PAPER • **OPEN ACCESS**

Efficient sparse spiking auto-encoder for reconstruction, denoising and classification

IOP Publishing Neuromorph. Comput. Eng. 4 (2024) 034005 https://doi.org/10.1088/2634-4386/ad5c97

NEUROMORPHIC
Computing and Engineering

CrossMark

OPEN ACCESS

RECEIVED
11 March 2024

REVISED
7 June 2024

ACCEPTED FOR PUBLICATION
27 June 2024

PUBLISHED
2 August 2024

PAPER

Efficient sparse spiking auto-encoder for reconstruction, denoising and classification

Ben Walters[1], **Hamid Rahimian Kalatehbali**[2], **Zhengyu Cai**[3], **Roman Genov**[3], **Amirali Amirsoleimani**[2], **Jason Eshraghian**[4] and **Mostafa Rahimi Azghadi**[1,*]

1 College of Science and Engineering, James Cook University, Townsville, Australia
2 Department of Electrical Engineering and Computer Science, York University, Toronto, Canada
3 Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada
4 Department of Electrical and Computer Engineering, University of California, Santa Cruz, CA, United States of America
* Author to whom any correspondence should be addressed.

E-mail: mostafa.rahimiazghadi@jcu.edu.au

Keywords: SAE, STDP, spiking neural networks (SNNs), neuromorphic computing

Abstract

Auto-encoders are capable of performing input reconstruction, denoising, and classification through an encoder-decoder structure. Spiking Auto-Encoders (SAEs) can utilize asynchronous sparse spikes to improve power efficiency and processing latency on neuromorphic hardware. In our work, we propose an efficient SAE trained using only Spike-Timing-Dependant Plasticity (STDP) learning. Our auto-encoder uses the Time-To-First-Spike (TTFS) encoding scheme and needs to update all synaptic weights only once per input, promoting both training and inference efficiency due to the extreme sparsity. We showcase robust reconstruction performance on the Modified National Institute of Standards and Technology (MNIST) and Fashion-MNIST datasets with significantly fewer spikes compared to state-of-the-art SAEs by 1–3 orders of magnitude. Moreover, we achieve robust noise reduction results on the MNIST dataset. When the same noisy inputs are used for classification, accuracy degradation is reduced by 30%–80% compared to prior works. It also exhibits classification accuracies comparable to previous STDP-based classifiers, while remaining competitive with other backpropagation-based spiking classifiers that require global learning through gradients and significantly more spikes for encoding and classification of MNIST/Fashion-MNIST inputs. The presented results demonstrate a promising pathway towards building efficient sparse spiking auto-encoders with local learning, making them highly suited for hardware integration.

1. Introduction

In the era of big data and artificial intelligence, the sheer magnitude of data necessitates efficient processing. Effective training and inference schemes are thus vital research areas. Self-supervised learning has emerged as a promising method, eliminating the laborious process of labeling datasets [1]. Auto-encoders stand out as networks capable of self-supervised learning, encoding and decoding inputs to reconstruct the original input [2]. Widely applied across various domains, auto-encoders facilitate tasks such as anomaly detection in brain images [3] and analyzing traffic presence based on noise [4].

Spiking Auto-Encoders represent a unique form of auto-encoder that harnesses voltage/current spikes, mimicking the behavior of biological neurons to convey information within the network [5]. Spiking networks are of significant interest as they can leverage their inherent temporal dynamics to learn spatiotemporal patterns and features. Furthermore, their asynchronous operation means calculations are executed at the occurrence of events, substantially reducing power consumption. Whilst these networks are incompatible with well-known gradient-based error backpropagation algorithms that are widely used in artificial neural networks, spiking neural networks are an attractive prospect for edge-processing applications [6, 7].

Furthermore, we consider our network to be hardware friendly compared to other spiking approaches such as graph-based [8, 9] and image-based [10, 11] processes. Graph-based spiking neural networks require the calculation of dynamically generated graphs, whose features are then extracted for the training process. This method requires high memory overhead to store the dynamically changing graphs. Meanwhile, image-oriented approaches like those proposed in [10, 11] require high complexity calculations such as the calculation of time-surfaces, whereas our approach mostly utilizes multiply and accumulate operations.

While surrogate gradients and spike-time differentiability have been used effectively in training SNNs [12, 13], they are expensive algorithms due to the need for weight symmetry, and spatiotemporal credit assignment problems [14]. These issues arise because each weight update requires a gradient with respect to a loss signal, and every gradient must be routed in the reverse direction of the forward pass. Storing each of these paths, along with the gradient computation for all synaptic weights during many training cycles is very power- and resource-hungry [15]. Although some backpropagation-based works have attempted to resolve some of these issues [16], there remain opportunities to explore unsupervised learning paradigms inspired by biological principles, aiming to achieve efficiencies similar to those observed in biological systems.

One such opportunity is the STDP [17] learning rule. STDP is a biologically derived model for weight updates which leverages the specific times at which spikes are presented to the network to calculate the synaptic weight update [18, 19]. Furthermore, STDP is a local learning rule, requiring only the information between adjacent nodes, allowing for a much simpler implementation. Remarkable results have also been achieved using STDP in image classification and pattern recognition [20–22]. However, oftentimes STDP underperforms compared to global gradient-based error backpropagation learning, in data-driven tasks [23]. Improvements can be made to the STDP rule by considering an error or reward signal, often referred to as error-modulated or reward-modulated STDP [24, 25]. Previous implementations have used error-modulated STDP in a variety of ways. In most cases, the reward/error signal term is multiplied by the synaptic weight change (Δw) to contribute to the synaptic weight update [26, 27].

In our work, we introduce a novel method of calculating the error signal for error modulated STDP. While most applications of STDP are focused on low-dimensional tasks, such as the classification of patterns or simple datasets, we demonstrate how to enhance the learning capabilities of local STDP learning to perform the more challenging task of image reconstruction, by using error-modulation in a spiking auto-encoder.

In addition, we investigate and develop our SAE to utilize very few spikes in its local learning, while needing only one spike for encoding each of its input image pixels, achieved through TTFS. This is to reduce the amount of data communication and computation in the network, tasks that demand a significant amount of the system's operational energy and resources. Our approach is also further aligned with an understanding that the brain relies mostly on spike-timing in light of energy constraints, rather than maximizing the firing rate of particular neurons [28]. However, as already mentioned, despite the sparsity advantage, time-based encoding schemes severely underperform in gradient-based approaches. Hence, devising novel time-based learning rules, e.g. our proposed error-modulated STDP, for temporally encoded data is significant. This pushes the potential of the STDP-driven literature towards tasks that go beyond classification on toy problems.

Our specific contributions are listed as follows:

- We have developed a novel method of implementing error-modulated STDP.
- We have designed a spiking auto-encoder capable of MNIST and Fashion-MNIST image reconstruction and denoising using only STDP-based learning rules.
- Our auto-encoder needs only an average of 9.8 spikes in its hidden layer for reconstruction of MNIST digits, leading to a very sparse and hence potentially power-efficient implementation.
- We have validated that our reconstructions are suitable for downstream tasks like classification, and verified our network's robustness to noise [29].

2. Previous work

Reconstructing and filtering inputs on devices with limited resources and power constraints holds significant practical implications. Previous research has explored the use of SAEs for this purpose. Convolutional architectures trained with backpropagation learning rules, as investigated by [2, 30], are deemed power- and resource-intensive paradigms. Kamata *et al* [2] implements a variational auto-encoder learning rule, by randomly sampling SNN outputs to simulate a Bernoulli process. In their work, many convolutional layers are trained using membrane potential backpropagation, resulting in a large network structure. Furthermore, this work requires random sampling of individual SNN network outputs to replicate the Bernoulli sampling process. Thus, many output spikes generated through this procedure are often discarded, which is inefficient for learning. Hübotter *et al* [30] demonstrates how activity regularization schemes can control the spike density and avoid bursting and dying neurons. However, their convolutional architecture and backpropagation-based learning rule detracts from the various advantages that are present with more neuromorphic approaches.

An approach that is more representative of the brain's learning mechanisms for use in spiking auto-encoders are models that learn independently of symmetric backward-passes, such as those explored in [31, 32]. Shimmyo *et al* [31] investigates the trade-off between number of timesteps and reconstruction error, where they noted that longer timesteps typically led to lower reconstruction error. However, their work adopts surrogate gradients to perform backpropagation as well as a Poissonian rate encoding scheme to encode the input. This significantly increases the complexity and power consumption of the network [32]. explores how temporal coding can be utilized in auto-encoder models. Their model only requires an input, hidden and output layer, however utilizes spike time backpropagation, an inefficient process. Furthermore, additional training pulses are required to assist the learning.

In order to remove the weight transport issues associated with backpropagation, unsupervised local learning paradigms have also been investigated in input reconstruction attempts. [33–35] delve into investigating image reconstruction using STDP-based learning rules. Notably, [33] employs the STDP rule for reconstructing 5×5 patches of the original image. Image reconstruction is then possible by organizing the learned patches in order and transposing the weight matrix for each patch to generate a decoder. However, their approach employs a rate-based input encoding scheme, leading to a significant spike count across the network. Similarly, [34] investigates an engineered STDP rule for reproducing handwritten digits. Their rule is similar to other spike based learning procedures in [36, 37] except for the fact that their rule is spatio-temporally local. However their convolutional structure combined with its excessive spike usage in its encoding scheme detracts from its power and resource efficiency. On the other hand, [35] utilizes a mirrored STDP learning rule for handwritten digit classification. This mirrored STDP involves both a feed-forward and feedback connection between the input and hidden layers, allowing for the weight matrix to be adequately transposed for the decoding section. However, their network requires image pre-processing and introduces high power consumption through synaptic current input encoding.

Hence, our research endeavors to achieve image reconstruction efficiently by minimizing spike count not only in our input encoding scheme, but also using our innovative local STDP learning rule, which operates with significantly fewer spikes compared to the state-of-the-art. Our work paves the way for hardware-based, low-power STDP implementation, in a similar fashion to prior art such as [38–41].

3. Network architecture

An auto-encoder network is a type of artificial neural network that consists of two main components: an encoder and a decoder. The encoder part compresses the input data into a lower-dimensional representation, often called a latent space or encoding. This process involves reducing the input data's dimensionality while preserving its essential features. The decoder, on the other hand, aims to reconstruct the original input data from the encoded representation generated by the encoder. It takes the compressed representation and maps it back to the original data space, attempting to generate an output that closely resembles the input. By training the auto-encoder to minimize the reconstruction error between the input and output data, it learns to capture meaningful patterns and features in the data while efficiently compressing and decompressing it. In this section, we explain the details of the main components of our proposed spiking auto-encoder architecture, shown in figure 1.

3.1. Input encoding

As shown in figure 1, the input is encoded using a TTFS scheme. In this scheme, pixels with high intensities are encoded into the earliest spike times whilst the low intensity pixels fire at the latest times. The use of TTFS encoding is advantageous due to its sparse nature of one spike per one input. This is in contrast to rate encoding schemes carrying less information per spike [42], but needing generation and transmission of more spikes leading to increased energy use and more learning steps. Additionally, TTFS can enhance dynamic sparsity in a network. We, therefore, encode input pixels to have one spike per input neuron, as illustrated in the figure. For the MNIST dataset, this results in $28 \times 28 = 784$ spikes being generated to encode each input image, corresponding to the number of pixels in an MNIST image.

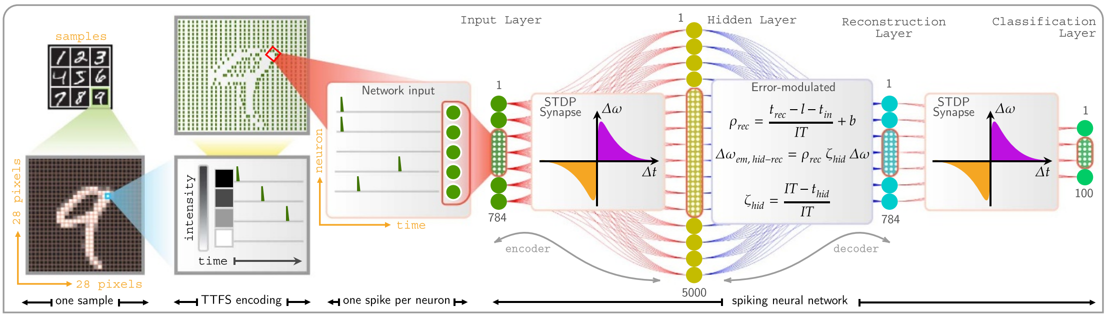

Figure 1. The proposed STDP-based spiking auto-encoder architecture, with a final classification layer added for reconstructed input classification as an example downstream task.

3.2. Encoder

In figure 1, the network receives a single spike per neuron as input, and each of the 784 input neurons is connected to the hidden layer via STDP synapses. Equation (1) is used to implement the STDP learning,

$$\Delta w = \begin{cases} A^{+} e^{\frac{-\Delta t}{\tau_{+}}} & \Delta t > 0 \\ A^{-} e^{\frac{\Delta t}{\tau_{-}}} & \text{otherwise} \end{cases} \tag{1}$$

where Δw is the change in synaptic weight, A^{+} and A^{-} are the amplitudes of potentiation and depression, and τ_{+} and τ_{-} are the time constants for potentiation and depression, respectively [43]. Furthermore, to improve efficiency, each synaptic operation only occurs once during each image presentation. To achieve this, the earliest spike times in the hidden layer are used to compute the change in synaptic weight.

The hidden layer consists of 5000 Leaky Integrate-and-Fire (LIF) [44] neurons, which accumulate input voltage/current spikes by increasing their internal state variable known as the membrane potential. Once the membrane potential of a neuron surpasses a predefined threshold, it generates a post-synaptic output spike, propagating it to the next layer. Equation (2) describes the LIF membrane potential dynamics,

$$U_{t+1} = \beta\, U_t + I_{\text{in},\, t+1} \tag{2}$$

where β denotes the membrane potential decay rate, U_t represents the membrane potential at timestep t, and I_{in} is the input synaptic current. Once the membrane potential exceeds the neuron's threshold, the membrane potential is reset to the resting potential of 0. These neurons are also connected with inhibitory synapses from other hidden neurons, to achieve competitive learning in a Winner-Takes-All setting.

The fine-tuning of each neuron's threshold occurs through a homeostatic regulation scheme, where the rate of activity, particularly post-synaptic spiking events, is monitored. Homeostatic regulation here refers to the regulation of neuronal behavior based on disproportionate activity levels. During training, some neurons spike more frequently than others, which has a deleterious effect on the results. To combat this, we lowered the neuron's threshold at a fixed rate of τ_{th} when inactive and increased the neuron's threshold by A_{th} when it spiked. This homeostatic regulation is achieved by equation (3),

$$\Delta V_{\text{th}}[t] = A_{\text{th}} S[t] - \tau_{\text{th}} \tag{3}$$

where $\Delta V_{\text{th}}[t]$ is the change in the neuron's threshold, A_{th} is a positive value that determines how much the threshold is increased, τ_{th} is the decay rate of the threshold, and $S[t]$ is equal to 1 when the neuron has spiked at time t and 0 otherwise.

3.3. Decoder

The decoder consists of 784 LIF neurons in its reconstruction layer, connected to the hidden layer through error-modulated STDP synapses. figure 1 shows that each neuron in the reconstruction layer represents a pixel correlating to the original image, where the timing of the spikes in this layer represent the pixel intensity. Given that the input was encoded using the TTFS scheme, the output of the reconstruction layer is interpreted using the same encoding scheme. Therefore, neurons firing earliest are indicative of pixels with higher intensity. In contrast to the hidden layer, no inhibition is imposed on the reconstruction layer, allowing neurons in this layer to fire concurrently. Similar to the hidden layer, we experimented with including and removing homeostatic regulation in the reconstruction layer and found that no regulation worked best.

The reconstruction ability of our auto-encoder is achieved through its novel error-modulated STDP. The concept of error-modulated STDP has been explored in previous works [26, 27, 45]. The majority of these works utilize the error as a binary or ternary signal to change the sign of the weight update. In this study, we expanded upon this concept, by ensuring that the sign and magnitude of weight update is proportional to the error signal itself. By doing this, weight updates in this layer would converge to a specific value that best represents the spike timing of the output. The calculation of this value can be expressed using equation (4), where ρ_j represents the error factor associated with the jth post-synaptic neuron, T_{rec} denotes the reconstruction layer spike time, l is a latency term to account for the accumulation of features, T_{in} is the TTFS encoded time, IT is the total length of time for which the input is presented, and b is a biasing term aiding in potentiation

$$\rho_j = \frac{(T_{\text{rec}} - l - T_{\text{in}})}{IT} + b. \tag{4}$$

Clearly, our values of rho ensure that the network aims to minimize the error difference between reconstructed and input images.In our method, ρ_j is multiplied by the weight update induced by STDP. Previous studies implementing auto-encoder learning rules have highlighted the need for hidden activity to contribute to error-modulated weight updates [35]. Hence, an additional multiplicative term, ζ_i for the ith neuron in the hidden layer, is introduced. This term ensures that the hidden neuron most responsible for the reconstruction layer firing receives are attributed with higher values of weight update than those that were not. The range for ζ_i is from 0 to 1, where higher values are associated with earlier spikes in the hidden layer. Equation (5) illustrates this effect, where IT denotes the total time during which the image is presented to the network, and T_{hid} represents the firing time of the neuron in the hidden layer

$$\zeta_i = \frac{(IT - T_{\text{hid}})}{IT}. \tag{5}$$

Equation (6) defines the weight update rule applied to the error-modulated STDP synapses. In this expression, $\Delta w_{\text{em},\,ij}$ represents the change in synaptic weight for the error-modulated synapse connected between the ith presynaptic hidden layer neuron and jth post-synaptic reconstruction neuron, and Δw is the change in synaptic weight as a result of standard STDP from equation (1)

$$\Delta w_{\text{em},\,ij} = \rho_j \zeta_i \,\Delta w. \tag{6}$$

3.4. Classification layer

To demonstrate the potential utility of the reconstructed input image in downstream tasks, we choose to conduct experiments with a classification task. This involves integrating a classification layer following the reconstruction layer, aimed at clustering inputs in an unsupervised manner. To ensure reliance solely on local, non-gradient-based learning, we utilize standard STDP synapses described by equation (1), in conjunction with LIF neurons. Furthermore, all-to-all inhibition is activated within this layer to establish a Winner-Takes-All setup. The neuron that exhibits the earliest firing in this layer is designated as the one classifying the input, and these responses are monitored during testing to ascertain classification accuracy.

To further assess the reconstruction capabilities of our auto-encoder, we employ the LeNet-5 [55] deep learning architecture as an alternative to STDP-based classification. In this approach, the spiking output of the reconstruction layer is translated back into a normalized image, which is subsequently fed into the deep learning classifier. This additional classifier demonstrates classification accuracy exceeding 99% for standard MNIST classification tasks—a performance level unattainable through conventional STDP methods. Hence, it provides a more comprehensive reflection of the usefulness of our auto-encoder reconstructions for downstream tasks.

4. MNIST results

To verify the reconstruction ability of our network, in the first instance, the MNIST dataset was used with the following experimental setup.

4.1. Training

To train the auto-encoder, the 60 000 training MNIST [55] images were used, with each image presented for 200 simulation timesteps. The learning hyperparameters, including A^+, A^-, τ_+, and τ_-, along with the neuron parameter β, as well as threshold regulation parameters A_{th} and τ_{th}, collectively play a crucial role in the network's ability to reconstruct the input. The optimization of these parameters was achieved through a grid search, and a summary of these hyperparameters for MNIST dataset can be found in table 1. The

Table 1. Overview of auto-encoder hyperparameters for the MNIST dataset.

	STDP hyperparameters						Neuron hyperparameters						
Layers	A^+	A^-	τ_+	τ_-	l	b	# of Neurons	β	A_{th}	τ_{th}	$\mathrm{Max}_{\mathrm{th}}$	$\mathrm{Min}_{\mathrm{th}}$	Scaling
Hidden	0.05	0.03	20	200			5000	0.9	0.05	5.00×10^{-08}	300	50	0.999
Reconstruction	0.05	0.005	30	200	10	0.5	784	0.9	0	0			
Classification	0.2	0.12	20	200			100	0.9	0.05	2.50×10^{-06}	300	50	0.9

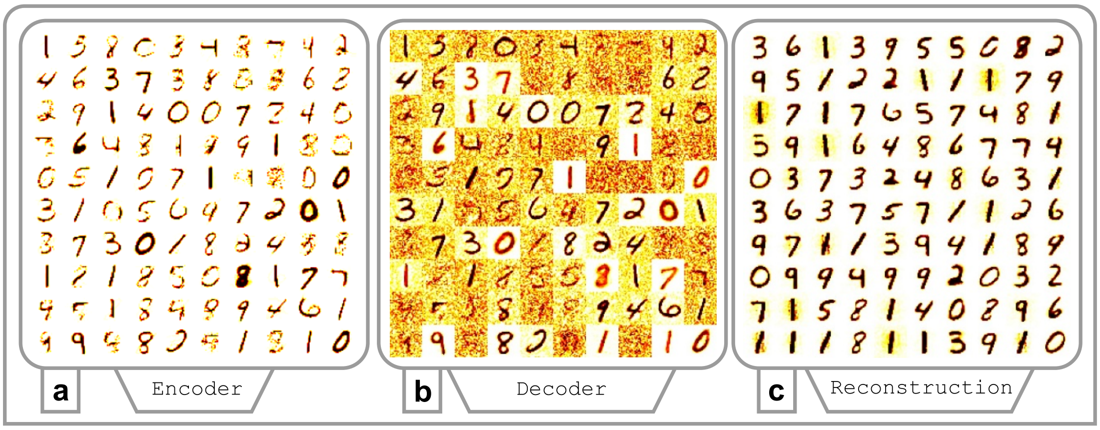

Figure 2. Receptive Field Analysis of MNIST Network in Three Layers: (a) 100 Standard STDP Synapses. (b) 100 Error-Modulated STDP Synapses. (c) 100 Standard STDP Synapses.

simulations were conducted using the snnTorch simulation platform [14]. It was also noted that clamping of the threshold voltage was required, as shown in the table 1 This was due to the network reaching extreme threshold values in the training phase, that resulted in significantly longer training time. Additionally, periodic scaling of the voltage thresholds was implemented, where, for every 1000 inputs, the threshold of all input and classification neurons was multiplied by a scaling factor less than one, as listed in the table 1.

It is noted that for the particular threshold adaptation scheme used, a relationship exists between A_{th} and τ_{th}, as defined by equation (7), where N represents the number of neurons and IT is the number of time-steps during which an input is presented

$$A_{\mathrm{th}} = N \times IT \times \tau_{\mathrm{th}}. \tag{7}$$

This is a notable observation, as the network aims to achieve a resource efficient scheme where all neurons participate in the learning phase. Whilst a TTFS scheme is adopted, it is necessary to monitor and regulate the activity in the hidden layer. This is to ensure that all neurons in the hidden layer spike throughout the training and testing.

Throughout the hyperparameter optimizations, the number of neurons in the hidden layer also played a significant role. For MNIST classification, it was found that 5000 neurons were adequate for the reconstruction of all 70 000 images in both the training and testing datasets. Reducing the number of hidden neurons resulted in some non-reconstructions, where sometimes no spiking activity in the reconstruction layer (decoder) was observed. Interestingly, this had a high correlation with no spiking in the hidden layer (encoder), suggesting that the features learnt by each neuron are highly specific.

4.2. Reconstructions

Following the training, the 10 000 images in the MNIST test set were then used to test the network's ability for image reconstruction. The weights and threshold regulation were held fixed during this testing procedure.

Figure 2 depicts the receptive fields associated with the input, reconstruction, and classification layer synapses. Upon inspection, it is evident that each neuron in the hidden layer corresponds to a specific digit, though some may pose challenges in recognition. Notably, the error-modulated STDP weights in the reconstruction layer exhibit fewer depressed weights compared to the standard synapses. Due to the lack of activity regularization in this layer, if the error-modulated synaptic weight is not high enough to cause post-synaptic firing, then no weight update will take place. The receptive field in the decoder layer was correlated with the receptive field in the encoder layer, as a result of the hidden neurons being representative of a particular digit. Subtle qualitative differences could also be observed between encoder and decoder

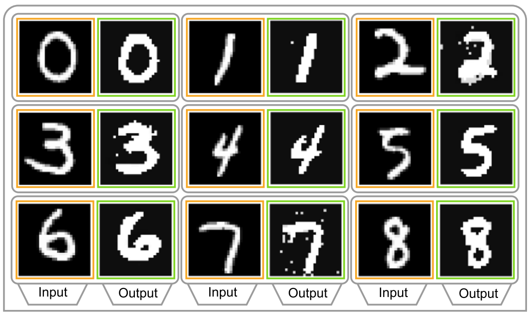

Figure 3. MNIST Image Reconstruction.

receptive fields, such as the existence of intermediate weights and the increased thickness of the decoder digits compared to the encoder digits. Both of these can be attributed to the error-modulated STDP learning rule, as the standard rule typically generates bimodal distribution of weights. Thus these pixels will continue to remain at these weights rather than depress further. Another observation is the fact that the receptive fields for the input and classification layers are notably different even though the synaptic learning rule is the same. Interestingly, this can be attributed to the fact that the classification layer learns the reconstructed images, which are clear synthetic versions of the input, as opposed to the original MNIST images presented to the hidden layer.

Figure 3 shows sample MNIST digit reconstructions. These results demonstrate an adequate reconstruction capability of the proposed auto-encoder. As already mentioned, a notable observation is the ability of the auto-encoder to utilize STDP for building a synthetic binary version of the input image, that does not try to replicate the exact shape of the input, rather it shows an overall shape for the input digit learned through seeing various cases of that digit during the encoding-decoding learning phase.

An additional advantage of the proposed network is its requirement for an average of only 9.8 spikes in the hidden layer for image reconstructions. This indicates an extremely efficient structure that encodes the input into merely 9.8 spikes. Moreover, considering the layer comprises 5000 neurons, it further highlights the remarkable sparsity achieved, where at most 10 neurons spike for each image. Detailed investigation of the network reconstruction ability showed that approximately 200 images presented in the testing phase did not produce any activity in the reconstruction layer. This is simply due to the very sparse nature of the proposed auto-encoder, which can result in not having enough features encoded in the hidden layer to lead to a synthetic class reconstruction. Despite this, the network still shows an impressive 98% MNIST test set reconstruction ability, while it learns to reconstruct all 60 000 training images in two training epochs.

4.3. Classification

The classification task of the proposed network yielded promising results, as listed in table 2. It should be noted that a direct comparison of classification performance is not straightforward, as all previous works in table 2, perform training on the actual 60 000 MNIST training images, and then test on the 10 000 test set. However, our network is trained for reconstruction and classification at the same time and performs classification of its reconstructed inputs, which as shown in figure 3 are synthetic class representative, rather than single digit representative.

In comparison to previous studies employing solely STDP for classification within a similar network architecture but utilizing a Poisson rate encoding scheme [49, 50], the attained accuracy of 72% falls slightly below. Nonetheless, despite this margin, several advantages justify the trade-off. Firstly, the adoption of a TTFS encoding scheme, rather than a rate-based approach, augments the information embedded within each spike while concurrently diminishing the need for a high number of spikes for network functionality. Additionally, synaptic weight updates occur solely once per image presentation, mitigating the memory overhead associated with storing the timing details of individual neurons. Lastly, the main objective of our study around minimizing the spike count for both image reconstruction and classification is achieved with an average of only 9.8 spikes for reconstruction and 2.49 for classification, significantly lower than previous works.

To further investigate the classification performance of our reconstructed MNIST images, the LeNet-5 architecture was used instead of the normal STDP classifier in the final layer. Initially, it was tested and achieved 99% accuracy on standard MNIST classification. However, when applied to classify image reconstructions, LeNet-5 achieved an accuracy of 83.2%. While this surpasses the previous accuracy of 72%

Table 2. Comparative Analysis of Reconstruction Classification Ability.

Dataset	Work	Input Encoding	Learning Rule / Supervision	Classification Network	Accuracy	Spike Number
MNIST	[21]	Poisson Rate Encoding	Weight-dependant STDP	784-6400	95%	8746 Input, 17 Hidden
	[46]	Poisson Rate Encoding	STDP	784-300	93.5%	Not Specified
	[47]	Poisson Rate Encoding	Pre-conditioned STDP	784-100	85.90%	8746 Input
	[48]	Poisson Rate Encoding	Weight-dependant STDP	400-100	89.15%	500 Input
	[49]	Poisson Rate Encoding	STDP	784-50	78.90%	200 Input
	[50]	Poisson Rate Encoding	STDP	784-100	80%	4312 Input
	[51]	Poisson Rate Encoding	Dopamine Assisted STDP	784-400-10	96.73%	8746 Input
	[52]	TTFS	Spike Timing Dependant Delay	784-4000	93.09%	100 Total
	[53]	TTFS	Weight-dependant STDP	784-100	88.57%	152 Total
	Proposed	TTFS	STDP	784-100	72%	784 Input, 2.49 Classification
	Proposed	TTFS	LeNet-5	784-(LeNet-5)	83.2%	784 Input
Fashion MNIST	[51]	Poisson Rate Encoding	Dopamine Assisted STDP	784-6400-10	85.30%	Not Specified
	[54]	Poisson Rate Encoding	STDP	784-1600	68.80%	Not Specified
	Proposed	TTFS	STDP	784-1000	51.25%	784 Input, 16.23 Classification
	Proposed	TTFS	LeNet-5	784-(LeNet-5)	66.2%	784 Input

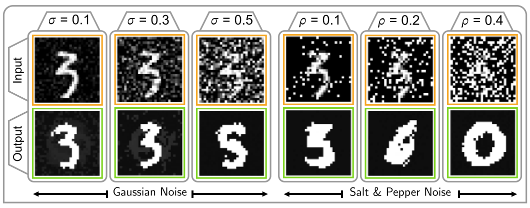

Figure 4. Image Reconstruction Across Different Levels of Gaussian Noise, and Salt & Pepper Noise.

using standard STDP, it falls short of its standard 99% accuracy. This is likely due to the feature extractor in the encoder, as explained in section 6. Nevertheless, the objective of the proposed study is not solely focused on attaining the highest classification performance. Instead, it aims to develop an auto-encoder with a minimal number of spikes capable of achieving competitive performance, a goal that has been accomplished.

4.4. De-noising

Image de-noising represents a significant application of auto-encoders. We investigated our network's ability to reconstruct images under the Salt & Pepper noise as well as Gaussian noise, and assessed the potential implications for downstream tasks. Figure 4 shows that the output images can be reconstructed accurately

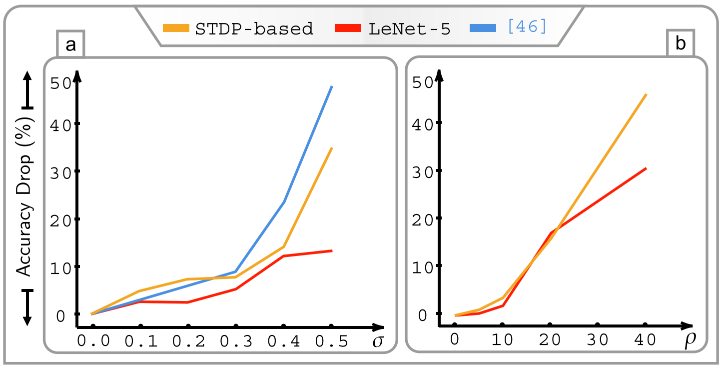

Figure 5. Classification Accuracy of Network with Noisy MNIST Inputs: (a) Gaussian Noise. (b) Salt & Pepper Noise.

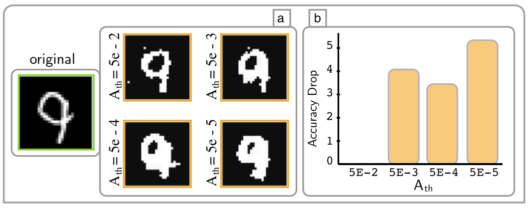

Figure 6. The effect of increased hidden layer activity on network reconstruction. (a) Qualitative evaluation of reconstruction for different activity levels. (b) Classification accuracy drop in % with various activity levels..

when the input is subjected to various levels of noise. To introduce Gaussian noise to input images, we normalized the MNIST image within the range of 0 to 1 and then added a random variable sampled from a normal distribution. The standard deviations of these distributions are denoted as σ values. As shown in the response to Gaussian noise, the network exhibits an ability to identify prominent features within the noise and undertake reconstruction efforts. Salt & Pepper noise was introduced by determining the probability of a pixel being affected and toggled to either the maximum or minimum value, as shown by the ρ values. Again, feature extraction is demonstrated even in considerably noisy inputs.

Figure 5 shows the classification accuracy drop for the STDP-based and the LeNet-5 classifiers used in our work as well as the accuracy drop presented in [56], the only work clearly describing the same method of Gaussian noise injection. Figure 5(a) shows that under various levels of Gaussian noise, the STDP-based classifier remains within 10% of its original accuracy for $\sigma \leqslant 0.3$. The LeNet-5 operated slightly better, exhibiting less accuracy drop than both the STDP-based classifier and [56]. Even though LeNet-5 achieves higher accuracy, it demonstrates a lower accuracy drop, suggesting that the MNIST reconstructions are relatively stable even when high levels of Gaussian noise are introduced.

Similarly, figure 5(b) shows similar stability for both the STDP-based and LeNet-5 classifiers under Salt & Pepper noise conditions. Significant performance degradation for the STDP-based classifier was noted when the probability of Salt & Pepper noise was increased to above 20%. LeNet-5 also had notable degradation at 20%, suggesting that the reconstructions have degraded at this level of noise. It also suggests that the reconstruction network is more tolerable to Gaussian noise as opposed to Salt & Pepper noise. As previously indicated in figure 4, the visual representation provides insight into why this may be the case, as more than 20% of Salt & Pepper noise clearly affects the pattern presentation.

4.5. Activity analysis

In this section, we describe the effect of increased activity on neuronal performance. To achieve this, the A_{th} parameter in the hidden layer was decreased to promote increased firing rates (whilst keeping τ_{th} fixed.) figure 6 shows an example of the qualitative effect of increased activity on the network reconstructions. Whilst this form of degradation was not observed in all cases, some reconstructions were clearly more affected. It is believed that the increased activity results in less specificity in the hidden layer, causing more generic and overlapped reconstructions. Figure 6 highlights the drop in classification (in %) accuracy as a result of increased activity, which can be attributed to the poorer quality of some reconstructions.

Table 3. Overview of Hyperparameters Variation for Fashion-MNIST Dataset.

	STDP Hyperparameters						Neuron Hyperparameters						
Layers	A^+	A^-	τ_+	τ_-	l	b	# of Neurons	β	A_{th}	τ_{th}	Max_{th}	Min_{th}	Scaling
Hidden	0.05	0.02	20	200			10000	0.9	0.05	2.5×10^{-08}	300	50	0.999
Reconstruction	0.1	0.001	30	200	10	0.15	784	0.9	0	0			
Classification	0.05	0.04	5	200			1000	0.9	1	0.00001	300	50	0.99

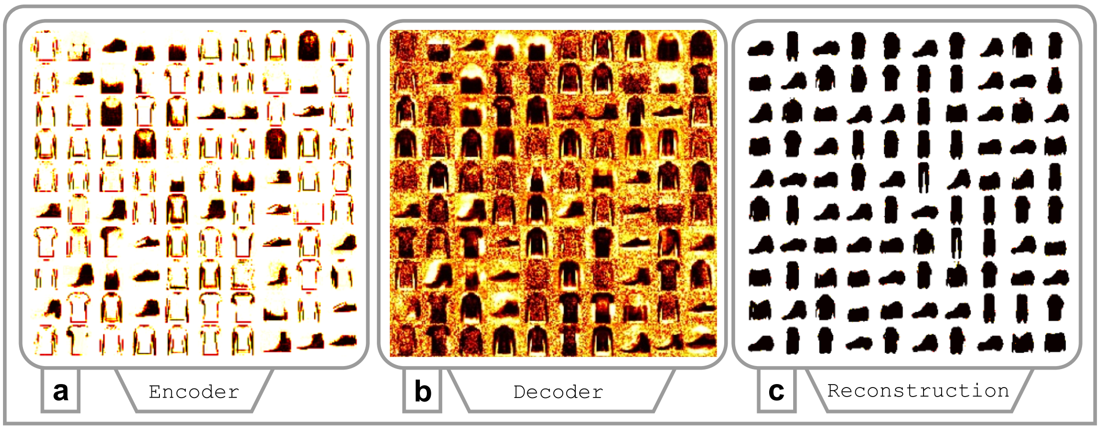

Figure 7. Receptive Fields Analysis of Fashion-MNIST Network in Three Layers: (a) 100 Standard STDP Synapses. (b) 100 Error-Modulated STDP Synapses. (c) 100 Standard STDP Synapses.

5. Fashion-MNIST results

5.1. Training

The next set of simulations involved using a more complex dataset known as Fashion-MNIST, featuring 28×28 grayscale images representing diverse clothing items such as shirts, trousers, pants, and more. Similar to MNIST dataset, this dataset consists of 10 different categories or labels, with the images divided into a 60 000-sample training set and a 10 000-sample test set. The hyperparameters used for training this network are listed in table 3. Like the previous section, these hyperparameters were optimized through a grid search. One of the most notable differences from previous was the increase in the number of hidden layer neurons. The more complex nature of Fashion-MNIST required more hidden layer neurons to extract more specific features. Consequently, the average activity of each neuron in the hidden layer is decreased, which is reflected in the voltage regulation parameters.

Figure 7 illustrates the learned weights between each layer when the network was trained using the Fashion-MNIST dataset. The most prominent features learned in these fields are the outlines of the inputs, such as shoes or pants. However, the details within these outlines were not as distinct. One notable contrast in these fields compared to figure 2 was the significant difference in weights between the Reconstruction-Classification layer and the Input-Hidden layer, despite both being trained using the same standard STDP rule. This implies that the image reconstructions have a considerable influence on the learning process. The relationship between the receptive field of the encoder and decoder layers was markedly different for Fashion-MNIST than for MNIST. The encoder layer typically learnt the outlines of objects. This is attributed to the bi-modal distribution of weights favoring depression over potentiation, and whilst subsequent attempts at improving potentiation via STDP window parameter adjustment could rectify this result, the reconstruction and classification results were poorer. This result was not observed in the decoder layer, as the decoder is able to learn intermediate weights. Finally, qualitative observation shows that the receptive fields for the encoder and decoder layers are correlated, as expected.

5.2. Reconstruction

Figure 8 illustrates that the attempted reconstructions for the Fashion-MNIST dataset are rough approximations of their original inputs, noting that even the original inputs are not of high quality. The initial observation of these reconstructions is the reduction in quality from input to reconstruction. This degradation can be attributed to the TTFS input utilized for the network. To achieve more precise

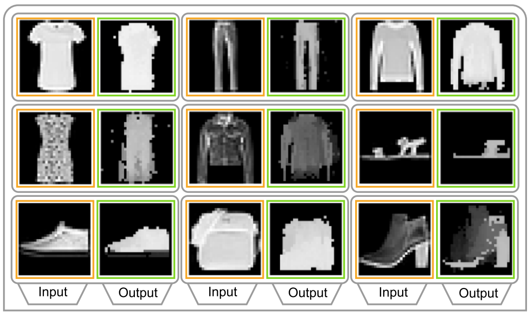

Figure 8. Fashion-MNIST Image Reconstructions.

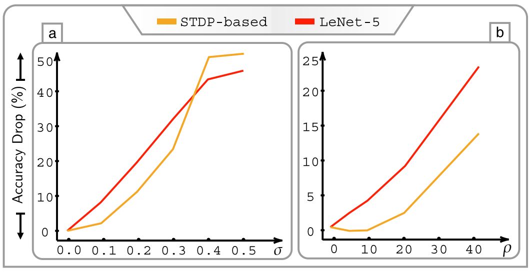

Figure 9. Classification Accuracy of Network with Noisy Fashion-MNIST Inputs: (a) Gaussian Noise Impact. (b) Salt & Pepper Noise Impact.

reconstructions, it becomes essential for the reconstruction layer to finely adjust the individual timings of each neuron to synchronize with each pixel. However, due to its sparse firing from the hidden layer, controlling the timings of each neuronal firing becomes considerably challenging. Nonetheless, this can be mitigated by implementing a learned delay mechanism. This mechanism would introduce a delay in transmitting spikes from the hidden layer to the reconstruction layer, thus facilitating the adjustment of neuronal event timings in the reconstruction layer. It is worth noting that, compared to MNIST needing an average of 9.8 spikes in its encoding component, the average activity of the hidden layer for Fashion-MNIST images was higher at 68.2 spikes per input, which is expected considering the more complex input shape.

5.3. Classification

The outcomes of the Fashion-MNIST classification task fell significantly short. The highest classification accuracy attained using only STDP was 51.25%. However, when the classifier was replaced with LeNet-5, the accuracy increased to 62.3%. One factor contributing to the lower accuracy was the utilization of STDP learning in the classification layer. As listed in table 2, unsupervised STDP learning for Fashion-MNIST is seldom used and shows inferior performance compared to other learning rules. However, further investigation revealed that the attempted reconstructions also play a role in the downgrade in classification accuracy.

5.4. De-noising

Figure 9 demonstrates the network's accuracy drop for various levels of Gaussian noise, and Salt & Pepper noise for Fashion-MNIST images. Compared to MNIST classification, the accuracy drop was much less stable, particularly for higher levels of additional noise. This effect was expected, because the added complexity of the dataset means that the reconstructions needed to contain more temporal information. Figure 9 also illustrates that LeNet-5 exhibits a higher accuracy drop than the STDP-based classifier. This is largely because the LeNet-5 classifier achieved a higher baseline accuracy, thus allowing more room for accuracy drop.

6. Discussion

Analysis of the network performance highlights how the network performs its task. One notable observation, particularly in the Fashion-MNIST dataset is the synthetic reconstruction produced, as opposed to attempting to exactly replicate the input. The synthetic reconstruction is formed through clustering in the

Table 4. Comparison of Spiking Auto-Encoders for MNIST/Fashion MNIST Image Reconstruction.

Dataset	Work	Input Encoding	Learning Rule / Supervision	Network Structure	Reconstruction Ability	Spike Number
MNIST	[30]	Poisson Rate Encoding	Membrane Potential Backpropagation	Convolutional	6.75 (MSE[a])	≈2000
	[35]	Synaptic Current Encoding	Mirrored STDP	784 — 5000	0.5 (PC)	≈150
	[34]	Spike Probability Encoding	Engineered STDP	Convolutional	0.01 (PC)	10000
	[2]	Direct Input Rate Encoding	Spatiotemporal Backpropagation	Convolutional	0.031 (MSE)	Not Specified
	Proposed	TTFS	STDP/Error modulated STDP	784-5000-784	56.65 (MSE)	9.8
Fashion MNIST	[2]	Direct Input Rate Encoding	Spatiotemporal Backpropagation	Convolutional	0.031 (MSE)	Not Specified
	[31]	Poisson Rate Encoding	Backpropagation Through Time	784-400-200-400-784	11.68 (MSE)	Not Specified
	[32]	TTFS	Spike Time Backpropagation	784-32-784	0.01 (MSE)	10^4
	Proposed	TTFS	STDP/Error modulated STDP	784-10000-784	44.13 (MSE)	68.2

[a] Mean Squared Error (MSE)

hidden layer, where various inputs are grouped based on their characteristics. The network then attempts reconstructions based on which group/neurons in the hidden layer are attributed to it.

Our auto-encoder network operates differently than the previously investigated auto-encoders, making a direct comparison challenging. Nonetheless, table 4 lists previous relevant works, which mostly implement some form of rate-based input encoding, which significantly increases the number of spikes required for reconstruction. Furthermore, these works utilize more complex learning rules, compared to our design using only STDP, while also relying on convolutional or more complex network architectures.

Interestingly, the only work found to use a TTFS encoding scheme [32] was also the work that required the most amount of spikes, due to the additional synchronization pulses in their network. One work not included in table 4 was [33]. Tavanaei *et al* [33] performed a recreation of patches of MNIST images, using STDP learning, and reported utilizing approximately 115 spikes to achieve a reconstruction Root Mean Square (RMS) error of 0.167. This work was not included based on its patch learning scheme, which makes it difficult to compare to other auto-encoder works.

The only other auto-encoders to utilize STDP learning are [34, 35]. Kotariya *et al* [34] modified the learning rules proposed in [36, 37] such that their learning algorithm is explicitly spike-based and temporally local. However, their unique input encoding method is equivalent to a rate-based learning rule, and they report representing each MNIST digit with 10 000 spikes, which is significantly higher than the 9.8 spikes needed for our proposed auto-encoder. Burbank *et al* [35] proposed a mirrored STDP learning rule, where feed-forward and feedback connections are updated using the same STDP rule. They utilized a unique network structure where feed-forward and feedback connections between a visible and hidden layer are used to generate input reconstruction. It is estimated that for MNIST reconstruction, they utilize approximately 150 spikes (excluding inhibitory neuron spikes) to attain a Pearson Correlation (PC) of 0.5 between input and reconstruction. However, image pre-processing was required to generate these results. As listed in table 4, no works have reported using less than 9.8 spikes in the hidden layer, indicating that our approach achieves minimal spike encoding. Naturally, this reduction in spikes inevitably leads to significantly higher reconstruction losses in our work. However, we intend to demonstrate that an exact replication of the input is not necessary in downstream tasks such as classification.

Table 5. Power consumption comparison for various spiking auto-encoders working on MNIST reconstruction..

	Work	Input (J) Generation	Input (J) Routing	Hidden (J) Generation	Hidden (J) Routing	Total (J)
Training	[30]	3.00×10^{-02}	9.78×10^{-03}	5.88×10^{-03}	1.92×10^{-03}	4.75×10^{-02}
	[34]	3.00×10^{-02}	9.78×10^{-03}	5.88×10^{-02}	1.92×10^{-02}	1.18×10^{-01}
	[35]*	$\mathbf{2.45 \times 10^{-04}}$	$\mathbf{8.00 \times 10^{-05}}$	3.68×10^{-05}	1.20×10^{-05}	$\mathbf{3.74 \times 10^{-04}}$
	Proposed	1.61×10^{-03}	5.27×10^{-04}	$\mathbf{1.23 \times 10^{-05}}$	$\mathbf{4.02 \times 10^{-06}}$	2.16×10^{-03}
Testing	[30]	4.99×10^{-04}	1.63×10^{-04}	9.80×10^{-05}	3.20×10^{-05}	7.92×10^{-04}
	[34]	2.50×10^{-04}	8.15×10^{-05}	4.90×10^{-04}	1.60×10^{-04}	9.81×10^{-04}
	[35]*	$\mathbf{2.45 \times 10^{-05}}$	$\mathbf{8.00 \times 10^{-06}}$	7.35×10^{-06}	2.40×10^{-06}	$\mathbf{4.22 \times 10^{-05}}$
	Proposed	3.84×10^{-05}	1.25×10^{-05}	$\mathbf{4.58 \times 10^{-07}}$	$\mathbf{1.50 \times 10^{-07}}$	5.16×10^{-05}

*This work downsamples the MNIST images, and the energy consumption of this process was not considered.

Compared to other studies that have performed either MNIST or Fashion-MNIST classification, our overall classification accuracies are slightly lower. For MNIST classification, our final accuracy for the STDP-based classifier was only 72%, compared to other works which typically achieve $>78\%$ using STDP, as shown in table 2. Interestingly, when using LeNet-5 for classification, it is comparable to these other networks with a classification accuracy of 83.2%. Given that LeNet-5 typically operates at around 99% accuracy for typical MNIST, it is evident that the reconstructions have marginally reduced the quality of the images generated. However, the reconstructions still retain adequate information such that classification is on par with typical STDP classification.

Fashion-MNIST classification was also lower than other works that have attempted the same task. As previously mentioned, backpropagation-based learning rules tend to outperform STDP learning rules and typically achieve classification accuracies between 80% and 90% [22, 56–59]. As shown in table 2, few works have attempted Fashion-MNIST classification with only STDP-based learning rules, but none has used TTFS along with STDP. In [51], dopamine assisted STDP achieved a classification accuracy of 85.3%. However, their particular STDP learning is supervisory in nature, and requires labeled datasets to achieve this result, whereas our work falls under the self-supervisory category and does not require labels to train. Lastly, [54] utilized STDP learning to achieve a 68.8% classification accuracy. Remarkably, this is similar to our result of 66.2% using LeNet-5 in the classifier layer. Our reconstruction network only uses STDP learning rules, so it suggests that the data contained in our image reconstructions contains similar features to those extracted in [54]. As expected, our STDP-based classifier performed worse than the LeNet-5 classifier, highlighting the current disparity between unsupervised neuromorphic and deep-learning approaches.

Table 2 also illustrates the number of spikes used in the prior art for encoding inputs for classification. This data is limited by the lack of reporting on spiking activities. For some rate-based inputs, we have approximated the input spiking behavior by using the rates and time lengths provided by each work. A direct comparison is not possible, because the previous works have not done reconstruction before classification. For our STDP-based classifiers, the spikes required for classification are 2.49 and 16.23 for MNIST and Fashion-MNIST, respectively. This further reflects the sparsity and efficiency of our network, albeit at the expense of a lower accuracy.

On top of monitoring the classification accuracy of the overall architecture that reconstructs the inputs before classifying them, we also monitored the hidden layer's responsiveness to particular inputs. When these responses are clustered, classification accuracy just within the hidden layer can be reported. When this was monitored in the test set of MNIST, the 5000 neurons in the hidden layer achieved an 87.04% classification accuracy. Similarly, the $10,000$ neurons for Fashion-MNIST classification achieved 72.4%. Clearly, these classification results are higher than our other STDP results reported in table 2. However, compared to the results obtained with the LeNet-5 classifier, these results are only slightly reduced, and much more comparable. Thus, this indicates that the reduced performance in classification accuracy can be attributed to the imperfect nature of the feature extraction in the hidden layer.

table 5 compares the calculated power consumption in both the training and testing phases of various spiking auto-encoders on MNIST reconstruction. To calculate these values, we considered both the energy required to generate spikes, as well as the energy required to transmit those spikes within the network. For each spike, we estimated that it takes approximately 4.9 pJ to generate [60] and 1.6 pJ to transmit those spikes [61]. Initially, we also considered the energy required for synaptic weight update, however this data was not reported for previous auto-encoders. As shown, in both the input and hidden layers, our network consumes less power, except when compared to [35]. However, [35] downsamples the image to nearly half of its original size. The energy consumption of this process was not considered. We have also compared our total network latency to previous auto-encoders. We use 200 timesteps to encode each image, which is slightly poorer than the values of 50 timesteps and 100 timesteps reported in [34] and [30], respectively. [35] reported a value of 65 ms for training and 30 ms for testing for each image.

Whilst we have demonstrated a highly efficient and sparse auto-encoder, using simple local STDP learning to perform input reconstruction, our auto-encoder has some drawbacks. Its most significant limitation is the lack of temporal resolution in the reconstruction layer. The minimal spiking in the hidden results in the reconstruction layer being stimulated very sparsely, which while making our network very attractive from an energy point of view, restricts the information transmitted to the next layer. An approach that could be investigated in future works is to produce many spikes at many different times, using an adjustable delay or utilizing some other proxy for time.

Furthermore, the scalability of the network's learning rule heavily favors TTFS encoding schemes. Whilst this can encompass a wide variety of data, future studies could apply it to event-based datasets that utilize the temporal dimension of input data more effectively. Furthermore, the standard STDP learning rule used in our work could be improved, to more precisely learn the temporal structures presented as opposed to the bi-modal distribution observed in our receptive fields.

Our error modulated learning rule in its current form is only applicable for decoders aiming to reconstruct the original input. However, in some cases, it may be applicable to reconfigure the decoder so that the output is more suitable for downstream tasks. For example, [5] investigated audio to image conversion using spiking auto-encoders. Currently, our error-modulated learning rule does not support this, however, a similar self-supervised approach could be explored for future investigation.

7. Conclusion

In our work, we performed self-supervised learning using a novel form of error-modulated STDP. We determined that with enough hidden neurons, we were able to reconstruct some images within the MNIST and Fashion-MNIST datasets. Furthermore, we demonstrated our network's tolerance to various types and levels of noise. Our proposed network only requires an average of 9.8 spikes in its hidden layer to reconstruct MNIST images and 68.2 spikes to reconstruct Fashion-MNIST images. This is remarkable and shows potential for very low-power image processing on spiking hardware in future works.

Data availability statement

NA The data that support the findings of this study are available upon reasonable request from the authors. https://github.com/jc427648/Efficient-Sparse-Spiking-Auto-Encoder-for-Reconstruction-Denoising-and-Classification.

ORCID iDs

Ben Walters https://orcid.org/0000-0001-5464-8468
Jason Eshraghian https://orcid.org/0000-0002-5832-4054
Mostafa Rahimi Azghadi https://orcid.org/0000-0001-7975-3985

References

[1] Huang S-C, Pareek A, Jensen M, Lungren M P, Yeung S and Chaudhari A S 2023 Self-supervised learning for medical image classification: a systematic review and implementation guidelines *npj Digit. Med.* **6** 74

[2] Kamata H, Mukuta Y and Harada T 2022 Fully spiking variational autoencoder *Proc. AAAI Conf. on Artificial Intelligence* vol 36 pp 7059–67

[3] Kascenas A, Pugeault N and O'Neil A Q 2022 Denoising autoencoders for unsupervised anomaly detection in brain MRI *Proc. 5th Int. Conf. on Medical Imaging With Deep Learning* (*Proc. Machine Learning Research*) vol 172, ed E Konukoglu, B Menze, A Venkataraman, C Baumgartner, Q Dou and S Albarqouni (avaiable at: https://proceedings.mlr.press/v172/kascenas22a.html) pp 653–64

[4] Czyżewski A, Kurowski A and Zaporowski S 2019 Application of autoencoder to traffic noise analysis *Proc. Meetings Acoust.* **39** 055003

[5] Roy D, Panda P and Roy K 2019 Synthesizing images from spatio-temporal representations using spike-based backpropagation *Front. Neurosci.* **13** 621

[6] Nguyen T N, Veeravalli B and Fong X 2022 Hardware implementation for spiking neural networks on edge devices *Predictive Analytics in Cloud, Fog and Edge Computing: Perspectives and Practices of Blockchain, IoT and 5G* (Springer) pp 227–48

[7] Xue J, Xie L, Chen F, Wu L, Tian Q, Zhou Y, Ying R and Liu P 2023 EdgeMap: an optimized mapping toolchain for spiking neural network in edge computing *Sensors* **23** 6548

[8] Sironi A, Brambilla M, Bourdis N, Lagorce X and Benosman R 2018 HATS: histograms of averaged time surfaces for robust event-based object classification *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 1731–40

[9] Gehrig D, Loquercio A, Derpanis K G and Scaramuzza D 2019 End-to-end learning of representations for asynchronous event-based data *Proc. IEEE/CVF Int. Conf. on Computer Vision* pp 5633–43
[10] Zhang P, Wang C and Lam E Y 2024 Neuromorphic imaging and classification with graph learning *Neurocomputing* **565** 127010
[11] Yin N, Wang M, Chen Z, De Masi G, Xiong H and Gu B 2024 Dynamic spiking graph neural networks *Proc. AAAI Conf. on Artificial Intelligence* vol 38 pp 16495–503
[12] Neftci E O, Mostafa H and Zenke F 2019 Surrogate gradient learning in spiking neural networks: bringing the power of gradient-based optimization to spiking neural networks *IEEE Signal Process. Mag.* **36** 51–63
[13] Bohte S M, Kok J N and La Poutré H 2002 Error-backpropagation in temporally encoded networks of spiking neurons *Neurocomputing* **48** 17–37
[14] Eshraghian J K, Ward M, Neftci E O, Wang X, Lenz G, Dwivedi G, Bennamoun M, Jeong D S and Lu W D 2023 Training spiking neural networks using lessons from deep learning *Proc. IEEE* **111** 1016–54
[15] Frenkel C and Indiveri G 2022 ReckOn: a 28nm sub-mm2 task-agnostic spiking recurrent neural network processor enabling on-chip learning over second-long timescales *2022 IEEE Int. Solid- State Circuits Conf. (ISSCC)* vol 65 pp 1–3
[16] Quintana F M, Perez-Peña F, Galindo P L, Netfci E O, Chicca E and Khacef L 2023 ETLP: event-based three-factor local plasticity for online learning with neuromorphic hardware (arXiv:2301.08281)
[17] Markram H, Gerstner W and Sjöström P J 2012 Spike-timing-dependent plasticity: a comprehensive overview *Front. Synaptic Neurosci.* **4** 2
[18] Bi G-Q and Poo M-M 1998 Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength and postsynaptic cell type *J. Neurosci.* **18** 10464–72
[19] Rahimi Azghadi M, Iannella N, Al-Sarawi S F, Indiveri G and Abbott D 2014 Spike-based synaptic plasticity in silicon: design, implementation, application and challenges *Proc. IEEE* **102** 717–37
[20] 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
[21] Diehl P and Cook M 2015 Unsupervised learning of digit recognition using spike-timing-dependent plasticity *Front. Comput. Neurosci.* **9** 99
[22] Cai Z, Kalatehbali H R, Walters B, Azghadi M R, Amirsoleimani A and Genov R 2023 Spike timing dependent gradient for direct training of fast and efficient binarized spiking neural networks *IEEE J. Emerg. Sel. Top. Circuits Syst.* **13** 1083–93
[23] Deng L, Wu Y, Hu X, Liang L, Ding Y, Li G, Zhao G, Li P and Xie Y 2020 Rethinking the performance comparison between SNNS and ANNS *Neural Netw.* **121** 294–307
[24] Bekolay T and Eliasmith C 2011 A general error-modulated STDP learning rule applied to reinforcement learning in the basal ganglia 01
[25] Legenstein R, Pecevski D, Maass W and Graham L J 2008 A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback *PLoS Computat. Biol.* **4** 1–27
[26] Lu H, Liu J, Luo Y, Hua Y, Qiu S and Huang Y 2021 An autonomous learning mobile robot using biological reward modulate STDP *Neurocomputing* **458** 308–18
[27] Bing Z, Meschede C, Huang K, Chen G, Rohrbein F, Akl M and Knoll A 2018 End to end learning of spiking neural network based on R-STDP for a lane keeping vehicle *2018 IEEE Int. Conf. on Robotics and Automation (ICRA)* pp 4725–32
[28] Olshausen B A and Field D J 2006 What is the other 85 percent of V1 doing
[29] Walters B 2024 Efficient-Sparse-Spiking-Auto-Encoder-for-Reconstruction-Denoising-and-Classification (available at: https://github.com/jc427648/Efficient-Sparse-Spiking-Auto-Encoder-for-Reconstruction-Denoising-and-Classification)
[30] Hübotter J F, Lanillos P and Tomczak J M 2021 Training deep spiking auto-encoders without bursting or dying neurons through regularization (arXiv:2109.11045)
[31] Shimmyo Y Okuyama Y and Abdallah A B 2022 Training spiking autoencoders by truncated BPTT under trade-offs between simulation steps and reconstruction error *2022 IEEE Int. Conf. on Development and Learning (ICDL)* pp 293–8
[32] Comşa I-M, Versari L, Fischbacher T and Alakuijala J 2021 Spiking autoencoders with temporal coding *Front. Neurosci.* **15** 712667
[33] Tavanaei A, Masquelier T and Maida A 2018 Representation learning using event-based STDP *Neural Netw.* **105** 294–303
[34] Kotariya V, Biswas A and Ganguly U 2023 E-STDP: a spatio-temporally local unsupervised learning rule for sparse coded spiking convolutional autoencoders *2023 Int. Joint Conf. on Neural Networks (IJCNN)* pp 1–7
[35] Burbank K S and Graham L J 2015 Mirrored STDP implements autoencoder learning in a network of spiking neurons *PLoS Computat. Biol.* **11** 1–25
[36] Bhatt V and Ganguly U 2018 Sparsity enables data and energy efficient spiking convolutional neural networks *Artificial Neural Networks and Machine Learning – Icann 2018*, ed V Kůrková, Y Manolopoulos, B Hammer, L Iliadis and I Maglogiannis (Springer International Publishing) pp 263–72
[37] Zylberberg J, Murphy J T, DeWeese M R and Sporns O 2011 A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of V1 simple cell receptive fields *PLoS. Biol.* **7** 1–12
[38] Bensimon M, Greenberg S and Haiut M 2021 Using a low-power spiking continuous time neuron (SCTN) for sound signal processing *Sensors* **21** 1065
[39] Bensimon M, Greenberg S, Ben-Shimol Y and Haiut M 2021 A new SCTN digital low power spiking neuron *IEEE Trans. Circuits Syst. II* **68** 2937–41
[40] Masquelier T 2018 STDP allows close-to-optimal spatiotemporal spike pattern detection by single coincidence detector neurons *Neuroscience* **389** 133–40
[41] Diehl P U and Cook M 2014 Efficient implementation of STDP rules on spinnaker neuromorphic hardware *2014 Int. Joint Conf. on Neural Networks (IJCNN)* (IEEE) pp 4288–95
[42] Frenkel C 2021 Sparsity provides a competitive advantage *Nat. Mach. Intell.* **3** 742–3
[43] Rahimi Azghadi M, Iannella N, Al-Sarawi S, Abbott D and Wennekers T 2014 Tunable low energy, compact and high performance neuromorphic circuit for spike-based synaptic plasticity *PLoS One* **9** e88326
[44] Scharstein H 1979 Input-output relationship of the leaky-integrator neuron model *J. Math. Biol.* **8** 403–20
[45] Shrestha A, Fang H, Wu Q and Qiu Q 2019 Approximating back-propagation for a biologically plausible local learning rule in spiking neural networks *Proc. Int. Conf. on Neuromorphic Systems* pp 1–8
[46] Querlioz D, Bichler O, Dollfus P and Gamrat C 2013 Immunity to device variations in a spiking neural network with memristive nanodevices *IEEE Trans. Nanotechnol.* **12** 288–95
[47] Tao T, Li D, Ma H, Li Y, Tan S, xiao Liu E, Schutt-Aine J and Li E-P 2023 A new pre-conditioned STDP rule and its hardware implementation in neuromorphic crossbar array *Neurocomputing* **557** 126682

[48] Demin V, Nekhaev D, Surazhevsky I, Nikiruy K, Emelyanov A, Nikolaev S, Rylkov V and Kovalchuk M 2021 Necessary conditions for STDP-based pattern recognition learning in a memristive spiking neural network *Neural Netw.* **134** 64–75
[49] Guo Y, Wu H, Gao B and Qian H 2019 Unsupervised learning on resistive memory array based spiking neural networks *Front. Neurosci.* **13** 812
[50] Walters B, Lammie C, Yang S, Jacob M V and Rahimi Azghadi M 2023 Unsupervised character recognition with graphene memristive synapses *Neural Comput. Appl.* **36** 1569–84
[51] Hao Y, Huang X, Dong M and Xu B 2020 A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule *Neural Netw.* **121** 387–95
[52] Hazan H, Caby S, Earl C, Siegelmann H and Levin M 2022 Memory via temporal delays in weightless spiking neural network (https://doi.org/10.48550/arXiv.2202.07132)
[53] Guo W, Fouda M E, Eltawil A M and Salama K N 2021 Neural coding in spiking neural networks: a comparative study for robust neuromorphic systems *Front. Neurosci.* **15** 638474
[54] Putra R V W and Shafique M 2020 FSpiNN: an optimization framework for memory-efficient and energy-efficient spiking neural networks *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **39** 3601–13
[55] Lecun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition *Proc. IEEE* **86** 2278–324
[56] Cheng X, Hao Y, Xu J and Xu B 2020 LISNN: improving spiking neural networks with lateral interactions for robust object recognition (Yokohama) 1519–25
[57] Zhang W and Li P 2020 Temporal spike sequence learning via backpropagation for deep spiking neural networks *Adv. Neural Inf. Process. Syst.* **33** 12022–33
[58] Opiełka P, Starczewski J T, Wróbel M, Nieszporek K and Marchlewska A 2019 *Application of Spiking Neural Networks to Fashion Classification* (Springer) pp 172–80
[59] Kheradpisheh S R, Mirsadeghi M and Masquelier T 2022 BS4NN: binarized spiking neural networks with temporal coding and learning *Neural Process. Lett.* **54** 1255–73
[60] Kang S M *et al* 2021 How to build a memristive integrate-and-fire model for spiking neuronal signal generation *IEEE Trans. Circuits Syst. I* **68** 4837–50
[61] Dalgaty T, Moro F, Demirağ Y, De Pra A, Indiveri G, Vianello E and Payvand M 2024 Mosaic: in-memory computing and routing for small-world spike-based neuromorphic systems *Nat. Commun.* **15** 142