Spike-based Classification with ultrafast Microlaser Neurons using Surrogate Model-assisted Genetic Algorithm training

Gibaek Kim, Matthieu Dubernard, Sami El-Nakouzi, Amir Hossein Masominia, Sylvain Barbay, Laurie E. Calvet

Spike-based Classification with ultrafast Microlaser Neurons using Surrogate Model-assisted Genetic Algorithm training

Gibaek Kim[1*], Matthieu Dubernard[1†], Sami V. El-Nakouzi[1], Amir-Hossein Masominia[2], Sylvain Barbay[2**], Laurie E. Calvet[1***]

[1] LPICM, CNRS-Ecole Polytechnique, IPP, Palaiseau, France.
[2] Université Paris-Saclay, CNRS, Centre de Nanosciences et de Nanotechnologies, Palaiseau, France.
†Current address : Université Paris-Saclay, CNRS, Centre de Nanosciences et de Nanotechnologies, Palaiseau, France.

*email: gibaek.kim@polytechnique.edu

** email : sylvain.barbay@c2n.upsaclay.fr

*** email : laurie.calvet@cnrs.fr

Abstract
Spike-based machine learning offers high computational efficiency with minimal resources, utilizing sparse coding and brain-inspired methods. Optically based systems potentially enable very fast classifications but, despite decades of research, demonstrations remain simplistic. We propose an ultra-fast photonic, spike-based hardware architecture benchmarked on the reduced MNIST dataset. The model uses efficient feature detection based on receptive fields coupled to microlaser neurons (MLNs) and is trained using a genetic algorithm. The computational cost of the simulations is significantly reduced by the introduction of a surrogate model for the MLNs. The resulting classification accuracy is comparable to a state-of-the-art multi-layer perceptron of higher structural complexity. We estimate 1000X (10X) improvement in energy consumption and 10X (1000X) faster classification can be obtained compared to software. We anticipate that our method can be quite general and apply to other spike-based hardware architectures, while ultimately leading to hardware classifications that can take just a few nanoseconds.

1. Introduction

Artificial intelligence (AI) has advanced rapidly over the past decade[1–6], however, current von Neumann architecture has limitations that continue to restrict computational efficiency[7,8]. A promising alternative is photonic neuromorphic hardware, which offers increased processing bandwidth, parallelism and low power consumption[9–13]. At the same time, spike-based neuromorphic computing has emerged as an active field that aims to realize artificial intelligence while increasing the energy efficiency of computing platforms[14]. A particularly promising hardware is therefore photonic spiking architectures.

Spiking microlasers can emulate neuron behavior with spike times as short as a few hundred picoseconds, enabling very rapid, sequential computational tasks[10,15]. Recently, spiking vertical-cavity surface emitting lasers (VCSELs) have shown potential for fast image preprocessing[11]. They have been used as the reservoir layer in reservoir computing[16,17], enabling Bayesian inference[18]. In addition they have been used for all-optical spiking networks with learning capabilities[19]. One notable advancement is an efficient feature detection where receptive fields (RFs) and photonic Micro-Laser Neurons (MLNs), VCSELs integrated with a saturable absorber, are used [10,20]. Notably, this research took advantage of biomimetic features including temporal summation, refractory time and time latency, to classify the data. However, simulating large-scale implementations remains challenging and computationally expensive, as it requires modeling complex optical neuron behavior. This work introduces a method to address these challenges.

Spiking neural networks (SNNs), which process spike trains as inputs and outputs, closely mimic biological nervour systems and offer potential improvements in energy efficiency and performance[21]. Here we model and train a single layer of optical spiking MLNs to classify the 8x8 reduced MNIST dataset. We consider ultrafast spiking MLNs whose behavior can be described by an optical neuron model (ONM) with spontaneous emission[22]. This ONM maps to a leaky integrate-and-fire neuron [23] and describes accurately the dynamics of VCSELs integrated with saturable absorbers, which display biomimetic features including excitability[22], refractoriness[24], spike latency[25] and temporal summation[26].

Optimal training and encoding for SNNs remain areas of active research. A typical approach is Spike Time Dependent Plasticity (STDP) that tunes a synaptic weight based on the relative timing of pre- and post-synaptic spikes[27–34]. Additionally, another supervised learning algorithms have been developed to minimize the error between the actual spike time and desired spike time, including the remote supervised method (ReSuMe)[35,36,34] and spike pattern association neuron (SPAN)[37]. Other methods such as S4NN and EventProp using spike time, latency or rank encodings have also been introduced[38–41], taking advantage of the sparse coding of spiking systems. Challenges persist, however, particularly with gradient-descent methods that risk becoming trapped in local optima and may not leverage hardware-specific advantages.

To overcome these limitations, heuristic approaches have also been employed to train the SNN systems, enabling the identification of globally optimized solutions[42–44]. However, these methods introduce a significant computational burden, especially in large-scale implementations. To address this, we have developed a surrogate model[45] based on a multilayer perceptron (MLP). It has a simpler calculation procedure compared to the numerical method can be vectorized for high computational efficiency. Due to this, we can use the genetic algorithm, which is a heuristic algorithm and computationally demanding, to train our system. The surrogate model also provides additional benefits. In the encoding process, we utilize RFs, which arefundamental components of biological image classification systems. However, the application of RFs to provide representations of the data input can result in an

increased computational burden due to data segmentation. The MLP model alleviates this issue by reducing the computational costs and enabling vectorized calculations, allowing effective use of RFs.

In summary, our methodology makes three key contributions. First, we demonstrate that the ONM can be compressed into an MLP surrogate model, significantly improving computational efficiency for hardware modeling. Second, this improvement allows the GA to identify optimal RF parameters, enabling the training of advanced classification tasks suited to general MLNs SNN architectures. Finally, we show that intricate data can be classified using SNNs constructed with optical components and effectively leveraging their biomimetic characteristics.

2. Methods

2.1 Spiking microlaser neuron models

2.1.1 Optical neuron model

The characteristics of ultrafast spiking MLNs can be modeled by the ONM, which consists of three coupled ordinary differential equations for the intracavity light intensity (I), and the excessive carrier densities with respect to transparency in the gain (G) and saturable absorber (SA) regions (Q):

$$\dot{I} = I(G - Q - 1) + \beta_{sp}(G + \eta_1)^2 \quad (1)$$

$$\dot{G} = \gamma_G(\mu_1 - G(1 + I)) \quad (2)$$

$$\dot{Q} = \gamma_Q(\mu_2 - Q(1 + sI)) \quad (3)$$

Parameters are μ_1, the intensity of the pump and most important parameter in practice, which controls the microlaser dynamical working regime and allows a channel for external data input, μ_2 the linear absorption, $\gamma_{G,Q}$ the gain and the saturable absorber relaxation rates, β_{sp} the spontaneous emission coefficient, s the saturation parameter, and η_1 is the carrier density at transparency. Among these parameters, the main physically controllable parameter is the pump intensity (μ_1). The time is rescaled to the cavity intensity decay time (on the order of 1.5 ps here), and the parameters $\gamma_{G,Q}$ are small parameters such that the system is of the slow-fast type.

All the simulations take place in the excitable regime of the laser. This regime occurs for a pump just below the laser threshold (when the laser emits almost no light) and is characterized by an all-or-none kind of response to perturbation. For perturbations below a given threshold (which depends on the parameters values), the microlaser will relax to its quiet, sub-laser threshold and low intensity state. For perturbations above this threshold, the microlaser will respond by a characteristic and large intensity pulse before returning to its rest state, analogous to the excitable behavior in biological neurons. Interestingly, the emitted optical pulse has a very short duration of the order of 150 ps[24]. These characteristics permit the construction of fast brain-inspired processing circuits.

Fig. 1a shows simulations using the ONM, where input pump perturbations are used to encode pixels intensity from an image. The conversion of the signals uses[20]:

$$\mu_1(t) = \mu_0 + \sum_i c_i \Pi_{\tau_p}(t - i * \tau_b) \quad (4)$$

where Π_{τ_p} is a boxcar function of duration τ_p, μ_0 is the value of base pump value, τ_b is the value of the 'bit time' duration between input image pixels, and c_i is the pump amplitude corresponding to a given pixel shade. The proper choice of pump values ensures that optical neurons can integrate multiple input signals, achieving complex feature detection. Whenever the net gain of the laser, represented by G-Q-1, goes beyond 0, an excitable pulse is emitted. The full system of equations (1-3) is in general too

computationally expensive for neural-network training. This is why we are going to introduce a surrogate model to ease computation of the system output.

2.1.2 Surrogate model

Various machine learning algorithms could replace the ONM in Eq. (1-3). However, we select a multilayer perceptron (MLP) as the surrogate model due to its simplicity in mapping low-dimensional inputs to outputs. Although more complex algorithmic schemes may allow further improvements in the accuracy of the model, there are the potential drawbacks of increment in training and computing times[46,47]. The inputs to the model are set to the base pump value (μ_0) and the amplitude (c_i) of the input bit perturbations, with the bit time τ_b set to 50 and the perturbation time τ_p set to 30. For the remaining parameters, we utilize typical values for semiconductor materials[26], including $s \approx 10$, $\gamma_{G,Q} = 0.005$, $\eta_1 = 1.6$, $\mu_2 = 0$, and $\beta = 10^{-4}$.

The successful training of the surrogate model hinges on the availability of a high-quality training dataset that contains a representative sample of the perturbations expected in the classification task. In our case, the surrogate model is tuned to predict the response of RFs and RFs encounter numerous zero pixels, as in Fig. 1c. Thus, to generate the input bit perturbations train set, we first produce a random binary signal, and then multiply it with a sequence of five randomly generated perturbation amplitudes ($[c_i]$). In addition, a randomly generated base pump value (μ_0) is associated with it. The generated input signals are calculated with the ONM, and the outputs spike intensities I_i and timings t_i are recorded in the order of arrival time as depicted in Fig. 1a.

In most cases, there are fewer output spikes than input bits (here we chose five) because of the sparsity of the response of microlasers. If there are less than five output spikes, the remaining ones are filled with zeros. As a result, there is an imbalance in the amount of information between different numbers of output spike sequences. To address, we in addition generate the binary signal as previously, but the random generator only makes a single amplitude value and assigns it to the signals. This schematic is expected to facilitate the generation of input perturbations with a greater potential to induce high order spikes compared to the initial method. We call it a supplementary process and observe its effect in Fig. S1. 80,000 different input datasets are generated with the initial method and 120,000 are with the supplementary process. It took 17 hours and 20 minutes of computation time using a single CPU of Intel XEON microprocessor 3.2 GHz running on a Dell PowerEdge server. Due to the difference in the magnitudes of the spike times and spike intensities, we used a min-max normalization to scale the results. The datasets were divided into a ratio of 8:1:1 for training, validation, and testing.

The surrogate model is ultimately trained to predict the output pulse intensities I_i and timings t_i in response to the defined input perturbation base pump value μ_0, perturbation amplitude c_i as illustrated in Fig. 1b. The MLP is composed of three hidden layers, comprising respectively 256-256-128 nodes, and connected to subsequent ones by a leaky rectified linear unit (Leaky ReLU) activation function. To train the MLP, an adaptive moment estimation (ADAM) optimizer is utilized with a learning rate scheduler that reduces the learning rate by a certain factor every 40 epochs. We also conduct optimization for hyperparameters such as batch size, learning rate, and size of train dataset and found that an initial learning rate of 1e-3, batch size of 32, and train dataset size of 200,000 were the best combination considering the surrogate model train time, dataset generation time and final quality of the surrogate model. The detailed results are presented in Fig. S2. The training process is run on the Nvidia A40 GPU and took 43 minutes to train the MLP. Furthermore, to mitigate the issue of overfitting, an early stopping method is employed with a patient value of 50.

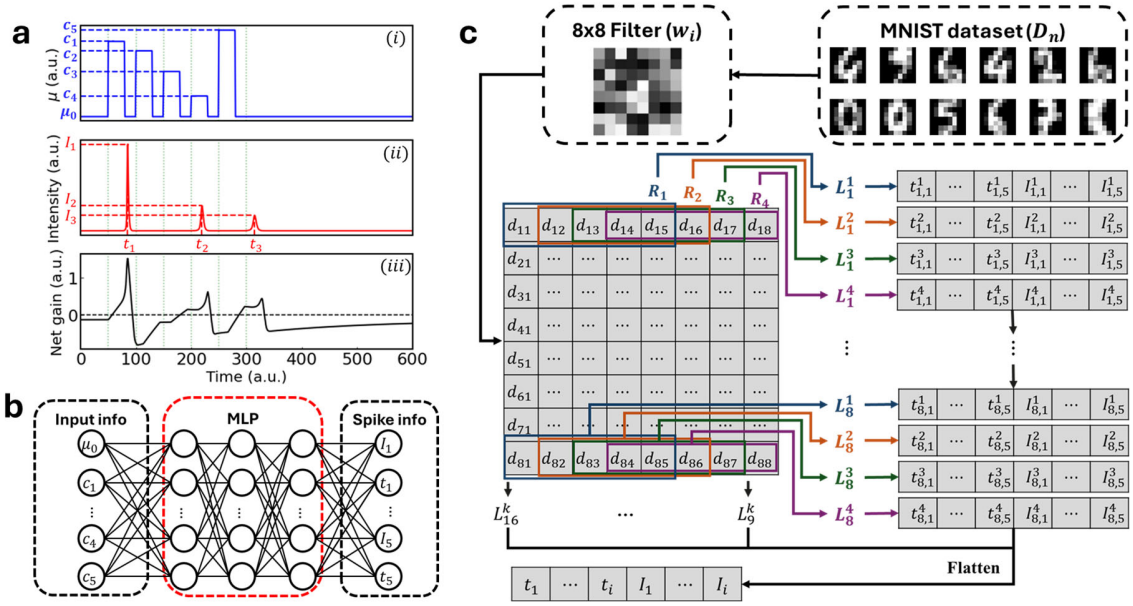


Figure 1 (a) Operation of the microlaser with input perturbations (i) input pump perturbations versus time, where μ_0 is the base pump value and c_i are perturbation amplitudes, (ii) corresponding net gain (G-Q-1) (iii) output intensity (I) of the laser. (b) Schematic of the surrogate model with multi-layer perceptron. The input parameters are the pump and each perturbation amplitude. The outputs are the spike intensities and timings. (c) Encoding system of the MNIST 8x8 images using MLNs. The raw data is passed through an 8x8 input filter, and then divided into RFs, denoted R_i. RFs are determined along horizontal and vertical rows and physically correspond to different microlasers. Finally, the calculated spike information is flattened, and the encoding process is terminated.

2.2 Optimization of MLNs feature detection

2.2.1 Encoding and decoding methods

Optimizing the encoding method for the reduced MNIST dataset is a challenging task. Algorithm 1 outlines the overall encoding procedure. For each incoming MNIST dataset image (D_n), an 8x8 filter matrix (w_i) is applied to enhance the extraction of key patterns from the raw image. The filtered data is then segmented by RFs and converted into an input pump signal as illustrated in Fig. 1c. Each RF divides the pixel data into 5 consecutive signals (horizontal or vertical), with each signal's amplitudes $[c_i]$ determined by the pixel's gray value. These signals are processed by the MLNs, each assigned a unique pump value (μ_i). Using Eq. 4, we convert the image data into input perturbations based on the amplitude and pump values. By carefully selecting pump values, the MLNs can effectively capture patterns within the image. This approach is based on previous work that heuristically explored the recognition of ten 5 x 5 binary digits using horizontal and vertical RFs[20]. The output of MLNs is combined and flattened, and we define this encoded spike information (S_n).

Selecting a segmentation option allows for control over feature extraction from input pixel data and the number of trainable parameters. Available options include the number of RFs per line and the scanning direction, both of which impact the parameter count. In the case that shown in Fig. 1c, a single 64-element filter matrix is used, with four RFs per line and scanning in both horizontal and vertical directions, resulting in 64 RFs. Since each RF is linked to a unique MLN with its own pump value, the system comprises 64 MLNs, totaling 128 trainable parameters. If only vertical scanning is used, the parameter count is reduced to 96. To accommodate the unique patterns of each digit, we construct and optimize ten separate systems, one for each digit. The optimization results for these systems are summarized in Table S1.

Algorithm 1 Encode Scheme of MNIST

Input: MNIST dataset (D_n), filter matrix (w_i), pump value (μ_i), Model (either *Surrogate* or ONM)
Initialize an encoding result S_n
function ENCODER(D_n, w_i, μ_i)
    $D_n \leftarrow D_n \times w_i$ ▷ Apply the filter to the matrix
    for each RFs k **do** ▷ Segment D_n to c_i with RFs
        Calculate the spike characteristic: $I_k, t_k \leftarrow \text{Model}(c_i, \mu_i)$
        Append selected spike information (I_k, t_k) to S_n
    end for
    $S_n \leftarrow \text{Flatten}(S_n)$
    return S_n
end function

Algorithm 1. Encoding scheme for input from the reduced MNIST dataset

To classify the labels (L_n) of MNIST data (D_n), we use the notion of the support vector machine (SVM)[48]. This entails identifying the hyperplane in the vector space of spike information resulting from the MLNs (typically spike time and intensity) and determining the plane that can separate the encoded vectors of one digit from those of the others. Unlike other algorithms, this method allows for straightforward decoding and classification of spike information without needing to define specific spike time ranges or intensities, enhancing its versatility for general SNN systems. The full description is given in Algorithm 2. First, we sort the encoded spike information (S_n) with the label L_n and define the sorted encoded result with label L as V_L. We then calculate the average point ($\overline{V_L}$), where the most of V_L are found in the vector space. When defining the $\overline{V_l}$, one may utilize information from either the spike intensities (I_i), the timings (t_i) or both (I_i, t_i). This choice is important, as spike intensities and timings exhibit unique characteristics and may have different capacities for distinguishing patterns. Optimization results for this choice are shown in Table S1. When the new data (D_i) is coming, we can determine the label L_i by estimating the distance (N) between the S_i and the $\overline{V_L}$. The label of D_i is to be the digit k for which N_k has the minimum distance among all digits.

Algorithm 2 Decode Scheme of Spike Information

Input: Spike information (S_n), Target label (L_n)
Output: Classified digit label ($\hat{L_n}$)
Initialize predict label $\hat{L_n}$
function DECODER(S_n, L_n)
    for each data S_n **do**
        $V_L \leftarrow S_N$ with label L ▷ Sort S_n with L_n
    end for
    $\bar{V_L} \leftarrow mean(V_L)$ ▷ Calculate average vector point for each L
    for each data S_n **do**
        $N_L \leftarrow |\bar{V_L} - S_n|$ ▷ Calculate the distance(N_L) between S_n and $\bar{V_L}$
        $\hat{l} \leftarrow argmin(N)$
        Append $\hat{l}$ to $\hat{L_n}$
    end for
    return $\hat{L_n}$
end function

Algorithm 2 Decoding scheme

2.2.2 Genetic Algorithm

During training, the goal of the GA is to identify the optimal combination of μ_i and w_i that can separate the $\overline{V_L}$ for specific digit L from those of the other digits. The GA begins by randomly generating an initial population of combinations, each representing different pump values and filter weights, as outlined in Algorithm 3. The combinations are assigned to MLNs, and the classification is performed through the encoding and decoding. The accuracy ratio (n_c/n_t), where n_c is the number of correct answers and n_t is the total number of answers is then calculated. After the evaluation of all populations in each generation,

elite candidates are selected, and new generations are produced by combining information from the best combinations. We choose the simple, yet powerful 128-point crossover method to generate the next generation candidate. Careful selection of hyperparameters such as mutation rate, population size, and number of generations is crucial, considering the size of the parameter space. Here, we run 100 generations with 300 candidates per generation and set the mutation rate to 0.1. These values are determined through hyperparameter optimization, as shown in Fig. S3 and S4.

Algorithm 3 Genetic Algorithm for Training Microlaser Neurons

Input:
target number (N_i), MNIST dataset(D), MNIST labels (L), number of population(P), number of generations (G), mutation rate (ϵ)
Initialize:
Best candidate(C_{best}) ← empty array
Local candidate(C) ← P randomly generated filter weights(w_i) and pump values (μ_i)
for each G **do**
  for each (w_i, μ_i) in C **do**
    S_i ← Encode(w_i, μ_i, D)
    classified labels (L') ← Decode(S_i, D)
    Objective value (O) ← Evaluate (w_i, μ_i) with L' and L
  end for
  C_{best} ← (w_i, μ_i, D) who has max(O)
  Select elite particles(E) based on O
  C ← genetic operations(E, ϵ)
end for
return C_{best}

Algorithm 3 Genetic algorithm for training MLNs

3. Results

3.1 Comparison between the surrogate model and ONM

Biomimetic features like refractory time, temporal summation, and spike latency characteristics of the MLN are important properties for neuromorphic computing. Here, we demonstrate that the surrogate model accurately predicts these characteristics. Fig. 2a depicts these phenomena using five input pump perturbations of strengths [c_i]= [5.2, 5.2, 1.3, 0, 1] and a base pump μ_0 of 2.51. The first two input perturbations are strong enough to trigger an excitable response (above-threshold), based on their bit time and pump values. However, the simulations show that only a single spike is produced because the second input perturbation (100–130) overlaps with the refractory period (103–116), preventing an additional excitable response. This confirms that the surrogate model accurately represents absolute refractory time.

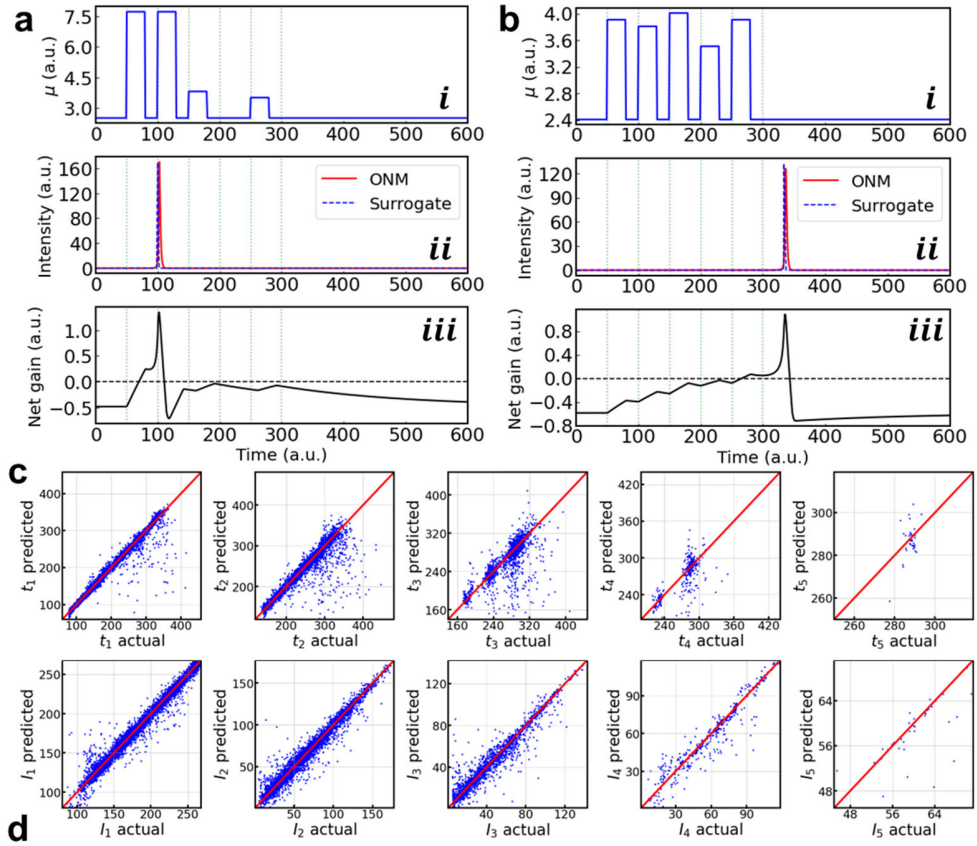
Temporal summation happens when consecutive subthreshold inputs accumulate within a short period, triggering a response spike. Fig. 2b illustrates the output spike information for five input signals, whose amplitudes are insufficient to elicit an excitable response on their own. However, both models predict one response spike, as the time interval between each pump perturbation is relatively short, around 20. This provides compelling evidence that the surrogate model can effectively predict temporal summation.

Spike time latency refers to the temporal delay between the onset of a stimulus and the corresponding response of the neuron. When the input signals are sufficiently intense to stimulate the MLN, as in Fig. 2a, the time latency is relatively brief: 20 and 22, respectively for the surrogate model and the ONM. When the amplitudes of input perturbations are closer to the excitable threshold, as in Fig. 2b, there is a significant time latency: approximately 53 and 56 respectively for surrogate model and the ONM. The predictions indicate that the surrogate model accurately predicts the nonlinear dependence of the time latency on the input perturbation amplitude.

Now, we expand the scope of our investigation by examining more general scenarios. We undertake a comparative analysis of the calculation outcomes between the ONM and the surrogate model, utilizing a test dataset. Fig. 2c illustrates the linear correlation between the actual value, as determined by the ONM, and the predicted value derived from the surrogate model. To conduct a more rigorous

quantitative analysis, three distinct evaluation metrics were calculated with the test dataset: the R2 score, the Pearson correlation coefficient (PCC), and the mean absolute error (MA). Fig. 2d demonstrates that most cases exhibit a high degree of alignment between the R2, PCC, and MA values and their optimal values. This indicates that the MLP surrogate model achieves accuracy comparable to the conventional numerical ONM.

Notably the surrogate model is more accurate for earlier spikes due to the greater amount of training information for these initial responses. Although supplementary data generation partially addresses this, some imbalance remains (see Fig. S1). However, this is expected to minimally affect the final classification model, as higher-order spikes indicate an absence of summation at the neuron level, resulting in reduced computational influence of the MLN.



	t_1	t_2	t_3	t_4	t_5	I_1	I_2	I_3	I_4	I_5
PCC	0.998	0.991	0.978	0.972	0.979	0.999	0.995	0.986	0.987	0.971
R2	0.995	0.983	0.956	0.943	0.958	0.999	0.991	0.973	0.975	0.941
RMSE	0.015	0.026	0.025	0.014	0.005	0.013	0.015	0.013	0.008	0.005

Figure 2 (a) Example response of the surrogate model to a train of pump input perturbations with pump value of 2.51 and amplitudes [5.2, 5.2, 1.3, 0, 1], (b) with pump value of 2.41 and amplitudes [1.5, 1.4, 1.6, 1.1, 1.5] (note that time is rescaled on order of 1.5 ps). (c) The scatter plots for values of spike intensity and time calculated by the ONM, versus predicted values by the surrogate model, (d) The accuracy of surrogate model for test dataset calculated with the three different evaluation metrics. Note that the optimal value for PCC and R2 is 1, and for RMSE is 0.

The utility of the surrogate model is evident when considering the training time of the MLNs. On our server, training the MLNs with the ONM would take approximately 4,809 days and 22 hours, while the surrogate model requires only 10 hours, representing a more than 10,000-fold improvement. The

reduction is due to the surrogate model's simpler computational structure and its ability to perform vectorized calculations, as illustrated in Fig. 3a. As the number of calculated signals increases, the calculation time for the ONM increases linearly, whereas for the surrogate model, it remains constant at 10^{-2} seconds. This vectorizability is a distinctive characteristic, enabling MLNs to be trained with a GA by significantly reducing the computational demands.

As an illustration, consider the training of an MLNs for the digit "8" with 300 populations and 100 generations. Fig. 1c illustrates this with an MLNs with four RFs per line, scanning both vertically and horizontally, resulting in 64 unique RFs. Given that each RF encodes five input bits, the ONM would need to process 320 input bits for a single image. In consideration of the 1617 reduced MNIST train data, 300 distinct candidates for (μ_i, w_i), and 100 generations, would result in a total of ~1.55 x 10^{10} input bits to be processed. In contrast, the total number of input bits that surrogate model processes are reduced to 3x10^4 by vectorizing the 64 RFs and the dataset.

While the advantageous computing time of the surrogate model is very clear, we now consider its precision. Due to the high computational cost of the GA when integrated with the ONM, evaluating training results with both methods is impractical. Instead, we compare the classification outcomes between the surrogate model and the ONM on an MLNs trained with the surrogate model and GA. Fig. 3b demonstrates that the accuracy of the trained MLNs is 0.996 for both models, with only one discrepancy among the 1,617 cases. Fig. 3c further confirms the computational efficiency of the surrogate model, as the ONM's calculation time reaches approximately 13 hours, while the surrogate model completes the task in just 6.7 seconds. This supports the conclusion that the surrogate model's low computational cost and vectorizability can be leveraged without compromising accuracy during training.

The inaccuracy of the surrogate model, particularly in the prediction of late arrival spike, has a negligible impact for two reasons. First, in this classification task, the surrogate model rarely encounters late arrival output spikes due to the MNIST dataset's use of numerous zeros and a 16-bit grayscale, which makes it unlikely that all five input signals would independently trigger the laser. Second, during decoding, hundreds of datasets are encoded to calculate the average vector space ($\overline{V}$) for each label. Consequently, despite the presence of a limited number of erroneous outputs, their impact is not significant.

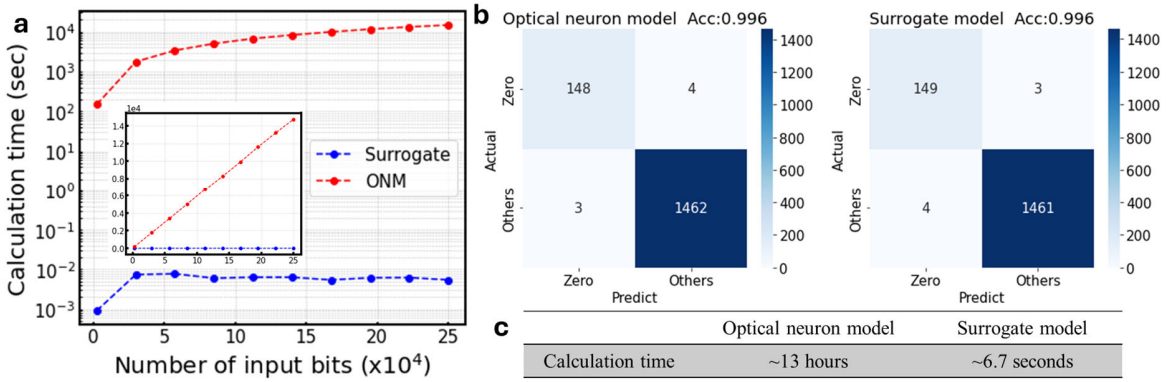


Figure 3 (a) The calculation time for calculating increasing numbers of input bits using different models. (b) Confusion matrices of MLNs for digit 0, trained with the surrogate assisted GA and classified on the test dataset with the ONM (left) and with the surrogate model (right). (c) Time consumption of ONM and surrogate model during calculating the classification behavior of trained MLNs.

3.2 Classification Result

We investigate the accuracy of the trained MLNs while varying the number of trainable parameters, which can be adjusted by modifying the MLN structure and encoding method, which is detailed in the supplementary information provided. To address variability resulting from the limited size of the training

dataset (reduced MNIST images), we employ a five-fold cross-validation approach and calculate the mean and standard deviation of test accuracy. Additionally, we compare the MLNs with a standard multilayer perceptron classifier (MLPC) with one hidden layer containing 128 neurons. Training details for both models are provided in supplementary information.

Fig. 4a shows that the test accuracy of the MLNs classification improves as the number of trainable parameters increases, reaching a maximum accuracy of 91.4% with 1,120 trainable parameters. For the MLPC, the accuracy is 93.5% with 9,472 trainable parameters. For the MLPC, accuracy reaches 93.5% with 9,472 trainable parameters. Given that the MLNs system is a single-layer model with roughly ten times fewer trainable parameters than the MLPC, we conclude that it demonstrates comparable capabilities for image classification.

We further evaluate the test accuracy of both models by adjusting the hyperparameters of their respective optimization algorithms, as the MLNs and MLPC use different training methodologies. The MLNs are optimized with a surrogate model-assisted genetic algorithm (GA), while the MLPC uses a stochastic gradient descent (SGD) optimizer. For the MLNs, we vary the population size, and for the MLPC, we adjust the learning rate. Fig. 4b shows that when the learning rate and population size are relatively low, the MLNs demonstrate higher accuracy, likely due to GA's capacity to balance exploration and exploitation effectively. However, as the learning rate increases, the MLPC achieves better accuracy, as gradient-based methods like SGD offer superior exploitation capabilities. Although this is not a direct comparison, it is clear that the MLNs system performs comparably to the MLPC, despite being a newer and developing technology, offering prospects for further enhancements.

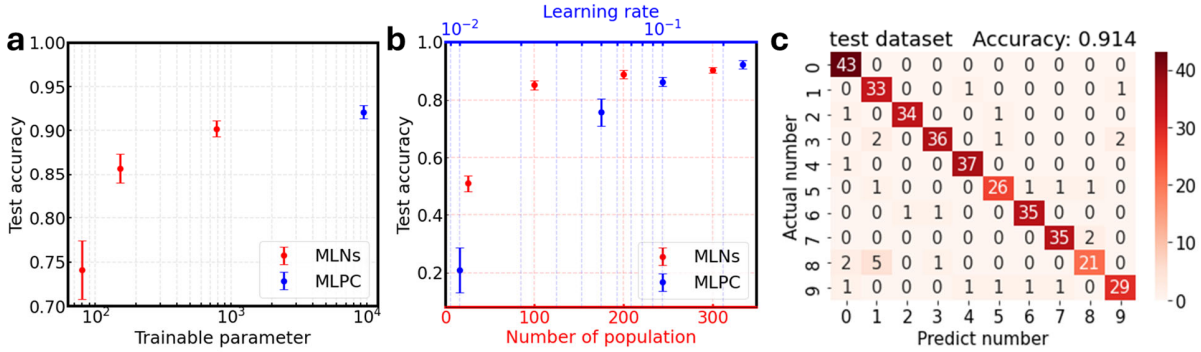


Figure 4 The accuracy of test data which is classified with (a) MLNs with varying number of trainable parameters, and MLPC, (b) MLNs and MLPC trained with varying hypermeters (number of parameter and learning rate, respectively), (c) confusion matrix of classification result of MLNs.

4. Discussion

In this paper, we selected a surrogate model for training MLNs and subsequently compared the encoding and decoding results from the surrogate model with those from the ONM, finding them to be nearly identical. While this replacement is effective in our case, it may not be universally applicable. When selecting models, two key factors should be considered: computational cost and accuracy. For applications requiring high accuracy, the ONM is optimal. However, when many iterations are needed, the MLP surrogate model can significantly reduce computational demands. In our case, the surrogate model is essential for training the MLNs with GA, which is a highly iterative and computationally expensive process. However, verifying that small inaccuracies do not compound during training is crucial to maintain the MLNs classification's functionality.

Although the surrogate model worked well with GA in our study, it may need further optimization for other algorithms, especially if small inaccuracies become significant. In our system, a single candidate set of filter weights and pump values encodes thousands of MNIST datasets, so minor encoding errors are

relatively inconsequential. However, in systems where small errors could have a substantial impact, a more precise surrogate model may be required. One possible solution is to adopt advanced neural networks, such as convolutional neural networks, long short-term memory (LSTM), or gated recurrent units (GRU) as surrogate models. Since the surrogate model's input and output signals are sequential, LSTMs and GRUs, which are specialized for sequence data, may be particularly suitable. However, this approach would increase computational costs, necessitating a careful balance between accuracy and computational efficiency.

To assess the accuracy of the MLNs architecture, we compared it with an MLPC. Although the two systems use different algorithms and varying hyperparameters, limiting the scope for strict comparative analysis, we found that the MLNs classifier achieved accuracy comparable to that of the MLPC. As the MLNs system is still in its early stages, there is potential for enhancement. Hybrid methods combining particle-based metaheuristic algorithms with gradient-based algorithms could improve performance, leveraging both exploration and exploitation capabilities. Another approach could involve developing a more complex, multi-layer MLNs architecture. As shown in Fig. **4**a, increasing the number of parameters enhances system accuracy, suggesting that stacking MLNs layers may yield further improvements.

The MLNs classification system benefits from optical components that enable ultrafast processing, eliminate information transfer delays, and enhance energy efficiency. For experimental implementation, we foresee no major conceptual challenges, though integrating components to input and collect digit information may be challenging. This could be solved using the approach of microfibers for connecting the MLNs[13]. In our encoding system, we have a filter that is multiplied with the input MNIST dataset. This filter can be implemented using a variety of methodologies, including wavelength-division multiplexed (WDM) signals with mirroring resonator (MRR) weight banks[49]. Also, this could also be simply processed by ultrafast electronics at the input of the MLNs.

By operating in parallel, each RF could be processed in under 5 ns[20]. A simpler but slower approach would involve processing the RFs sequentially, taking about 320 ns, with a power consumption of only a few tens of mW for pump laser power. We estimate that our architecture could perform classifications with energy consumption in the tens of nJ range, compared to tens of μJ for a single photonic neuron on a conventional computer. An array of photonic neurons could further increase processing speed by at least 100 times, although this would increase energy consumption proportionally. This inherent tradeoff allows the technique to be tailored to applications prioritizing either reduced power consumption or processing speed.

5. Conclusion

In this investigation, we used a surrogate model-assisted GA to train an MLN classification model. This approach leverages the biomimetic characteristics of spiking neurons—specifically, time latency, temporal summation, and refractory time. By taking advantage of the surrogate model's low computational cost and vectorizability, we achieved training speeds over 1,000 times faster than with the ONM, demonstrating that training the MLN model would be impractical without the surrogate model. A noted limitation of this technique was reduced accuracy for later-arriving spikes, which had limited impact for the reduced MNIST dataset but may be relevant in other classification tasks.

This study represents a significant step forward in developing photonic hardware for neuromorphic systems. We demonstrated a method that greatly reduces the computational load of simulating spiking photonic systems through the use of a surrogate model. Moreover, we showed that a spike-based architecture with MLNs is more energy efficient and faster with comparable accuracies to state-of-the-art second-generation neural networks. Although our results focus on specific hardware, this approach can be applied to other types of spiking systems. Our general method enables more complex

modeling of spike-based hardware, potentially paving the way for nanosecond-level classification in photonic systems.

Acknowledgments:
The authors acknowledge the support of the French Agence Nationale de la Recherche (ANR), under grants ANR-19-CE24-0006 (project ANACONDA) & ANR-21-CE24-0027 (project PHOTOMIC) and the LABEX nanosaclay, CANAPO project.

Reference

[1] Silver D, Huang A, Maddison C J, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V and Lanctot M 2016 Mastering the game of Go with deep neural networks and tree search *Nature* **529** 484–9

[2] Wu Y, Schuster M, Chen Z, Le Q V, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K, Klingner J, Shah A, Johnson M, Liu X, Kaiser Ł, Gouws S, Kato Y, Kudo T, Kazawa H, Stevens K, Kurian G, Patil N, Wang W, Young C, Smith J, Riesa J, Rudnick A, Vinyals O, Corrado G, Hughes M and Dean J 2016 Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation http://arxiv.org/abs/1609.08144

[3] Capper D, Jones D T, Sill M, Hovestadt V, Schrimpf D, Sturm D, Koelsche C, Sahm F, Chavez L and Reuss D E 2018 DNA methylation-based classification of central nervous system tumours *Nature* **555** 469–74

[4] Peurifoy J, Shen Y, Jing L, Yang Y, Cano-Renteria F, DeLacy B G, Joannopoulos J D, Tegmark M and Soljačić M 2018 Nanophotonic particle simulation and inverse design using artificial neural networks *Sci. Adv.* **4** eaar4206

[5] Khaireh-Walieh A, Langevin D, Bennet P, Teytaud O, Moreau A and Wiecha P R 2023 A newcomer's guide to deep learning for inverse design in nano-photonics *Nanophotonics* **12** 4387–414

[6] Jang D, Kim S and Kim J 2024 Deep-learning-based inverse design of colloidal quantum dots *Opt. Commun.* 130384

[7] Thompson N C, Greenewald K, Lee K and Manso G F 2020 The computational limits of deep learning *ArXiv Prepr. ArXiv200705558* **10**

[8] Berggren K, Xia Q, Likharev K K, Strukov D B, Jiang H, Mikolajick T, Querlioz D, Salinga M, Erickson J R and Pi S 2020 Roadmap on emerging hardware and technology for machine learning *Nanotechnology* **32** 012002

[9] Shastri B J, Tait A N, Ferreira de Lima T, Pernice W H, Bhaskaran H, Wright C D and Prucnal P R 2021 Photonics for artificial intelligence and neuromorphic computing *Nat. Photonics* **15** 102–14

[10] Pammi V A, Alfaro-Bittner K, Clerc M G and Barbay S 2019 Photonic computing with single and coupled spiking micropillar lasers *IEEE J. Sel. Top. Quantum Electron.* **26** 1–7

[11] Robertson J, Kirkland P, Alanis J A, Hejda M, Bueno J, Di Caterina G and Hurtado A 2022 Ultrafast neuromorphic photonic image processing with a VCSEL neuron *Sci. Rep.* **12** 4874

[12] Prucnal P R, Shastri B J, de Lima T F, Nahmias M A and Tait A N 2016 Recent progress in semiconductor excitable lasers for photonic spike processing *Adv. Opt. Photonics* **8** 228–99

[13] Moughames J, Porte X, Thiel M, Ulliac G, Larger L, Jacquot M, Kadic M and Brunner D 2020 Three-dimensional waveguide interconnects for scalable integration of photonic neural networks *Optica* **7** 640–6

[14] Roy K, Jaiswal A and Panda P 2019 Towards spike-based machine intelligence with neuromorphic computing *Nature* **575** 607–17

[15] Skalli A, Robertson J, Owen-Newns D, Hejda M, Porte X, Reitzenstein S, Hurtado A and Brunner D 2022 Photonic neuromorphic computing using vertical cavity semiconductor lasers *Opt. Mater. Express* **12** 2395–414

[16] Owen-Newns D, Robertson J, Hejda M and Hurtado A 2023 Photonic Spiking Neural Networks with Highly Efficient Training Protocols for Ultrafast Neuromorphic Computing Systems *Intell. Comput.* **2** 0031

[17] Owen-Newns D, Robertson J, Hejda M and Hurtado A 2022 GHz rate neuromorphic photonic spiking neural network with a single vertical-cavity surface-emitting laser (VCSEL) *IEEE J. Sel. Top. Quantum Electron.* **29** 1–10

[18] Ma B, Zhang J, Li X and Zou W 2023 Stochastic photonic spiking neuron for Bayesian inference with unsupervised learning *Opt. Lett.* **48** 1411–4

[19] Feldmann J, Youngblood N, Wright C D, Bhaskaran H and Pernice W H 2019 All-optical spiking neurosynaptic networks with self-learning capabilities *Nature* **569** 208–14

[20] Masominia A, Calvet L E, Thorpe S and Barbay S 2023 Online spike-based recognition of digits with ultrafast microlaser neurons *Front. Comput. Neurosci.* **17** 1164472

[21] Maass W 1997 Networks of spiking neurons: the third generation of neural network models *Neural Netw.* **10** 1659–71

[22] Barbay S, Kuszelewicz R and Yacomotti A M 2011 Excitability in a semiconductor laser with saturable absorber *Opt. Lett.* **36** 4476–8

[23] Nahmias M A, Shastri B J, Tait A N and Prucnal P R 2013 A leaky integrate-and-fire laser neuron for ultrafast cognitive computing *IEEE J. Sel. Top. Quantum Electron.* **19** 1–12

[24] Selmi F, Braive R, Beaudoin G, Sagnes I, Kuszelewicz R and Barbay S 2014 Relative Refractory Period in an Excitable Semiconductor Laser *Phys. Rev. Lett.* **112** 183902

[25] Selmi F, Braive R, Beaudoin G, Sagnes I, Kuszelewicz R, Erneux T and Barbay S 2016 Spike latency and response properties of an excitable micropillar laser *Phys. Rev. E* **94** 042219

[26] Selmi F, Braive R, Beaudoin G, Sagnes I, Kuszelewicz R and Barbay S 2015 Temporal summation in a neuromimetic micropillar laser *Opt. Lett.* **40** 5690–3

[27] Gerstner W, Kempter R, Van Hemmen J L and Wagner H 1996 A neuronal learning rule for sub-millisecond temporal coding *Nature* **383** 76–8

[28] Abbott L F and Nelson S B 2000 Synaptic plasticity: taming the beast *Nat. Neurosci.* **3** 1178–83

[29] Iakymchuk T, Rosado-Muñoz A, Guerrero-Martínez J F, Bataller-Mompeán M and Francés-Víllora J V 2015 Simplified spiking neural network architecture and STDP learning algorithm applied to image classification *EURASIP J. Image Video Process.* **2015** 4

[30] Chen T, Huang Y, Zhou P, Mu P, Xiang S, Chizhevsky V N and Li N 2023 Receptive Field-Based All-optical Spiking Neural Network For Image Processing *IEEE J. Quantum Electron.*

[31] Shrestha A, Ahmed K, Wang Y and Qiu Q 2017 Stable spike-timing dependent plasticity rule for multilayer unsupervised and supervised learning *2017 international joint conference on neural networks (IJCNN)* (IEEE) pp 1999–2006

[32] Diehl P U and Cook M 2015 Unsupervised learning of digit recognition using spike-timing-dependent plasticity *Front. Comput. Neurosci.* **9** 99

[33] Kheradpisheh S R, Ganjtabesh M, Thorpe S J and Masquelier T 2018 STDP-based spiking deep convolutional neural networks for object recognition *Neural Netw.* **99** 56–67

[34] Zhang M, Wu J, Belatreche A, Pan Z, Xie X, Chua Y, Li G, Qu H and Li H 2020 Supervised learning in spiking neural networks with synaptic delay-weight plasticity *Neurocomputing* **409** 103–18

[35] Kasinski A and Ponulak F 2005 Experimental Demonstration of Learning Properties of a New Supervised Learning Method for the Spiking Neural Networks *Artificial Neural Networks: Biological Inspirations – ICANN 2005* Lecture Notes in Computer Science vol 3696, ed W Duch, J Kacprzyk, E Oja and S Zadrożny (Berlin, Heidelberg: Springer Berlin Heidelberg) pp 145–52

[36] Ponulak F and Kasiński A 2010 Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting *Neural Comput.* **22** 467–510

[37] Mohemmed A, Schliebs S, Matsuda S and Kasabov N 2012 SPAN: SPIKE PATTERN ASSOCIATION NEURON FOR LEARNING SPATIO-TEMPORAL SPIKE PATTERNS *Int. J. Neural Syst.* **22** 1250012

[38] Gardner B and Grüning A 2021 Supervised learning with first-to-spike decoding in multilayer spiking neural networks *Front. Comput. Neurosci.* **15** 617862

[39] Bonilla L, Gautrais J, Thorpe S and Masquelier T 2022 Analyzing time-to-first-spike coding schemes: A theoretical approach *Front. Neurosci.* **16** 971937

[40] Kheradpisheh S R and Masquelier T 2020 Temporal Backpropagation for Spiking Neural Networks with One Spike per Neuron *Int. J. Neural Syst.* **30** 2050027

[41] Wunderlich T C and Pehle C 2021 Event-based backpropagation can compute exact gradients for spiking neural networks *Sci. Rep.* **11** 12829

[42] Turkson R E, Liu S, Baagyere E Y and Eghan M J 2019 Using meta-heuristic algorithm in spiking neural network for pattern recognition tasks *2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing* (IEEE) pp 22–8

[43] Schuman C D, Mitchell J P, Patton R M, Potok T E and Plank J S 2020 Evolutionary Optimization for Neuromorphic Systems *Proceedings of the 2020 Annual Neuro-Inspired Computational Elements Workshop* NICE '20 (New York, NY, USA: Association for Computing Machinery) pp 1–9

[44] Javanshir A, Nguyen T T, Mahmud M P and Kouzani A Z 2023 Training spiking neural networks with metaheuristic algorithms *Appl. Sci.* **13** 4809

[45] Ong Y S, Nair P B and Keane A J 2003 Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling *AIAA J.* **41** 687–96

[46] Shah D, Campbell W and Zulkernine F H 2018 A comparative study of LSTM and DNN for stock market forecasting *2018 IEEE international conference on big data (big data)* (IEEE) pp 4148–55

[47] Desai M and Shah M 2021 An anatomization on breast cancer detection and diagnosis employing multi-layer perceptron neural network (MLP) and Convolutional neural network (CNN) *Clin. EHealth* **4** 1–11

[48] Cortes C and Vapnik V 1995 Support-vector networks *Mach. Learn.* **20** 273–97

[49] Tait A N, Ferreira De Lima T, Nahmias M A, Shastri B J and Prucnal P R 2016 Continuous Calibration of Microring Weights for Analog Optical Networks *IEEE Photonics Technol. Lett.* **28** 887–90