

# Online Learning for Dynamic Structural Characterization in Electron Energy Loss Spectroscopy

M.L.Varshika<sup>†</sup>, Jonathan Hollenbach<sup>\*</sup>, Nicolas Agostini<sup>‡</sup>, Ankur Limaye<sup>‡</sup>, Antonino Tumeo<sup>‡</sup>, and Anup Das<sup>†</sup>

<sup>†</sup>*Drexel University*, <sup>\*</sup>*Johns Hopkins University*, <sup>‡</sup>*Pacific Northwest National Laboratory*

Corresponding Email: lm3486@drexel.edu

**Abstract**—In-situ Electron Energy Loss Spectroscopy (EELS) is a crucial technique for determining the elemental composition of materials through EELS Spectrum Images (EELS-SI). While recent innovations have made it possible for EELS-SI data acquisition at rates of 400 frames per second with near-zero read noise, the challenge lies in processing this massive stream of real-time data to capture nanoscale dynamic changes. This task demands advanced machine learning methods capable of identifying subtle and complex features in EELS spectra. Furthermore, the EELS data acquired in difficult experimental conditions often suffer from a low signal-to-noise ratio (SNR), leading to unreliable classification and limiting their utility. In response to this critical need, we introduce a spiking neural network (SNN)-based Variational Autoencoder (VAE) that embeds spectral data into a latent space, facilitating precise prediction of structural changes. VAEs are designed to learn efficient low-dimensional representations while capturing the inherent variability in the data, making them highly effective for processing multi-dimensional data. Additionally, SNNs, which use biological neurons, offer unmatched scalability and energy efficiency by processing information through binary spikes, making them ideal for high-throughput data. We validate our framework using MXene annealing data, achieving denoised spectrum images with an SNR of 28.3dB. For the first time, we present a fully online learning solution for dynamic structural tracking, implemented directly in hardware, eliminating the traditional bottleneck of offline training. Our method achieves reliable, real-time, on-device characterization of high-speed EELS data when evaluated on an FPGA platform. Joint experiments with the SNN-VAE model on both spiking autoencoder hardware and a software-trained hybrid configuration of hardware spiking encoders demonstrated latency reductions of  $25.2\times$ ,  $93.7\times$ , and  $1.04\times$ ,  $4.5\times$  in energy savings, respectively, compared to baseline.

## I. INTRODUCTION

Recent advances in electron imaging and spectroscopy have revolutionized our ability to probe materials at atomic and subatomic scales, using techniques such as scanning probe microscopy [1], [2], atom probe tomography [3], and transmission electron microscopy (TEM) [4]–[6] to determine the precise local structure of materials. Among these, In-Situ Electron Energy Loss Spectroscopy (EELS) stands out as a powerful analytical technique integrated into scanning transmission electron microscopy (STEM), where a focused energy beam interacts with the sample through elastic or inelastic collisions to generate high-contrast EELS spectrum images. The resulting spectra, particularly those with core-loss in energy, reveal crucial element-specific ionization events, offering profound insights into the composition and properties of the material [7]–[9].

The fine structure of EELS spectra is exceptionally rich in chemical information, allowing the identification of subtle variations such as oxidation states and bonding configurations [9]. Despite this, current practices still rely heavily on manual visual inspection for element identification [7]. The capability of in-situ EELS to generate real-time spectrum images while capturing highly spatial structural measurements at data acquisition rates exceeding 400 spectra per second [10]–[12] presents an extraordinary opportunity to advance material characterization. However, there are several key challenges.

i) **High Dimensionality:** EELS spectra contain vast amounts of detailed information across a broad range of energy loss values, making the analysis of such high-dimensional data inherently complex. ii) **Low Signal-to-Noise Ratio (SNR):** Under many experimental conditions, particularly at high resolutions, EELS spectra suffer from a low SNR [9], [13], [14], which obscures critical spectral features, complicating the classification and interpretation of data. Conventional methods fail to extract meaningful information from low-SNR data, leading to inaccurate classification and the omission of important details in dynamic processes. iii) **Dynamic Processes:** Capturing the temporal evolution at rapid data acquisition [11], which rely on offline batch processing, are fundamentally inadequate for dynamic processes requiring immediate feedback and real-time decision making.

In response to these challenges, we propose a novel framework combining Variational Autoencoders (VAEs) for capturing essential features in low-dimensional latent spaces and Spiking Neural Networks (SNNs) for energy-efficient, real-time data processing. This synergy leverages VAEs’ denoising capabilities [11] and SNNs’ scalability [15]–[18] to address challenges in high-dimensional spectral data analysis.

We introduce an online learning framework for dynamic structural tracking, implemented directly in hardware on a Field-Programmable Gate Array (FPGA) platform. Our approach eliminates the need for traditional offline training methods, which are resource-intensive and time-consuming, and enables real-time, on-device characterization of high-speed EELS data. Our results enable atomic-scale precision in material synthesis and characterization, accelerating the discovery of novel materials.

The following are our key **contributions**.

- We present the first-ever Spiking Neural Network Variational Autoencoder (SNN-VAE) specifically designed for real-time structural characterization of EELS Spec-

trum Images (EELS-SI). Our model uniquely incorporates spiking convolutional layers in both the encoder and decoder, drastically reducing reliance on bulky matrix operations while achieving superior SNR compared to the state-of-the-art methods [10], [11].

- We introduce a hybrid SNN-VAE framework for on-device characterization, leveraging pre-trained parameters for seamless hardware deployment. This framework optimizes the encoder architecture for hardware platforms, comprehensively evaluating critical performance metrics such as latency, power efficiency, and resource utilization.
- We present an SNN-VAE hardware design with fully integrated online learning capabilities, enabling efficient pipelining across multiple layers within the autoencoder.

We evaluate the performance of our designs in terms of spectrum reconstruction quality and latency and energy consumption of the hardware implementation.

## II. BACKGROUND AND RELATED WORKS

### A. Machine Learning Approaches

Advanced spectroscopy techniques, like TEM, effectively analyze materials' electronic structure, providing ultra-high spatial and temporal resolution data. Recently, machine learning (ML) techniques are used for analyzing complex, high-dimensional data in various domains, including image processing, natural language processing, and materials science.

For instance, PCA has been applied to XEDS spectrum images for noise reduction by linearly projecting data onto orthogonal vectors [19]. However, PCA's bias during decomposition reduces accuracy at low SNR, making it unsuitable for complex, non-linear patterns in EELS spectra, which require more robust machine learning models [20].

Dimensionality reduction techniques like matrix factorization effectively denoise large datasets by reconstructing them in smaller latent spaces. Non-negative matrix factorization (NMF) was used to reduce EELS spectrum dimensions [21], but its reliance on pre-determined hyperparameters limits flexibility in unsupervised learning [22]. Given the high throughput of modern EELS systems, machine learning algorithms must be both scalable and computationally efficient to process large volumes of data in real-time [23]–[25]. While convolutional neural networks have shown promise in tasks like metal species analysis [25], their high data and resource requirements make them less suitable for unsupervised learning and low-latency applications.

Conversely, Variational Autoencoders (VAEs) have demonstrated effectiveness in denoising by mapping spectrum images to non-linear latent embeddings [11], [26]. In [27], VAEs were employed to examine nanocrystal arrays via EELS spectrum images to determine their geometric properties. However, these methods rely on offline characterization, which is incompatible with the immediate post-processing required for real-time analysis.

### B. Hardware Design Space

An FPGA-based inference VAE is implemented in [28], where the autoencoder employs a deep neural network (DNN)

encoder-decoder architecture along with a Gaussian distribution in the latent space. The Gaussian sampling layer generates the mean and variance parameters from the latent vector, reducing data transfer between the CPU and memory blocks. However, traditional deep learning models like this are computationally intensive and consume significant energy, which limits their efficiency in resource-constrained environments.

In [29], a hybrid spiking VAE is implemented on Loihi neuromorphic hardware. This hybrid architecture utilizes a CNN-based decoder trained on the DVSGesture dataset, while a spiking neural network (SNN) encoder maps variations in event data into the latent space. Despite its efficiency, the hybrid autoencoder design does not support online learning, restricting its adaptability to real-time applications. We introduce a fully end-to-end trainable SNN-VAE, implemented on an FPGA platform. Our model achieves low-latency performance while maintaining the energy efficiency required for real-time, high-speed data processing.

## III. SNN-VAE MODEL FOR REAL-TIME EELS DATA PROCESSING

We describe the architecture, key components, and how the model enables noise-resilient analysis of complex spectral data in this section.

### A. Variational Autoencoder (VAE)

The VAE is a generative model that encodes input data  $x$  in a high-dimensional space  $N$  into a lower-dimensional latent space  $z$  by learning a probability distribution  $q_\phi$  over the latent variables. The encoder outputs a distribution over the latent space, typically parameterized as a Gaussian distribution with mean  $\mu$  and variance  $\sigma^2$ .

$$q_\phi(z|x) = \text{Gaussian}(z, \mu, \sigma^2) \quad (1)$$

$$p_\phi(x|z) = \text{Gaussian}(\hat{x}, \mu_\theta, \sigma_\theta^2) \quad (2)$$

Equation 1 describes the probability distribution function of an encoder with fixed parameter  $\phi$ . The decoder maps the latent variable  $z$  back to reconstruct  $\hat{x}$  shown in Equation 2. Re-parameterized latent space has  $\mu_\theta$ , and  $\sigma_\theta^2$  as mean and variance at  $\theta$  parametric condition, with  $p_\phi(x|z)$  being the probability of constructing  $x$  from the latent variable  $z$ .

A latent variable helps the model decide the next character to generate. The model trained on a sample assigns a latent variable  $z$ . For every input, the model compares  $z$  for every image feature. A loss function measures the difference between the decoded sample and the original image. VAEs use prior distributions to learn robust, smooth latent representations that capture key data variations.

### B. Spiking Neural Networks

Spiking Neural Networks (SNNs) are event-driven neural networks, where neurons communicate by transmitting discrete spikes in response to specific events. Unlike traditional neural networks, which process information continuously, SNNs operate by accumulating spikes over time, firing only after a certain threshold is reached. We employ Leaky-Integrate-and-Fire (LIF) neurons, which are modeled by

$$v(t+1) = v(t) + S_i(t+1) \quad (3)$$

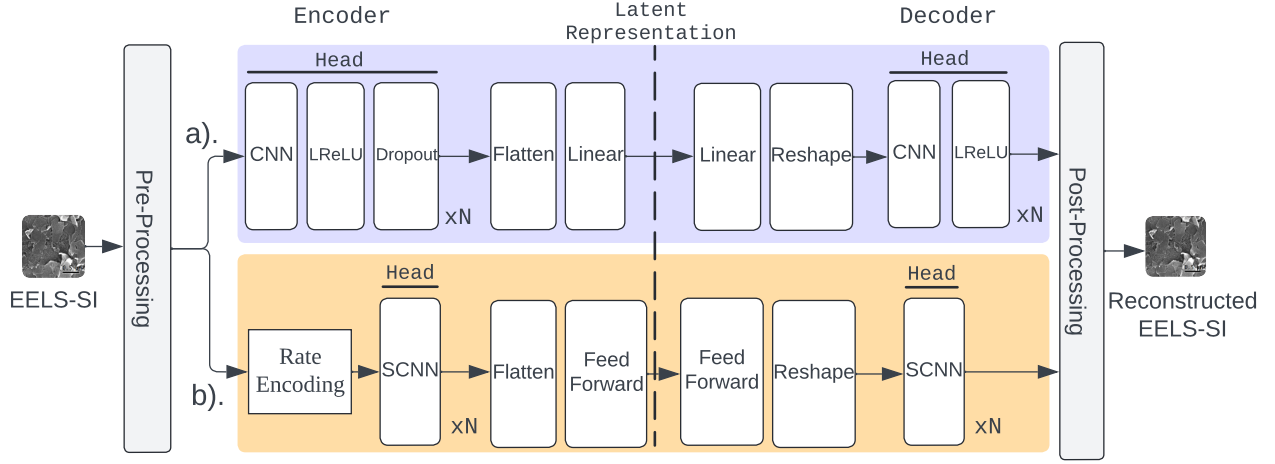


Fig. 1: Internal architecture of VAE with  $N$  heads. a) Baseline CNN-VAE model with convolution (CNN), Leaky ReLU (LReLU), and Unflatten layer (Reshape). b) SNN-VAE model with spiking convolution layer (SCNN) and spiking feed forward layer (Feed Forward).

$$NO_i(t) = \begin{cases} 1 & v(t) \geq V_{th} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

When a presynaptic neuron generates an input current  $S_i(t)$ , it causes the membrane potential  $v(t)$  of the postsynaptic neuron to increase by a value proportional to the synaptic weight  $w$  with a gradual leak. If the accumulated membrane potential exceeds a predefined threshold  $V_{th}$ , the neuron fires, producing an output spike  $NO_i$  to reset the potential.

### C. VAE Architecture

The baseline CNN-VAE model is built on the RapidEELS framework [11]. Figure 1b illustrates the SNN-VAE model, which we describe below.

1) *Pre-Processing*: EELS-SI digital micrograph 4 (DM4) spectrum images are imported with HyperSpy [30] to spectrum data. The input dataset to the autoencoder is limited to localized information, without incorporating spatial information, as done in [10].

2) *Encoder*: The baseline encoder consists of  $N$  heads, each composed of a series of CNN, Leaky ReLU (LReLU), and Dropout layers. A convolutional layer is followed by a Leaky ReLU instead of a pooling layer to retain finer features. A dropout layer with a 20% rate is used, as shown in [11], to prevent overfitting. Each successive head contains fewer neurons and larger filter sizes than the previous one, ensuring dimensionality reduction at each stage.

Our SNN-VAE encoder extracts latent variables from the rate-encoded input dataset. The stream of spikes is passed through  $N$  heads of spiking convolutional layers (SCNN). Since the SCNN output consists of spikes, a normalization layer is unnecessary. The sparse nature of the rate-encoded input compensates for the need for a dropout layer.

3) *Latent space*: The features extracted by the encoder are embedded into a latent space. In the baseline design, the latent space is appended with a linear classifier layer trained on the same dataset as the VAE. A similar approach is used for the SNN-VAE, where a spiking feed-forward layer

at the end of the encoder assists in classifying the latent representations. Additionally, an auxiliary classifier [10] is employed to provide a low-latency classification suitable for hardware implementation.

4) *Decoder*: The decoder is a mirror architecture of the encoder. It consists of  $N$  heads of CNN and LReLU layers in the baseline model, and SCNN layers in the SNN-VAE model. Each head progressively increases the number of neurons, gradually reconstructing the image. The spectrum data is then passed to *post-processing* to rebuild the spectrum images.

Both VAEs use Kullback-Leibler (KL) divergence [31] loss to measure the divergence between the learned latent distribution and the Gaussian distribution.

## IV. FPGA-ENABLED SYSTEM DESIGN

In this section, we introduce our two different approaches for SNN-VAE implementation on FPGA hardware. Our design is inspired by the layer-based architecture in [32]–[34].

### A. Fully Integrated SNN-VAE Architecture

Figure 2 illustrates our design, which we describe below. The architecture consists of two primary components: the *Inference Engine* and the *Learning Engine*.

The *Inference Engine* is responsible for dimensionality reduction in the autoencoder. The spiking convolution layers (SCNN) of the encoder and feed-forward layer of SNN-VAE are mapped as the *Encoder* and *Linear* layers, respectively. Spectrum data is first converted to spike-based Address Event Representation (AER) input. *AER Interpreter* decodes events to corresponding spikes for their respective neurons. Read Data (RD) blocks of the *Encoder* extract the weight value of a neuron receiving an input spike from the *Weight memory*.

For a convolution layer with  $n$  neurons and  $F$  filters,  $n \times F$  convolution units (CU) and leaky-integrate-and-fire (LIF) neurons are initialized.

A CU and LIF neuron together form the core operation of an SCNN layer in the *Encoder*. As shown in Figure 3a, CU performs convolution of kernel weights  $K_i$  with input.

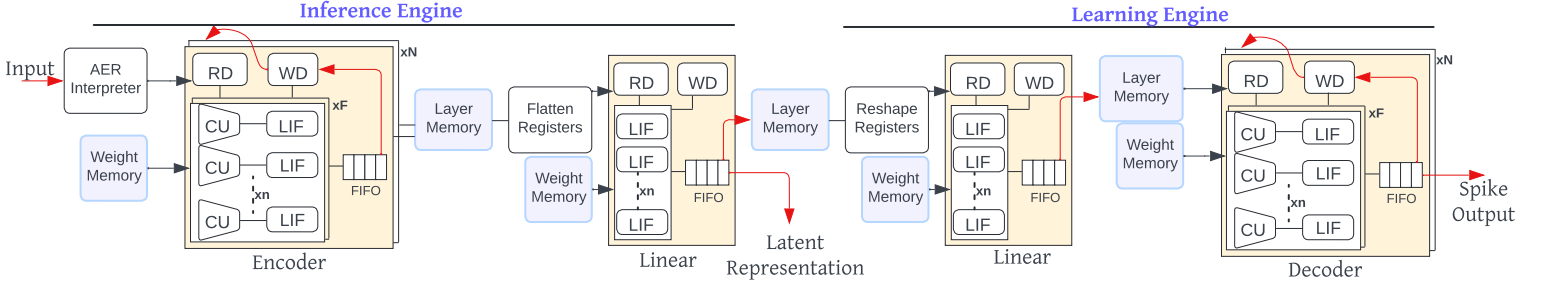


Fig. 2: System design of SNN-VAE model where convolution unit (CU), leaky-integrate-and-fire (LIF) neuron, read data (RD), write data (WD), and first-in-first-out (FIFO) buffer.  $n, F, N$  are number of neurons, number of filters, and number of heads.

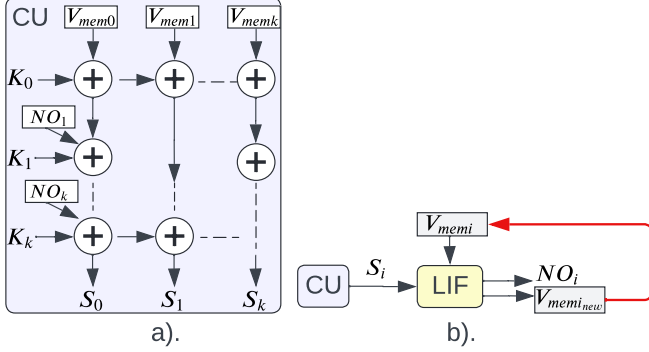


Fig. 3: a) Detailed design of convolution unit (CU) in SNN-VAE. b) Data flow in one filter of SCNN.

Since the input is a binary spike, the convolution operation is executed using a series of adders. Each neuron in a convolution layer has  $k$  neighboring synaptic connections, where  $k$  is the size of the kernel. The membrane potential  $V_{memi}$  of a postsynaptic neuron is integrated with  $k$  synaptic weights, provided presynaptic neuron has spiked. A series of  $k$  adders generates the convoluted output  $S_i$  of neuron  $i$ . One CU performs  $k$  weighted convolutions simultaneously in Algorithm 1.

Three levels of parallelism are employed in Algorithm 1. First, each neuron in an SCNN or linear layer computes the CU-LIF unit in parallel. To enable this, the kernel weight array is partitioned into four blocks to avoid memory read violations. Second, all the filters of an SCNN layer operate in parallel, writing the outputs to a FIFO memory. Finally, the outputs are sent via Write Data (WD) to the head of the next encoder in the pipeline.

Additionally, consecutive CUs are only dependent on access to the  $V_{memi}$  register, allowing CUs to be fully pipelined with a delay of just one register write clock cycle. The output  $S_i$  is sent to the LIF neuron to compute the output spike  $NO_i$  of a neuron as seen in Figure 3b. The new membrane potential  $V_{memi\_new}$ , after computation, is written back to the register  $V_{memi}$ .

The output of the SCNN layer is stored in a FIFO buffer and the *Layer Memory*. The FIFO caches the output for immediate access, while the WD blocks of the *Encoder* store refractory period, membrane potential, and layer output from the FIFO in registers before passing them to the next stage of the *Encoder*. A flatten operation reshapes the output vector, which

#### Algorithm 1: HLS algorithm of Inference Engine

```

Input: AER
Output: spike_output
1  ref, mem = 0          /* Initialize refractory, membrane */
2  W = RAM(K)            /* Read kernel weights */
3  #pragma HLS pipeline
4  for  $n_i \in n$  do      /* For each neuron in input layer */
5      spike_input(AER) = 1 /* AER Interpreter */
6  #pragma HLS pipeline
7  while  $h \neq \text{Head}$  do /* Run for all SCNN heads of the
   encoder */
8      #pragma HLS pipeline
9      for  $f_i \in F$  do /* For each filter of SCNN in  $F$  */
10         #pragma HLS pipeline
11         for  $n_i \in n$  do /* For each neuron in SCNN layer */
12             #pragma HLS array partition K block 4
13              $S_i = \text{CU}(K, \text{mem}, \text{spike\_input})$  /* Compute
               convoluted output */
14              $NO_i = \text{LIF}(\text{ref}, \text{mem}, S_i)$  /* Compute neuron
               output */
15             BRAM write  $NO_i, \text{ref}, \text{mem}$ .
16         for  $n_i \in n$  do /* For each neuron in Linear layer */
17             spike_output =  $\text{LIF}(\text{ref}, \text{mem}, NO_i)$  /* Compute LIF
               neuron output */
18             BRAM write spike_output, ref, mem.
19

```

is implemented using registers and LUTs (*Flatten Registers*). The final layer in the *Inference Engine* is a feed-forward classifier *Linear* layer that generates the latent representation. This *Linear* layer functions similarly to the SCNN layer, but without the CU unit. Spike inputs and the weight matrix, read by RD, are processed in parallel by  $n$  LIF neurons, whose outputs are recorded in the FIFO. During testing, the latent representation is used to classify images.

The second component of our design is the *Learning Engine*, which reconstructs denoised spectrum data. It reads the latent representation from the *Layer memory* and reshapes it to a vector for further computation. The *Learning Engine* mirrors the architecture of the *Inference Engine*, with a *Linear* layer followed by  $N$  convolution transpose layers in the *Decoder*. The *Decoder* also monitors the loss of spike outputs during training.

#### B. Hybrid Framework

We developed a co-design framework in which only the *Inference Engine* is implemented on chip. Figure 4 shows our hybrid framework, which is designed using PyTorch [35] and High-Level Synthesis (HLS) platforms.

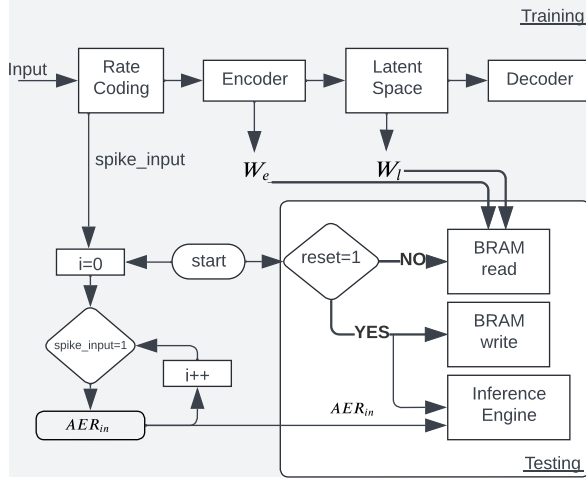


Fig. 4: The proposed software interface used to design hybrid architecture. The software passes input to on-chip *Inference Engine*.

The Latent space in Figure 4 corresponds to the flatten, reshape and feed-forward layers of the SNN-VAE. Rate-coded training data is run through the SNN-VAE Pytorch model to extract the trained weight parameters  $W_e$  of all the encoder layers and  $W_l$  of the Latent space. These trained parameters, along with spike input, are passed to the hardware-based *Inference Engine*. Prior to this, the spike input is converted to its AER ( $AER_{in}$ ) value as shown in the flowchart of Figure 4.

The weight parameters are stored in BRAM at the start of the inference, and once reset, the *Inference Engine* becomes ready for testing.

## V. EXPERIMENTAL SETUP AND RESULTS

### A. Experimental Setup

To validate the framework outlined above, an in-situ EELS-SI dataset [10] was employed, focusing on three distinct scan regions of  $Cr_2TiC_2O_xF_y$  subjected to annealing at temperatures up to 800°C. The dataset encompasses 24 EELS-SI acquisitions, comprising 570 individual spectra. The input spectra are averaged over energy, as shown in Figure 5, and normalized to align  $Cr, F, O, Ti$  edges.

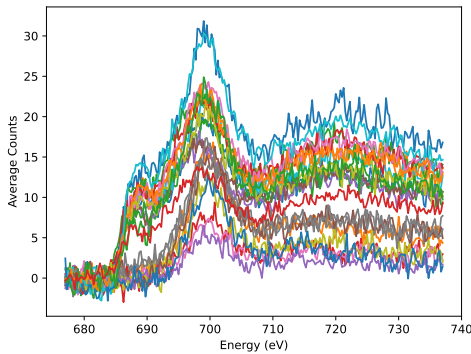


Fig. 5: Spectra plot of 24 EELS-SI averaged in energy range of 680eV-740eV.

### B. Reconstruction Quality Metrics

SNN-VAE design is synthesized at 100 MHz using Xilinx Vitis High-level Synthesis (HLS) in conjunction with Vivado to obtain area and power metrics. We utilized the PyTorch libraries for the SNN-VAE implementation and the system software interface within the hybrid model. We use 4 heads for spectrum size of 240 and a latent dimension of 16.

Our design is evaluated against a baseline CNN-VAE model [11], as illustrated in Figure 1. Figure 6 presents the reconstructed spectrum from both the baseline and our SNN-VAE for one of the spectrum images. Notably, the denoised spectra from both designs exhibit lower signal intensity compared to the original spectrum. This reduction is attributed to the reconstruction process, where the latent representation of the autoencoder effectively eliminates background noise, facilitating improved feature extraction.

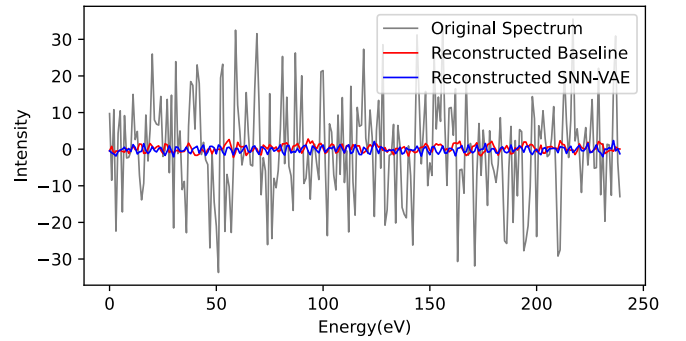


Fig. 6: Comparison of the reconstructed spectrum from the baseline model and the SNN-VAE with the original input spectrum.

The reconstructed spectrum is mapped to a two-dimensional image using HyperSpy [30] and Matplotlib [36], as shown in Figure 7. The EELS-SI for two samples annealed at temperatures of 300°C and 400°C is presented in Figures 7a and 7b, respectively. These images are processed using HyperSpy functions to generate the corresponding spectral data for evaluation. The output of the reconstructed and denoised spectra from our SNN-VAE is illustrated in Figures 7c and 7d.

Figure 8 presents the autoencoder spectrum and the latent representation from the *Inference Engine* for one sample. As shown in Figure 8a, the SNN-VAE and baseline spectra closely track each other across the plotted range. However, the latent representations of the SNN-VAE and the baseline, depicted in Figure 8b, do not exhibit concordance. The Mean Squared Error (MSE) was calculated for the three models to evaluate their respective deviations from the actual data. The baseline model demonstrated an MSE of 1.011, while the SNN-VAE exhibited an MSE of 0.75. Similarly, the baseline latent representation had an MSE of 14.2, compared to the *Inference Engine*'s MSE of 5.4. The lower MSE values for the SNN-VAE and the *Inference Engine* in the hybrid design indicate a better fit to the target values, suggesting that it models the data more accurately than the other approaches.



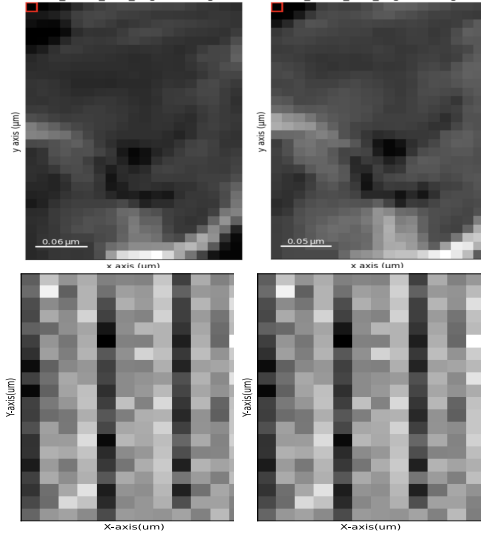


Fig. 7: From top left to right (a) EELS image of 300°C annealed sample. (b) 400°C annealed sample. (c) Denoised 300°C image. (d) Denoised 400°C image.

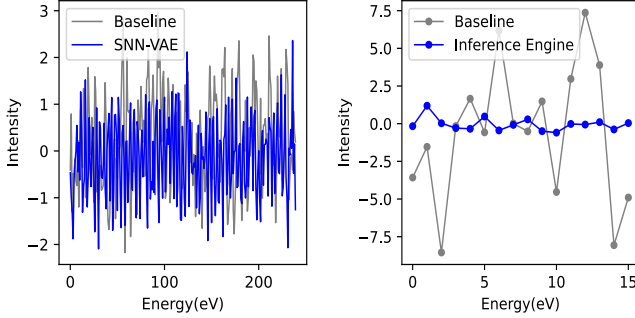


Fig. 8: (a) SNN-VAE vs baseline reconstructed spectrum. (b) Hybrid vs baseline inference spectrum.

### C. SNR Optimization and VAE Loss Reduction

The autoencoder compares the loss in the reconstructed spectrum to the original spectrum. The average MSE loss for the SNN-VAE autoencoder is 25% lower than that of the baseline.

Additionally, we calculate the SNR using the equation below, where  $s$  is the signal and  $M$  is the length of the signal.

$$SNR = \log_{10} \frac{s^2/M}{MSE} \quad (5)$$

Our SNN-VAE achieves an SNR of 28.3dB which is considered a good quality image [37]. The SNR is one of the standard indicators used to evaluate the effectiveness of image-denoising results.

### D. Performance Profiling of Hardware Architectures

Table I summarizes the FPGA implementation results of both of our designs. We make the following key observations:

1) **Latency:** We achieve latency savings of 25.2× and 93.7× for the SNN-VAE and hybrid models, respectively, compared to the baseline architecture. Notably, the most significant improvement is observed in the hybrid design, which

TABLE I. Hardware utilization on a Xilinx Virtex-7 (XC7VX485T-2FFG1761C) FPGA.

	Performance		Resource Utilization			
	Latency	Power	FF	LUT	DSP	BRAM
Baseline	131.3sec	2.29W	2288	13227	30	75
Hybrid	1.4sec	0.51W	2233	16138	2	28
SNN-VAE	5.2sec	2.20W	2802	5091	10	33

is primarily driven by the highly pipelined *Inference Engine* outlined in Algorithm 1. Furthermore, the fully integrated SNN-VAE retains the advantages of data-level parallelism by operating on multiple CU-LIF units simultaneously within a layer.

2) **Energy:** For edge-level evaluation, our hybrid architecture demonstrates energy savings of 4.5× to the baseline. This reduction is achieved as spiking neural networks (SNNs) replace complex convolution operations with a series of additions, as illustrated in Figure 3. The SNN-VAE consumes an overall power of 2.20W which is 1.04× the baseline. While fully online learning requires more memory, impacting the architecture’s static power consumption, the dynamic power of the SNN-VAE is 0.10W, much lower than the performance of the hybrid design.

3) **Utilization:** The baseline CNN-VAE exhibits the highest memory utilization, with memory distributed across BRAMs and LUTs on the device. It consumes a significantly higher number of DSPs due to the implementation of multiply-and-accumulate (MAC) units. This creates substantial data dependencies within each filter of every layer in both the encoder and decoder, a challenge that escalates as the number of heads in the model increases. Whereas, resource utilization is less than 1% for both the SNN-VAE and the hybrid framework because binary spikes are stored in registers, allowing fast data access.

## VI. CONCLUSIONS

We introduce a spiking neural network variational autoencoder (SNN-VAE) platform, comprising an *Inference Engine* and a *Learning Engine*, to reconstruct denoised spectra from EELS-SI data. Additionally, we propose a hybrid software framework, integrated with the *Inference Engine*, for compiling spiking convolutional neural networks (SCNNs) onto the proposed design. Using an MXene EELS-SI dataset, we demonstrate significant improvements in both latency and energy efficiency, enabling edge processing and real-time tracking of structural dynamics.

## VII. ACKNOWLEDGEMENT

This work is supported by the Adaptive Tunability for Synthesis and Control via Autonomous Learning on Edge (AT SCALE) Laboratory Directed Research and Development (LDRD) Initiative at Pacific Northwest National Laboratory (PNNL), Accenture LLP, the DOE DE-SC0022014, and NSF CCF-1942697.

## REFERENCES

- [1] N. Arif, S. Gul, M. Sohail, S. Rizwan, and M. Iqbal, "Synthesis and characterization of layered nb<sub>2</sub>c mxene/zns nanocomposites for highly selective electrochemical sensing of dopamine," *Ceramics International*, vol. 47, no. 2, pp. 2388–2396, 2021.
- [2] S. Sahare, P. Ghoderao, P. Yin, A. S. Saleemi, S.-L. Lee, Y. Chan, and H. Zhang, "An assessment of mxenes through scanning probe microscopy," *Small Methods*, vol. 6, no. 6, p. 2101599, 2022.
- [3] M. Krämer, B. Favelukis, A. A. El-Zoka, M. Sokol, B. A. Rosen, N. Eliaz, S.-H. Kim, and B. Gault, "Near-atomic-scale perspective on the oxidation of ti<sub>3</sub>c<sub>2</sub>tx mxenes: Insights from atom probe tomography," *Advanced Materials*, vol. 36, no. 3, p. 2305183, 2024.
- [4] J. L. Hart, K. Hantanasirisakul, A. C. Lang, B. Anasori, D. Pinto, Y. Pivak, J. T. van Omme, S. J. May, Y. Gogotsi, and M. L. Taheri, "Control of mxenes' electronic properties through termination and intercalation," *Nature communications*, vol. 10, no. 1, p. 522, 2019.
- [5] I. Persson, J. Halim, T. W. Hansen, J. B. Wagner, V. Darakchieva, J. Palisaitis, J. Rosen, and P. O. Persson, "How much oxygen can a mxene surface take before it breaks?" *Advanced Functional Materials*, vol. 30, no. 47, p. 1909005, 2020.
- [6] D. B. Williams, C. B. Carter, D. B. Williams, and C. B. Carter, *The transmission electron microscope*. Springer, 1996.
- [7] A. Annys, D. Jannis, and J. Verbeeck, "Deep learning for automated materials characterisation in core-loss electron energy loss spectroscopy," *Scientific Reports*, vol. 13, no. 1, p. 13724, 2023.
- [8] Z. Wang, D. Santhanagopalan, W. Zhang, F. Wang, H. L. Xin, K. He, J. Li, N. Dudney, and Y. S. Meng, "In situ stem-eels observation of nanoscale interfacial phenomena in all-solid-state batteries," *Nano letters*, vol. 16, no. 6, pp. 3760–3767, 2016.
- [9] B. H. Goodge, D. J. Baek, and L. F. Kourkoutis, "Atomic-resolution elemental mapping at cryogenic temperatures enabled by direct electron detection," *arXiv preprint arXiv:2007.09747*, 2020.
- [10] J. D. Hollenbach, C. M. Pate, H. Jia, J. L. Hart, P. Clancy, and M. L. Taheri, "Embedding theory in ml toward real-time tracking of structural dynamics through hyperspectral datasets," *arXiv preprint arXiv:2312.05201*, 2023.
- [11] C. M. Pate, J. L. Hart, and M. L. Taheri, "Rapiddeels: machine learning for denoising and classification in rapid acquisition electron energy loss spectroscopy," *Scientific Reports*, vol. 11, no. 1, p. 19515, 2021.
- [12] J. L. Hart, A. C. Lang, A. C. Leff, P. Longo, C. Trevor, R. D. Twisten, and M. L. Taheri, "Direct detection electron energy-loss spectroscopy: a method to push the limits of resolution and sensitivity," *Scientific reports*, vol. 7, no. 1, p. 8243, 2017.
- [13] R. Egerton, "Parallel-recording systems for electron energy loss spectroscopy (eels)," *Journal of Electron Microscopy Technique*, vol. 1, no. 1, pp. 37–52, 1984.
- [14] R. R. Meyer and A. I. Kirkland, "Characterisation of the signal and noise transfer of ccd cameras for electron detection," *Microscopy Research and Technique*, vol. 49, no. 3, pp. 269–280, 2000.
- [15] P. K. Huynh, M. L. Varshika, A. Paul, M. Isik, A. Balaji, and A. Das, "Implementing spiking neural networks on neuromorphic architectures: A review," *arXiv preprint arXiv:2202.08897*, 2022.
- [16] M. Bouvier, A. Valentian, T. Mesquida, F. Rummens, M. Reyboz, E. Vianello, and E. Beigne, "Spiking neural networks hardware implementations and challenges: A survey," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 15, no. 2, pp. 1–35, 2019.
- [17] S. Davidson and S. B. Furber, "Comparison of artificial and spiking neural networks on digital hardware," *Frontiers in Neuroscience*, vol. 15, p. 651141, 2021.
- [18] M. L. Varshika, F. Corradi, and A. Das, "Nonvolatile memories in spiking neural network architectures: Current and emerging trends," *Electronics*, vol. 11, no. 10, p. 1610, 2022.
- [19] P. Potapov and A. Lubk, "Optimal principal component analysis of stem xeds spectrum images," *Advanced Structural and Chemical Imaging*, vol. 5, pp. 1–21, 2019.
- [20] S. Lichtert and J. Verbeeck, "Statistical consequences of applying a pca noise filter on eels spectrum images," *Ultramicroscopy*, vol. 125, pp. 35–42, 2013.
- [21] J. Ryu, H. Kim, R. M. Kim, S. Kim, J. Jo, S. Lee, K. T. Nam, Y.-C. Joo, G.-C. Yi, J. Lee *et al.*, "Dimensionality reduction and unsupervised clustering for eels-si," *Ultramicroscopy*, vol. 231, p. 113314, 2021.
- [22] M. Shiga, K. Tatsumi, S. Muto, K. Tsuda, Y. Yamamoto, T. Mori, and T. Tanji, "Sparse modeling of eels and edx spectral imaging data by nonnegative matrix factorization," *Ultramicroscopy*, vol. 170, pp. 43–59, 2016.
- [23] S. P. Gleason, D. Lu, and J. Ciston, "Prediction of the cu oxidation state from eels and xas spectra using supervised machine learning," *npj Computational Materials*, vol. 10, no. 1, p. 221, 2024.
- [24] S. Tetef, N. Govind, and G. T. Seidler, "Unsupervised machine learning for unbiased chemical classification in x-ray absorption spectroscopy and x-ray emission spectroscopy," *Physical Chemistry Chemical Physics*, vol. 23, no. 41, pp. 23 586–23 601, 2021.
- [25] J. Timoshenko, D. Lu, Y. Lin, and A. I. Frenkel, "Supervised machine-learning-based determination of three-dimensional structure of metallic nanoparticles," *The journal of physical chemistry letters*, vol. 8, no. 20, pp. 5091–5098, 2017.
- [26] M. Ziatdinov, A. Ghosh, C. Y. Wong, and S. V. Kalinin, "Atomai framework for deep learning analysis of image and spectroscopy data in electron and scanning probe microscopy," *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1101–1112, 2022.
- [27] K. M. Roccapriore, M. Ziatdinov, S. H. Cho, J. A. Hachtel, and S. V. Kalinin, "Predictability of localized plasmonic responses in nanoparticle assemblies," *Small*, vol. 17, no. 21, p. 2100181, 2021.
- [28] Q. Huang, C. Hong, J. Wawrzyniak, M. Subedar, and Y. S. Shao, "Learning a continuous and reconstructible latent space for hardware accelerator design," in *2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 2022, pp. 277–287.
- [29] K. Stewart, A. Danielescu, T. Shea, and E. Neftci, "Encoding event-based data with a hybrid snn guided variational auto-encoder in neuromorphic hardware," in *Proceedings of the 2022 Annual Neuro-Inspired Computational Elements Conference*, 2022, pp. 88–97.
- [30] F. de la Peña, E. Prestat, V. T. Fauske, J. Lähnemann, P. Burdet, P. Jokubauskas, T. Furnival, C. Francis, M. Nord, T. Ostasevicius, K. E. MacArthur, D. N. Johnstone, M. Sarahan, J. Taillon, T. Aarholt, pquinn dls, V. Migunov, A. Eljarrat, J. Caron, T. Nemoto, T. Poon, S. Mazzucco, actions user, N. Tappy, N. Cautaearts, S. Somnath, T. Slater, M. Walls, pietsjoh, and H. Ramsden, "hyperspy/hyperspy: v2.1.1," 2024. [Online]. Available: <https://zenodo.org/doi/10.5281/zenodo.592838>
- [31] J. M. Joyce, *Kullback-Leibler Divergence*. Springer Berlin Heidelberg, 2011, p. 720–722. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-04898-2\\_327](http://dx.doi.org/10.1007/978-3-642-04898-2_327)
- [32] A. K. Mishra, N. Kandasamy, and A. Das, "Hardware-software co-design for on-chip learning in ai systems," in *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, 2023, pp. 624–631.
- [33] M. L. Varshika, A. Balaji, F. Corradi, A. Das, J. Stuijt, and F. Catthoor, "Design of many-core big little  $\mu$ brains for energy-efficient embedded neuromorphic computing," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 1011–1016.
- [34] J. Stuijt, M. Sifalakis, A. Yousefzadeh, and F. Corradi, " $\mu$ brain: An event-driven and fully synthesizable architecture for spiking neural networks," *Frontiers in neuroscience*, vol. 15, p. 664208, 2021.
- [35] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.
- [36] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [37] S. Jian and W. Wen, "Study on underwater image denoising algorithm based on wavelet transform," in *Journal of Physics: Conference Series*, vol. 806, no. 1. IOP Publishing, 2017, p. 012006.