Beyond weight plasticity: Local learning with propagation delays in spiking neural networks

Jørgen Jensen Farner [1] **Ola Huse Ramstad** [1 2] **Stefano Nichele** [* 1 3] **Kristine Heiney** [* 1 4]

Abstract

We propose a novel local learning rule for spiking neural networks in which spike propagation times undergo activity-dependent plasticity. Our plasticity rule aligns pre-synaptic spike times to produce a stronger and more rapid response. Inputs are encoded by latency coding and outputs decoded by matching similar patterns of output spiking activity. We demonstrate the use of this method in a three-layer feed-foward network with inputs from a database of handwritten digits. Networks consistently showed improved classification accuracy after training, and training with this method also allowed networks to generalize to an input class unseen during training. Our proposed method takes advantage of the ability of spiking neurons to support many different time-locked sequences of spikes, each of which can be activated by different input activations. The proof-of-concept shown here demonstrates the great potential for local delay learning to expand the memory capacity and generalizability of spiking neural networks.

1. Introduction

The brain has a great capacity for learning and memory, and the mechanisms that allow it to reliably and flexibly store information can provide new foundational mechanisms for learning in artificial networks. Perhaps the most widely discussed mechanism associated with learning is Hebbian plasticity (Hebb, 1949; Markram et al., 2011). This theory on neural learning states that when one neuron causes repeated excitation of another, the efficiency with which the first cell excites the second is increased.

The basic idea underlying Hebbian mechanisms is the brain's ability to change: local activity changes how neurons in a network communicate with each other, in turn affecting the overall behavior. In Hebbian plasticity, these changes are to the strength of connections between neurons. However, experimental observations (Bucher & Goaillard, 2011; Grossman et al., 1979; Hatt & Smith, 1976; Lüscher et al., 1994) have demonstrated that local activity can affect not only the *strength* of connections but also the *speed* with which action potentials travel between neurons. This alteration in transmission delays is likely an inherent part of how the brain learns and stores memories, as encoding information in time-locked sequences expands the computational capacity of a network (Izhikevich, 2006).

Local plasticity rules, such as spike-timing-dependent plasticity (STDP) (Markram et al., 1997), that change synaptic weights in an activity-dependent manner are of great interest in the context of unsupervised deep learning in deep spiking neural networks (SNNs) (Tavanaei et al., 2019). But why should plasticity in SNNs be confined to synaptic weights, when we are aware of a much richer repertoire of plastic changes that occur in the brain (Gittis & du Lac, 2006; Zhang & Linden, 2003)? In particular, there is evidence that neurons may change the speed of spike transmission in an activity-dependent manner (Lin & Faber, 2002; Debanne, 2004). This type of delay plasticity would allow networks to encode information and learn using spike times, and a similar type of learning could be translated to neuromorphic event-based hardware (Taherkhani et al., 2020; Grimaldi et al., 2022). Delay plasticity in neural networks has been explored, but the majority of studies have used supervised methods (Schrauwen & van Campenhout, 2004; Wang et al., 2019; Taherkhani et al., 2015; Johnston et al., 2006), with one noteworthy study using an unsupervised method to train only the readout layer of a reservoir (Paugam-Moisy et al., 2008). Supervised methods come with many drawbacks, including high requirements for memory and less flexibility for real-time applications. The development of a local delay learning rule that uses time-based coding would allow

*Co-senior authors [1]Department of Computer Science, Oslo Metropolitan University, Oslo, Norway [2]Department of Neuromedicine and Movement Science, Norwegian University of Science and Technology, Trondheim, Norway [3]Department of Computer Science and Communication, Østfold University College, Halden, Norway [4]Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway. Correspondence to: Kristine Heiney <kristine.heiney@gmail.com>, Stefano Nichele <stefano.nichele@hiof.no>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

the advancement of more robust and flexible neuromorphic computing devices.

Here, we present a local activity-dependent delay plasticity algorithm for unsupervised learning with spike times (Farner, 2022). In this learning rule, the timing of pre- and post-synaptic spikes influences the *delay* of the connection rather than its weight, causing any subsequent spike transmission between a pair of neurons to occur at a different speed. The mechanism of our method is to better align all pre-synaptic spikes causally related to a post-synaptic spike, with the purpose of producing a faster and stronger response in the post-synaptic neuron. We applied our developed delay learning method to the classification of handwritten digits (LeCun & Cortes, 2005) in a simple proof-of-concept and demonstrated that training delays in a feedforward SNN is an effective method for information processing and classification. Our networks consistently outperformed their untrained counterparts and were able to generalize their training to a digit class unseen during training.

2. Delay learning in spiking neural networks

This section presents the activity-dependent delay plasticity method developed in this study and the encoding and decoding approaches of latency coding (LC) and polychronous group pattern (PGP) clustering used in our delay learning framework[1]. The goal of our proposed learning method is to consolidate the network activity associated with similar inputs that constitute a distinct input class, so that the network will produce similar patterns of activity to be read out. With this aim in mind, the delays of pre-synaptic neurons that together produce activity in a post-synaptic neuron are adjusted to better align the arrival of their spikes at the post-synaptic neuron. Our framework was developed using Izhikevich regular spiking (RS) neurons.

Analogous to how STDP potentiates connections between causally related neurons to enhance the post-synaptic response, our delay plasticity mechanism increases the post-synaptic response by better aligning causally related pre-synaptic spikes. This alignment process is illustrated in Fig. 1(a) for the case of four pre-synaptic neurons connected to one post-synaptic neuron. As shown in this figure, the pre-synaptic spikes (purple lines) that arrive (green lines) before the post-synaptic spike (blue line) are pushed towards their average arrival time (yellow line). The delay $d_{i,j}$ between pre-synaptic neuron i and post-synaptic neuron j changes according to the following equation:

$$\Delta d_{i,j} = -3\tanh\left(\frac{t_i + d_{i,j} - \bar{t}_{\text{pre}}}{3}\right), \qquad 0 \leq \Delta t_{\text{lag}} < 10\text{ ms}, \tag{1}$$

[1]Code available at https://github.com/DelayLearninginSNN/DelayLearninginSNN.

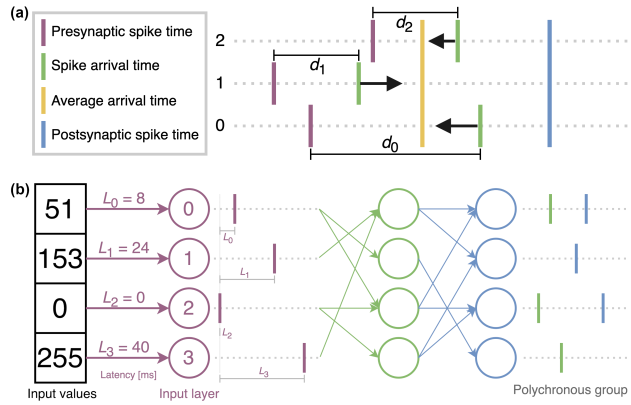

Figure 1. Schematics of algorithms for delay learning and encoding/decoding. (a) Delay learning algorithm. Purple vertical lines indicate presynaptic spike initiation times, green lines indicate presynaptic spike arrival times according to their delays d_i, and the blue line indicates the post-synaptic spike time. The learning mechanism works by pushing pre-synaptic spikes that arrive before the post-synaptic spike towards their average arrival time, indicated by the yellow line. (b) Encoding/decoding. Left: Input values are encoded as spike latencies. Right: PGPs are defined as sets of sequential activity triggered by inputs, and they are clustered in a hierarchical manner by checking the ratio of matching spikes with other PGPs.

where t_i is the spike time of neuron i, $\bar{t}_{\text{pre}}$ is the average pre-synaptic arrival time across all neurons with spikes arriving within 10 ms before the post-synaptic spike, and $\Delta t_{\text{lag}} = t_j - t_i + d_{i,j}$ is the time lag between when the pre-synaptic spike arrives at the post-synaptic neuron and when the post-synaptic neuron fires. The time window of 10 ms was selected because this is the window in which a pre-synaptic spike elicits a post-synaptic response.

The encoding and decoding approaches are illustrated in Fig. 1(b). In LC, inputs are encoded in the relative spike timing of the input neurons. That is, input channels with a value of 0 will fire first, followed by other channels in order of increasing input value. Through experimentation, we determined that rescaling the dynamic range to relative latencies of $[0, 40\text{ ms}]$ produced good results.

Our decoding approach of PGP clustering is based on the concept of polychronization, introduced by Izhikevich as the occurrence of "reproducible time-locked but not synchronous firing patterns" (Izhikevich, 2006). Thus, a PGP is one such time-locked pattern in the output layers of a network, consistently produced in response to the same input. Because different inputs from the same class do not activate precisely the same input neurons, we also introduced an unsupervised method of clustering PGPs into output classes. A given PGP is described as a nested set in our framework. Each neuron that fires in a given layer is appended to the

Table 1. Network architecture and experimental parameters

Layer size	Number of layers	Connection probability	Weight	Digits (unseen)	Train instances	Test instances	PGP match threshold
100	3	0.1	6	0, 1, (2)	20	25	80%, 90%

corresponding level of the set. Presynaptic spikes from an earlier layer are connected to postsynaptic spikes they participate in eliciting. The resulting PGP describes how activity flows through the network, retaining the ordinal relationships of the spikes without explicitly including time.

We cluster these output PGPs using hierarchical clustering. First, a pair of PGPs is said to belong to the same cluster if the number of intersecting elements in the set is greater than 95% of the mean number of spikes in the two PGPs. Each cluster obtained by this pairwise comparison is then described by the pattern averaged over all PGPs in the cluster. This is repeated dropping the matching threshold by 5% until the target threshold θ is reached. In our proof-of-concept, we used $\theta = 80\%$ and 90%.

3. Proof-of-concept: Classification of handwritten digits

To demonstrate the utility of our proposed delay learning method, we applied it to the classification of handwritten digits (LeCun & Cortes, 2005). This dataset consists of images of 28×28 pixels; we scaled these images down to a size of 10×10 and assigned an input neuron to each pixel. The details of our experimental setup are given in Table 1. We used feedforward networks with three layers, including the input layer, and fixed homogeneous connection weights.

In each iteration of the experiment, a feedforward network was generated with connectivity between layers according to the connection probability, and each connection was assigned an initial delay randomly drawn from the set of integers between 0 and 40 ms. We then provided inputs from the selected digit classes to this untrained network with local plasticity switched off to give a performance baseline for random delays. In the training phase, different inputs of the same digit classes were fed into the network with local delay plasticity switched on. Following training, we again switched off local plasticity and provided the same set of inputs as given in the baseline test phase to assess the performance of the trained network. One digit class was selected as an "unseen" class, i.e., a class presented during testing but not training, to evaluate the network's ability to generalize.

Fig. 2 shows the accuracy before and after training, calculated as the ratio of the count of the most common PGP class to the total presented inputs. In nearly all cases where the network could separate the digit classes, the trained network performed better than the corresponding untrained network; however, some networks were unable to separate the classes (2.4% and 45% of networks for PGP thresholds $\theta = 90\%$ and 80%, respectively; see Fig. 2(a)). Networks were also able to generalize their learning to a digit class unseen during training (Fig. 2(b)). Here, the accuracy remained low for the more stringent $\theta = 90\%$ but reached up to 64% for $\theta = 80\%$ (mean accuracy 32% in 38 networks able to separate the unseen class). Flexibility with the PGP threshold can thus allow networks to generalize its training to unseen classes while maintaining good performance on trained classes.

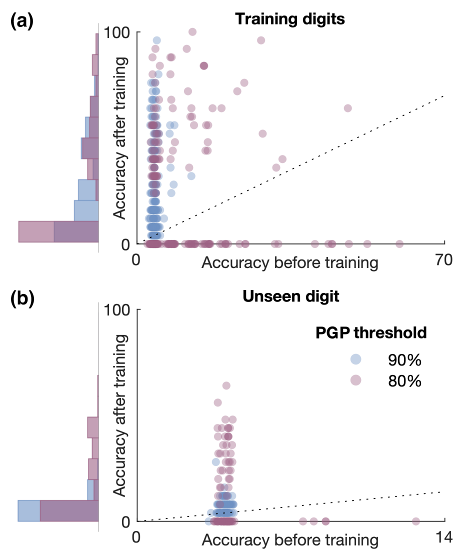

Figure 2. Accuracy of classifying handwritten digits before and after training using delay learning. (a) Two training digit classes (0,1), $N = 500$ networks. (b) One unseen digit class (2), $N = 100$ networks. Results are plotted with jitter for the sake of visualization. Histograms show the accuracy distribution after training. Accuracy of 0 indicates non-separable classes.

Examples of the activity in the output layers before and after training are shown in Fig. 3, with correct trials colored different shades of blue according to trial number and incorrect trials colored orange (for a PGP matching threshold of $\theta = 80\%$). The neurons are ordered according to their mean spike time across all trials with inputs in digit class 0; note that the top and bottom rows have different neuron order. These raster plots demonstrate the way the delay learning pushes the network to produce recognizably similar patterns

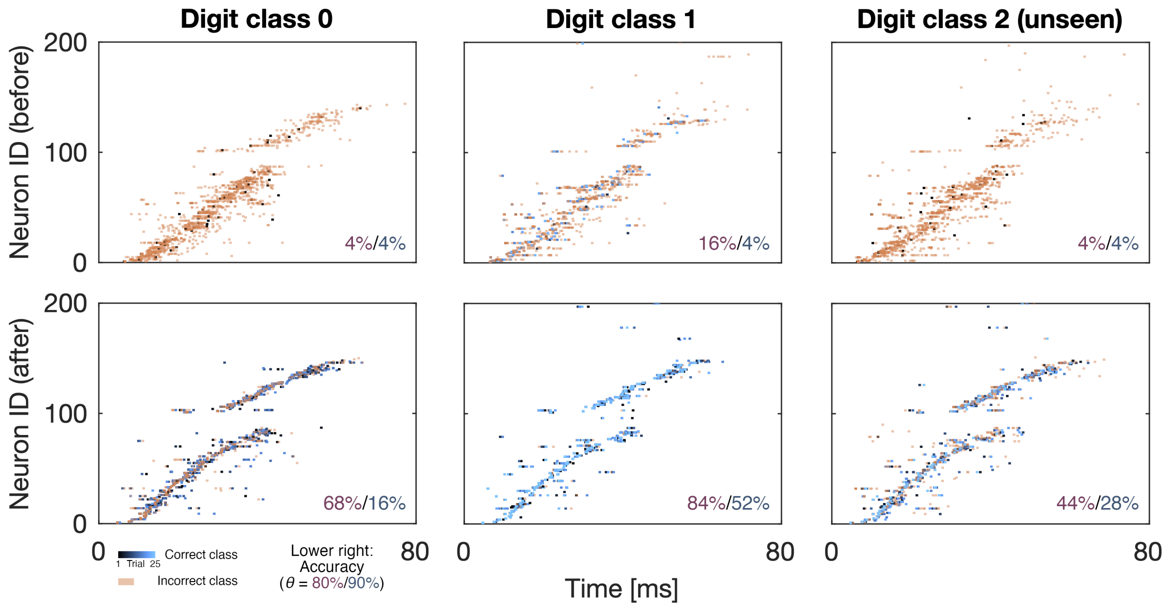

Figure 3. Raster plots of activity in layers 2 and 3 (neurons 1–100 and 101–200, respectively) before and after training for an example network. Digit classes 0 and 1 were used for training, and 2 is an unseen third class presented only during testing. Neurons are sorted according to the mean spike time for all trials in digit class 0. Colors represent whether the class was correctly or incorrectly identified, for a PGP matching threshold of $\theta = 80\%$, with the blue color scale for the correct label showing different trials. Accuracies at PGP thresholds of 80% and 90% are reported in the lower right corner of each plot.

(PGPs) when presented with inputs from the same class, as evidenced by the greater overlap of activity patterns after training. Prior to training, the network activity is less structured overall and sparser in the final layer (neurons 101–200), whereas after training, the final layer is more active, and consistent spiking patterns can be observed across many inputs from the same class. In particular, inputs in digit class 1 produce very similar patterns, with very few spikes deviating from the main pattern.

Although our method yields networks with fair classification performance, the spiking patterns we show here indicate one drawback that will be addressed in future work: the representations of each digit class are very similar, which can cause erroneous class assignment. This may be solved by introducing competition in the network to encourage more diverse representations and thus greater separability among output patterns.

4. Discussion

Neural networks with carefully designed spike time delays can support many time-locked patterns of activity, expanding the coding capacity when compared with traditional rate models (Izhikevich, 2006). Delay learning enables such polychronization in populations of spiking neurons, and our results show that we can take advantage of this richness of activity to train networks that can generalize their training to new inputs. Our results demonstrate that feed-forward SNNs trained with our proposed local delay plasticity rule produce similar activity patterns in their output layers that can be well classified in some networks with a strict PGP matching threshold of 90%. Furthermore, lowering the threshold to 80% yielded some networks able to generalize their training to novel inputs unseen during the training period.

Our proof-of-concept shows the great potential for this local delay learning method; even with only a short training period of 20 digit presentations, PGPs emerge in the network activity that allow for improved classification accuracy. As shown in Fig. 2(a), the majority of the 500 networks showed improved accuracy after training, particularly in the case of $\theta = 90\%$.

What this increased accuracy entails is illustrated in the example raster plots in Fig. 3. Our delay learning method encourages reproducible time-locked sequences of activity to propagate through the network, which leads to earlier and stronger activation in the final layer and more consistent spike timing across trials. This has two effects that improve the classification accuracy. First, the reproducibility means that the output PGPs match each other more closely, making it easier for our clustering algorithm to identify similar patterns. Second, the enhanced activation of the final layer supports a richer repertoire of activity, meaning a greater number of representations can be supported.

This second point is what endows the network with the ability to generalize. Crucially, our output decoding method is not constrained in the number of clusters it yields. This means that the network is not confined to identifying only the input classes on which it is trained. With our local delay plasticity algorithm encouraging stronger responses in later layers, we hypothesize that the trained network is able to support a range of activity patterns that extends beyond those produced in response to the training inputs. The network may thus support representations of untrained input classes that our output clustering algorithm can recognize as distinct from those of the training classes.

Although these results are promising, there are some limitations to our current approach. Fig. 2 shows that not all networks perform well after training. In most of these cases, the poor performance is largely due to non-separability of input classes, frequently accompanied by a fairly high accuracy prior to training (see Fig. 2(a) with threshold 80%). These networks are likely being over-trained and producing a homogeneous PGP that represents multiple input classes; this is further evidenced by the similarity of the representations evident even in the relatively high-performing network shown in Fig. 3. To counteract this and improve separability, it would be beneficial to introduce a mechanism to produce stimulus-specific competition among the neurons in the population; this would make the resultant representations of each digit class sparser and avoid the close similarity evident in Fig. 3. Such stimulus-specific competition could be introduced by, for example, lateral inhibition in early layers (**?**). This would encourage stimulus specificity among neurons in the same layer and give preference to neurons that fire earlier in response to a given input, leading to sparser and more distinct representations of the different input classes.

In future work, plastic weights and diverse neuron types can be combined with our delay learning approach to expand the computational capacity and enable mixed learning strategies. Our delay learning approach does not yield accuracies comparable with state-of-the-art weight training methods; however, training with delays in combination with conventional weight training has been shown to improve efficiency and accuracy (Zhang & Linden, 2003). As such, our future work will similarly combine weight and delay training as a means to evaluate how delay learning can improve conventional weight-based approaches, rather than act as a substitute for weight training.

We also expect that approach to delay learning will prove useful in the training of neuromorphic event-based hardware (Grimaldi et al., 2022). Although SNNs are computationally demanding to implement in conventional hardware, novel unconventional hardwares can enable a more energetically efficient implementation, and as such, compatible training algorithms will be needed for these new computational systems. Using event-based computing in this way is expected to be particularly beneficial in time-based tasks, such as forecasting, and we hope to test our delay learning method on such tasks in the future.

Acknowledgements

This work was partially funded by the SOCRATES project (Research Council of Norway, IKTPLUSS grant agreement 270961) and the DeepCA project (Research Council of Norway, Young Research Talent grant agreement 286558).

References

Bucher, D. and Goaillard, J.-M. Beyond faithful conduction: short-term dynamics, neuromodulation, and long-term regulation of spike propagation in the axon. *Progress in neurobiology*, 94(4):307–346, 2011. ISSN 0301-0082.

Debanne, D. Information processing in the axon. *Nature Reviews Neuroscience*, 5(4):304–316, 2004. ISSN 1471003X. doi: 10.1038/nrn1397.

Farner, J. J. Activity dependent delay learning in spiking neural networks. Master's thesis, Oslo Metropolitan University, 2022.

Gittis, A. H. and du Lac, S. Intrinsic and synaptic plasticity in the vestibular system. *Current Opinion in Neurobiology*, 16(4):385–390, 2006. ISSN 0959-4388. doi: 10.1016/j.conb.2006.06.012. Sensory systems.

Grimaldi, A., Gruel, A., Besnainou, C., Jérémie, J.-N., Martinet, J., and Perrinet, L. U. Precise Spiking Motifs in Neurobiological and Neuromorphic Data. *Brain Sciences*, 13(1):68, 2022. ISSN 2076-3425. doi: 10.3390/brainsci13010068.

Grossman, Y., Parnas, I., and Spira, M. E. Differential conduction block in branches of a bifurcating axon. *The Journal of physiology*, 295(1):283–305, 1979. ISSN 0022-3751.

Hatt, H. and Smith, D. O. Synaptic depression related to presynaptic axon conduction block. *The Journal of physiology*, 259(2):367–393, 1976. ISSN 0022-3751.

Hebb, D. O. *The organization of behavior: A neuropsychological theory*. Wiley, New York, June 1949. ISBN 0-8058-4300-0.

Izhikevich, E. M. Polychronization: Computation with spikes. *Neural Comput*, 18(2):245–282, 2006. ISSN 1530-888X,0899-7667. doi: 10.1162/089976606775093882.

Johnston, S. P., Prasad, G., Maguire, L. P., and McGinnity, T. M. A hybrid learning algorithm fusing stdp with ga based explicit delay learning for spiking neurons. In *2006 3rd International IEEE Conference Intelligent Systems*, pp. 632–637. IEEE, 2006. ISBN 9781424401956.

LeCun, Y. and Cortes, C. The mnist database of handwritten digits. 2005.

Lin, J.-W. and Faber, D. S. Modulation of synaptic delay during synaptic plasticity. *Trends in Neurosciences*, 25(9):449–455, 2002. ISSN 01662236. doi: 10.1016/S0166-2236(02)02212-9.

Lüscher, C., Streit, J., Quadroni, R., and Lüscher, H.-R. Action potential propagation through embryonic dorsal root ganglion cells in culture. i. influence of the cell morphology on propagation properties. *Journal of Neurophysiology*, 72(2):622–633, 1994. doi: 10.1152/jn.1994.72.2.622.

Markram, H., Lübke, J., Frotscher, M., and Sakmann, B. Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, 275(5297):213–215, 1997. doi: 10.1126/science.275.5297.213.

Markram, H., Gerstner, W., and Sjöström, P. J. A history of spike-timing-dependent plasticity. *Frontiers in synaptic neuroscience*, 3:4–4, 2011. ISSN 1663-3563.

Paugam-Moisy, H., Martinez, R., and Bengio, S. Delay learning and polychronization for reservoir computing. *Neurocomputing (Amsterdam)*, 71(7):1143–1158, 2008. ISSN 0925-2312.

Schrauwen, B. and van Campenhout, J. Extending spike-prop. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, volume 1, pp. 471–475, 2004. doi: 10.1109/IJCNN.2004.1379954.

Taherkhani, A., Belatreche, A., Yuhua, L., and Maguire, L. P. Dl-resume: A delay learning-based remote supervised method for spiking neurons. *IEEE Trans Neural Netw Learn Syst*, 26(12):3137–3149, 2015. ISSN 2162-237X. doi: 10.1109/TNNLS.2015.2404938.

Taherkhani, A., Belatreche, A., Li, Y., Cosma, G., Maguire, L. P., and McGinnity, T. M. A review of learning in biologically plausible spiking neural networks. *Neural Networks*, 122:253–272, 2020. ISSN 18792782. doi: 10.1016/j.neunet.2019.09.036.

Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., and Maida, A. Deep learning in spiking neural networks. *Neural Networks*, 111:47–63, 2019. ISSN 0893-6080. doi: 10.1016/j.neunet.2018.12.002.

Wang, X., Lin, X., and Dang, X. A delay learning algorithm based on spike train kernels for spiking neurons. *Frontiers in neuroscience*, 13:252–252, 2019. ISSN 1662-4548.

Zhang, W. and Linden, D. J. The other side of the engram: experience-driven changes in neuronal intrinsic excitability. *Nature Reviews Neuroscience*, 4(11):885–900, 2003. doi: 10.1038/nrn1248.