99.59%). Figure 4a, c show an example UCN detection result by the NVIDIA RTX A3000 GPU and OSENN, respectively, where the ground-truth bounding box is drawn in red and the predicted in cyan. As mentioned in the Data Generation section, the predicted bounding box may not exactly match the ground-truth but will capture the UCN hit. The speed of the trained RPN model to process UCN hit images is 46.43 frames per second (FPS) for the NVIDIA RTX A3000 GPU and approximately 13.12 FPS for the OSENN assuming a modulation speed of 10 GHz. We note that both the NVIDIA RTX A3000 GPU and OSENN computation speed is done for images in series. Furthermore, we note that we approximate the OSENN computational speed as the PyTorch-ONN model simulates the response of the actual OSENN chip while training for the weights $\Sigma_{i,j}$ to be deployed to the actual chip. The OSENN computational speed is dependent on the modulation speed such that the compute time is approximated as the total number of submatrix-vector computations divided by the modulation frequency. While the OSENN FPS is slower than the NVIDIA RTX A3000 GPU, it can be integrated with the CMOS image sensor as a compact edge-device with very low power consumption.

For the task of UCN super position resolution, both the electronic and OSENN models achieves sub-pixel position resolution and sub-micron position resolution. The worst MAE metric is for the OSENN in predicting the Exit y coordinate with an MAE value of 0.3 pixels or 0.501 $\mu$m, as tabulated in Table 2. Note that one pixel is equivalent to 1.67 $\mu$m. Figure 4b, d shows an example prediction by the NVIDIA RTX A3000 GPU and OSENN, respectively, for same input hit image. The ground truth entry and exit positions are marked by a red 'x' and the blue KDE shows the FCNN Monte Carlo predictions. In this example, the OSENN prediction for the exit position is slightly worse than the NVIDIA RTX A3000 GPU but still within our goal of sub-micron position resolution. We also compare our method with the traditional methods of 2D Gaussian fitting and the center of gravity (COG) in predicting the hit entry position and the computation time in Table 2. Our proposed method achieves the most accurate entry position resolution, while also being able to predict the exit position and the incident angle $\theta$. Table 2 tabulates the computation time to obtain results for a singular UCN hit image. We note that the computation time for our method represents the time to compute 500 Monte Carlo runs and thus, will be much faster if the FCNN dropout layers are disabled during inference to predict one result. Meanwhile, the 2D Gaussian fitting is the slowest as the fitting is an iterative process. In conclusion, our method achieves improved position resolution with fast computation speed in comparison to traditional methods.

## Conclusion

This work proposed a two-stage framework for UCN detection and super position resolution. The detection network uses a RPN to predict bounding boxes to detect and extract UCN hits from input hit images captured by the CMOS image sensor. The extracted hits are then fed into the FCNN to predict the entry and exit coordinates for super position resolution. To train the neural networks, we introduce a data generation method that produces experiment like synthetic data using Allpix Squared. Lastly, we train and test the performance of the two-stage framework using a NVIDIA RTX A3000 GPU and an edge-deployed OSENN. The performance results show that our proposed framework can achieve both high UCN detection accuracy and sub-micron position resolution. While our framework has been applied to UCNs, we note that it can be applied to general neutron detection applications of varying neutron energies including thermal and cold neutrons. In addition, it can be applied to image sensors of any size by processing the the image in patches, with computational time dependent on the number of patches.

Future work includes designing the integrated hardware architecture between the edge-deployed OSENN and the CMOS image sensor for real-time hardware testing. Current experiments use a CMOS image sensor that captures UCN videos with a frame rate of 3 FPS and an image size of $3872 \times 2764$ pixels, which results in approximately 70 image patches of size $416 \times 416$. We can increase the OSENN computation speed in several ways including but not limited to using multiple OSENN chips, increasing the OSENN submatrix size, using a more aggressive modulation speed and switching from thermal-optical turning to electro-optical tuning. Additional future directions following the results from this work include further studies of the radiation hardness of the OSENN and extending the detection network to include the classification of the charged particles $^{4}$He and $^{7}$Li.

## Data availability

The datasets used in this work are available from the corresponding authors upon reasonable request.

Received: 22 September 2025; Accepted: 17 November 2025
Published online: 27 November 2025

## References

1. Knoll, G. F. *Radiation detection and measurement* (John & Wiley Sons Inc, 2010).
2. Klett, A. *Neutron Detection* 759–790 (Springer, Berlin Heidelberg, Berlin, Heidelberg, 2012). https://doi.org/10.1007/978-3-642-13271-1_31.
3. Strobl, M. et al. Advances in neutron radiography and tomography. *J. Phys. D Appl. Phys.* **42**, 243001 (2009).
4. Podurets, K. et al. Modern methods of neutron radiography and tomography in studies of the internal structure of objects. *Crystallogr. Rep.* **66**, 254–266 (2021).
5. Ziesche, R. F., Kardjilov, N., Kockelmann, W., Brett, D. J. & Shearing, P. R. Neutron imaging of lithium batteries. *Joule* **6**, 35–52 (2022).
6. Perfect, E. et al. Neutron imaging of hydrogen-rich fluids in geomaterials and engineered porous media: A review. *Earth Sci. Rev.* **129**, 120–135 (2014).
7. Al Hamrashdi, H., Monk, S. D. & Cheneler, D. Passive gamma-ray and neutron imaging systems for national security and nuclear non-proliferation in controlled and uncontrolled detection areas: Review of past and current status. *Sensors* **19**, 2638 (2019).
8. Bernstein, A. et al. Colloquium: Neutrino detectors as tools for nuclear security. *Rev. Mod. Phys.* **92**, 011003 (2020).
9. Ericsson, G. Advanced neutron spectroscopy in fusion research. *J. Fusion Energy* **38**, 330–355 (2019).
10. Dubbers, D. & Schmidt, M. G. The neutron and its role in cosmology and particle physics. *Rev. Mod. Phys.* **83**, 1111–1171 (2011).
11. Swinhoe, M. T. et al. A new generation of uranium coincidence fast neutron collars for assay of lwr fresh fuel assemblies. *Nucl. Instrum. Methods Phys. Res., Sect. A* **1009**, 165453 (2021).
12. Kirch, K., Lauss, B., Schmidt-Wellenburg, P. & Zsigmond, G. Ultracold neutrons–physics and production. *Nucl. Phys. News* **20**, 17–23 (2010).
13. Musedinovic, R. et al. Measurement of the free neutron lifetime in a magneto-gravitational trap with in situ detection. *Phys. Rev. C* **111**, 045501 (2025).
14. Broussard, L. & Collaboration, U. Ucnb: The neutrino asymmetry in polarized ultracold neutron decay. In *AIP Conference Proceedings*, vol. 1560, 149–151 (American Institute of Physics, 2013).
15. Brown, M.-P. et al. New result for the neutron $\beta$-asymmetry parameter $a_0$ from ucna. *Phys. Rev. C* **97**, 035505 (2018).
16. Saunders, A. et al. Performance of the los alamos national laboratory spallation-driven solid-deuterium ultra-cold neutron source. *Rev. Sci. Instrum.***84** (2013).
17. Abel, C. et al. Measurement of the permanent electric dipole moment of the neutron. *Phys. Rev. Lett.* **124**, 081803 (2020).
18. Ichikawa, G. et al. Observation of the spatial distribution of gravitationally bound quantum states of ultracold neutrons and its derivation using the wigner function. *Phys. Rev. Lett.* **112**, 071101 (2014).
19. Wei, W. et al. Position-sensitive detection of ultracold neutrons with an imaging camera and its implications to spectroscopy. *Nucl. Instrum. Methods Phys. Res., Sect. A* **830**, 36–43 (2016).
20. Pietropaolo, A. et al. Neutron detection techniques from $\mu$ev to gev. *Phys. Rep.* **875**, 1–65 (2020).
21. Wang, Z. et al. A multilayer surface detector for ultracold neutrons. *Nucl. Instrum. Methods Phys. Res., Sect. A* **798**, 30–35 (2015).
22. Morris, C. L. et al. A new method for measuring the neutron lifetime using an in situ neutron detector. *Rev. Sci. Instrum.***88** (2017).
23. Kuk, K. et al. Projection imaging with ultracold neutrons. *Nucl. Instrum. Methods Phys. Res., Sect. A* **1003**, 165306 (2021).
24. Farhadi, A. & Redmon, J. Yolov3: An incremental improvement. In *Computer vision and pattern recognition*, vol. 1804, 1–6 (Springer Berlin/Heidelberg, Germany, 2018).
25. Lin, S. et al. Demonstration of sub-micron ucn position resolution using room-temperature cmos sensor. *Nucl. Instrum. Methods Phys. Res., Sect. A* **1057**, 168769 (2023).
26. Lin, S. et al. Neural network methods for radiation detectors and imaging. *Front. Phys.* **12**, 1334298 (2024).
27. Prinzie, J., Simanjuntak, F. M., Leroux, P. & Prodromakis, T. Low-power electronic technologies for harsh radiation environments. *Nat. Electron.* **4**, 243–253 (2021).
28. Khanna, V. K. *Extreme-Temperature and Harsh-Environment Electronics: Physics, Technology and Applications* (IOP Publishing, 2023).
29. Feng, C. et al. Integrated photonics for computing and artificial intelligence. In *2023 IEEE Photonics Society Summer Topicals Meeting Series (SUM)*, 1–2 (IEEE, 2023).

30. Gu, J., Feng, C., Zhu, H., Chen, R. T. & Pan, D. Z. Light in ai: toward efficient neurocomputing with optical neural networks–A tutorial. *IEEE Trans. Circuits Syst. II Express Br.* **69**, 2581–2585 (2022).
31. Lalović, M. et al. Ionizing radiation effects in silicon photonics modulators. *IEEE Trans. Nucl. Sci.* **69**, 1521–1526 (2022).
32. Spannagel, S. et al. Allpix2: A modular simulation framework for silicon detectors. *Nucl. Instrum. Methods Phys. Res., Sect. A* **901**, 164–172 (2018).
33. Ren, S., He, K., Girshick, R. & Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.***28** (2015).
34. Liu, W. et al. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 21–37 (Springer, 2016).
35. Dai, J., Li, Y., He, K. & Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.***29** (2016).
36. Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988 (2017).
37. Sevilla, J. et al. Compute trends across three eras of machine learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–8 (2022).
38. Shen, Y. et al. Deep learning with coherent nanophotonic circuits. *Nat. Photon.* **11**, 441–446 (2017).
39. Shastri, B. J. et al. Photonics for artificial intelligence and neuromorphic computing. *Nat. Photon.* **15**, 102–114 (2021).
40. Feng, C. et al. A compact butterfly-style silicon photonic-electronic neural chip for hardware-efficient deep learning. *ACS Photon.* **9**, 3906–3916 (2022).
41. Ning, S. et al. Hardware-efficient photonic tensor core: Accelerating deep neural networks with structured compression. *Optica* **12**, 1079–1089 (2025).
42. Paszke, A. et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.***32** (2019).
43. Gu, J. et al. L2ight: Enabling on-chip learning for optical neural networks via efficient in-situ subspace optimization. In *Conference on Neural Information Processing Systems (NeurIPS)* (2021).

## Author contributions

S.L.: Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing; H.Z.: Software; S.C.: Writing - review & editing; C.L.M.: Writing - review & editing; D.P.: Writing - review & editing; Z.T.: Writing - review & editing; R.T.C.: Conceptualization, Writing - review & editing; Z.W.: Conceptualization, Writing - review & editing

## Funding

This work was supported in part by the U.S. Department of Energy (Los Alamos Report Number LA-UR-25-22828) under the Contract 89233218CNA000001, the Multidisciplinary University Research Initiative (FA9550-17-1-0071) and the Air Force Office of Scientific Research (FA9550-23-1-0452).

## Declarations

### Competing interests

The authors declare that they have no competing interests or personal relationships that could have appeared to influence the work reported in this paper.

### Correspondence

Questions regarding OSENN should be addressed to R.T.C.; other questions should be addressed to Z.W.

## Additional information

**Correspondence** and requests for materials should be addressed to R.T.C. or Z.W.

**Reprints and permissions information** is available at www.nature.com/reprints.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.