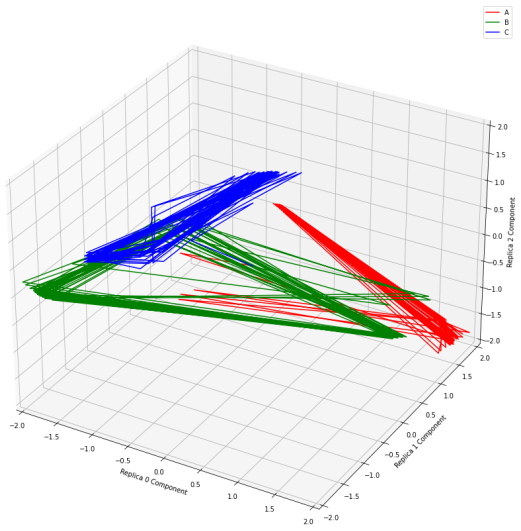
Learning Rules and Topologies for Liquid State Machines

A Survey of Performance and Representational Dynamics for Image and Speech Recognition

Ryan O'Loughlin

Advisors: *Marleen Schippers, Jelmer Borst, Thomas Tiotto, Guillaume Pourcel, Steve Abreu*

A thesis presented for the degree of
Master's of Artificial Intelligence



Faculty of Science and Engineering
Rijksuniversitet Groningen
The Netherlands
Summer 2022

Contents

1 Overview

Abstract

As prospect of the next hardware revolution looms, attending to unconventional computing paradigms becomes more important than ever. Recurrent spiking neural networks (RSNNs) are highly suitable for neuromorphic hardware and come ready with an actionable reservoir computing scheme known as the Liquid State Machine (LSM). An LSM employs an RSNN "reservoir" to process input such that meaningful features can be learned with a simple linear readout map trained on reservoir states. LSMs have garnered success in a number of domains, such as speech, image, and even video recognition. However, the hyper-parameter space for reservoir design is prohibitively large and all of these above mentioned successes make use of specialized LSMs, often with significant design-and-tuning overhead. Therefore, a pertinent task for the advancement of LSM literature is the development of an increasingly robust playbook for good reservoir design. To this end, we here survey LSMs in their many forms, both prominent and novel, with particular attention to learning rules and topologies. In terms of learning rules, Short Term Synaptic Plasticity (STSP), Spike Timing Dependent Plasticity (STDP), and their combination Long Short-Term Plasticity (LSTP) are investigated, along with non-learning simple Leaky-Integrate-and-Fire (LIF) neurons. For topologies, random, geometric, and small-world are tested. We characterize reservoir quality as a balance between diverse computational efficacy on two input modalities (image and speech) and coherent dynamics in representational state space. In doing so, we aim to distill favorable reservoir attributes and therefore better equip the state-of-the-art for LSMs that may exhibit the sort of versatility we would hope for in our artificial systems. Our findings indicate that the STSP learning rule paired with small-world topology is the most robust in terms of performance and favorable dynamics across input modalities and given other hyperparameter settings. We also observe a tentative relationship of high classification performance with certain representational dynamics in Principal-Component space. A correlation is found in full-dimensional space with the proposed representation-distance measure of *coherent separation* and performance (around a certain threshold). Therefore in total we conclude that reservoirs with STSP and small-world topologies merit further investigation and moreover that representational dynamics may be a viable option for reservoir quality assessment. Finally, these findings set the stage for proposed future research on adaptive reservoirs, synthetic learning rules, and semantically continuous representational spaces.

Summary

The experimental design of the here presented research is straightforward. A variety of liquid state machine configurations are explored by testing their performance on two simple classification tasks–image and speech recognition. LSMs are varied by hyperparameters effecting their input, reservoirs, and output layers. Special attention is given to reservoir variations on the basis of learning

rules and topologies. In the main parameter sweep, four learning rules are tested: leaky-integrate and fire (LIF), short-term synaptic plasticity (STSP), spike timing-dependent plasticity (STDP), and the novel long short-term plasticity (LSTP). LIF neurons technically do not learn, but we will still group this setting in under learning rules. Three topologies are tested: random, geometric, and small-world. Additionally, different balances of input and reservoir sparsity are tested, as well as toggling inhibitory synaptic behaviour (we call this the EI setting). Lastly, both re-initialized and continuous input feed for each sample are included in the sweep. All combinations of the above parameters are tested for classification performance in each input domain.

While procedurally simple methods are employed, empirical gain is multi-dimensional. As a function of classification performance, we observe the role of many LSM design considerations in terms of two input modalities and therefore many conclusions can be drawn about the relationship of different LSM information processing mechanisms to different input types. Not only are we here accomplishing the engineering results of classification performance, but moreover gaining insight on the fundamental properties of reservoirs themselves. To this end, we extend our reservoir analysis by applying Principal Component Anaylsis (PCA) to better characterize reservoir dynamics in terms of input type and performance. We also consider relative euclidean distances (and therefore representational organization) in full-dimensional space.

Our findings show that one learning rule in particular, Short-Term Synaptic Plasticity (STSP), is most favorable in all domains, but with the caveat that it's perfromance comes with higher variance than STDP and LSTP. Topological reservoir design is marginally in favor of small-world topology. Furthermore, we find the novel learning rule LSTP to be competitive with STDP, which typically stands as the strongest learning rule in LSM literature [17]. However, we are careful to contextualize these results in the understanding that computing with dynamical systems is highly sensitive to small changes in experimental design and therefore draw no further conclusions outside our specific classification task arenas and experimental conditions.

In terms of dynamics in PCA space, we find that high-performing reservoirs often have a coherent organization of input representation whereby intra-class input patterns are more closely grouped in PCA space than inter-class patterns. This suggests that even when reduced to the first three dimensions of maximum variance, linear classification is possible. We lend the term *coherent separation* to moments when input representations are organized well with respect to classes defined by the classification task.

To observe dynamical behaviour across time, average class PC positions are traced across the duration of each trial and several archetypes of representation behaviours are found: psuedo-noise, perfect cyclical, evolving cyclical, and dynamic encoding. We believe attending to representational dynamics may benefit future research in reservoir design and we here only lay a typological groundwork for doing so.

In full dimensional space, we find a relationship with a proposed mathematical definition of coherent separation (defined by relative eudlidean distances of

state-vectors) and performance.

Finally, we propose future research in deliberately constructed semantically continuous representational spaces, self-adjusting reservoirs, synthetic learning rules, and hierarchies of variable design for more complex concept embeddings.

2 Introduction

Any given state in any given system contains information. If it is a dynamical system, it will change states as a function of time such that information is processed. The universe itself occupies one of many (infinite?) possible states at any given moment, and transitions to its next state in the next moment via a set of update rules, which humanity has done its best to describe in terms of the laws of physics. This natural flow of information processing shares certain overlap with the aim of modern computers. Where a computer is fed specific input information, and performs an explicit and well-defined function on that input thereby producing an output in the desired form, the universe is processed according to the laws of physics (known and unknown) and thereby transitions from the past (input) to the future (output). The essential difference here, is that in the case of computers the entire procedure is predefined as determined by the user, whereas in the case of the universe, we have no such agency over the states and processing mechanisms that occur. To this end, modern computers have benefitted humanity greatly, but nonetheless there begs the question, *might we better harness the innate complexities of reality for more powerful and natural computation?*

In answer to this question, a viable approach is to introduce *reservoir computing* (RC), which incorporates a loosely defined reservoir of natural complexity in its otherwise explicit computational paradigm. In this way, like traditional computing, specific input results in a desired output, but the intermediate layer is a dynamical system that need not be controlled or even fully understood.

RC was invented simultaneously and separately in two distinct brands that still stand as the most common implementations today, each using its own flavor of artificial reservoirs. While *Echo State Networks* (ESNs) [2] use sigmoidal activation-based neurons akin to what is found in deep learning, *Liquid State Machines* (LSMs) [1] opt for more biologically plausible spiking neurons. In both cases, neurons are structured in a recurrent neural network, which allows for better information processing in the temporal domain [29]. While ESNs benefit from the continuous mathematics of activation-based neurons, LSMs stand to better exploit lessons from neuroscience for computational gain and often do so, for example, by implementing cortical-column-like reservoir structures [1], biologically plausible synaptic learning rules [17, 18, 22], and even discoveries in neuroscience as recent as astrocytes [21].

The advantages of biological plausibility do not stop at algorithms, and there has been a steady, growing push for *neuromorphic* hardware over the last thirty years [9], that has recently accelerated to an abundance of emerging novel hardware paradigms that are specifically tailored to expound on the natural advantages of the brain in terms of speed, energy efficiency, and performance. Many of these approaches are natively spiking [10–12] and therefore may viably host LSM computation, as has already been explored by [3, 5–7].

Thus, out of an interest for neuro-inspired algorithms that would be suitable to implement on promising new hardware paradigms, we here focus our research efforts on this latter brand of RC–The Liquid State Machine.

This deceptively simple paradigm carries surprising computational force, achieving state-of-the art performance on canonical machine learning tasks such as speech recognition [14, 16–18], image classification [18, 21], and even video event recognition and prediction [19, 20]. However, all of the above mentioned papers employ different reservoir schemes and readout mechanisms. The combinatoric explosion of possible architectures (when considering the nuances of learning rules, topologies, hyperparameters, and hierarchies) is gigantic and therefore LSMs come with significant overhead in the form of design and tuning. The question then becomes, how best to efficiently select for good LSM architectures? We here attempt to distill this inquiry to *what sorts of reservoir features (if any) result in generally effective liquid state machines?*

However, because inputs, reservoirs, and readout maps are inextricably entangled in the LSM paradigm, we must first preface the role of each before delving specifically into reservoir design and dynamics (see sections 3, 4, 5). A simple classification task is designed to observe computational efficacy of different reservoir configurations (section 7). Just two training samples of three classes are shown to the LSM before asking it to classify unseen samples from each class. This procedure is repeated for two input modalities–image, and speech–which allows us to draw more general conclusions about favorable reservoir properties. In total, 1008 experiments are run with unique combinations of meta-parameters (like how to feed in the input) and reservoir hyper-parameters (especially learning rules and topologies). A simple readout mapping (logistic regression on reservoir firing activity) is designed to achieve sufficiently generous performance such that fundamental information processing mechanisms of reservoirs should be well-represented rather than specific harmony with more elaborate readout maps. High-performing reservoirs of the most favored configurations are rerun 100 times for each learning rule to assess performance variance and temporal certainty (classification confidence over the duration of the trial).

Given the context of what sorts of reservoirs are performing well, representational dynamics (how different inputs are encoded into the firing activity of a reservoir) are analyzed using Principle Component Analysis (sections 6, 9), visualizing behaviour in 3-dimensions using the first three principle components. Considerations are made as to how these behaviors relate to performance. To extend this analytical approach to full-dimensionality, relative euclidean distances are measured to test for the correlation between the coherent separation of input representations with performance.

Finally, in section 10, tentative conclusions are drawn on the basis of these results and contextualized with current LSM literature. This preempts propositions for future works.

3 Liquid State Machines

The original LSM paper "Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations," [1] was published over 20 years ago. Since then, significant and diverse contributions have been made to the literature. Here, a conceptual overview of the LSM paradigm is presented in the form of a toy example. Then the original paper is summarized before notable achievements in the state-of-the-art are presented. Finally challenges are discussed.

3.1 A Toy Example

Consider the toy example of having three 3-dimensional geometric objects–a cube, sphere, and pyramid (see Fig. 1). The task is to classify these objects based on shape. These objects are therefore the *inputs* of the system. In order to compute features about these objects, they must be processed by some function. For this, an excitable medium–even a bucket of water–may be used as a *reservoir*. If any one of the objects is dropped into the bucket, the water is excited from its resting state. These perturbations are observable in the form of ripples on the surface of the water. It is reasonable to expect that for objects of different shapes and sizes, there would be different sorts or ripples that occur in response to them being dropped into the water. Moreover, it may also be expected that for all spheres, there may be a unique signature set of consequent ripples that are associated with the fundamental property of sphericalness and that this signature may be learned through observation. If enough cubes, spheres, and pyramids are dropped into the bucket, eventually one might learn which sorts of ripples are associated with which sorts of objects. In this way, a *readout map* is learned to compute salient input features through the dynamics of the reservoir. What is especially interesting, is that multiple features may be learned from the same sets of ripples–size, shape, mass, etc–and therefore multi-task is straightforward to obtain. In all cases, the pipeline is the same: *input, reservoir, output*. While this illustrative example could, in theory, be implemented, one might rather avoid wetware for the reservoir layer and instead implement an artificially generated dynamical system.

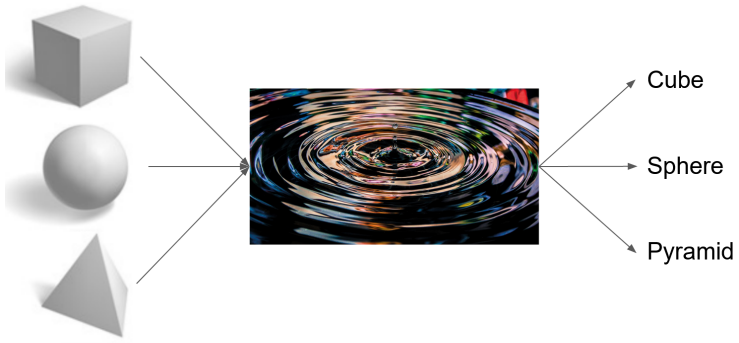


Figure 1: Toy example of an LSM.

3.2 A Paradigm Shift

What the advent of reservoir computing brought to artificial intelligence that was fundamentally new is the decoupling of an RNN from training via backpropagation. RC allows for RNNs to process information (with all of their temporal advantages) without the sometimes prohibitively complex task of backpropagating error through them [29]. The only supervised learning required for reservoir computing is on the weights connecting the RNN to a single readout layer, which circumvents the need to send error through the complex RNN. This allows for increasingly complex RNNs to be implemented without the usual training constraints. In the case of LSMs, a spiking (LIF) RNN (see Fig. 2) is employed, which would otherwise be difficult to train with backpropagation. The LSM pipeline adopts poisson spike trains as input which is processed by an RSNN reservoir and read out by a linear mapping, spiking or otherwise.

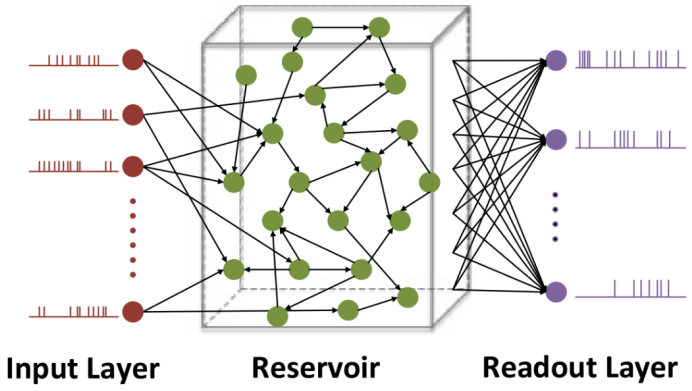


Figure 2: Intructive graphic of LSM architecture, reprinted with permission from [8].

Because RNNs have recurrent connections, information induced by input interacts with information of recently seen input that is still resonating in the activity of the RNN and therefore these RNNs are especially powerful in the time domain. RC allows for the straightforward exploitation of this advantage without the detriment of recurrent backpropagation. A consequence is that LSMs are universal real-time computers for analog functions of continuous time [1]. This statement implies that under idealized conditions, an LSM can approximate any function on continuous time series data. This is in contrast to Turing Machines that are universal approximators for discrete computations that must be offline, linear, and synchronized. An LSM has no requirement for linearity or synchronization (specific ordering of processing mechanisms) because the dynamical system used for the reservoir layer need not be stable or understood. In fact, it is the non-linear complexity of the reservoir itself that allows for a simple linear readout map to compute difficult functions on the input. This is because information is essentially expanded in the high-dimensional (and more complex) space of the reservoir and therefore it is easy to distinguish subtle differences of input. The greater the input differences are expanded by the reservoir, the greater the *separation property*. The better these differences can be learned with a readout map, the better the *approximation property*. These are the two canonical reservoir quality assessment metrics posed in the original paper [1], which will be discussed in greater detail in section 6, along with generally what is meant by representational spaces.

3.3 The State of The Art

While the original paper already managed to tackle a smorgasbord of abstract spatio-temporal classification and prediction tasks, and even included success in multi-task learning (multiple readout maps learned on same liquid states) and multi-reservoir architectures [1], the varieties of LSMs have nonetheless grown steadily in the interim of 20 years. We will here only sample from this large body of literature papers that exemplify unique additions to the original LSM paradigm that our relevant to our here presented research. Therefore our introduction should be sparse but still representative of the state-of-the-art and supportive of this work. Contributions will be grouped by the three main LSM domains: input, reservoir, and readout.

3.3.1 Inputs

Because LSMs can natively process spatio-temporal input, myriad domains have been explored. Naturally, sound and vision are go-to real world tasks. As aformentioned, LSMs have managed high-performance on speech recognition [14, 16–18], image recognition [18, 21], and even video event recognition and prediction [19, 20]. What is here important to realize is that arbitrary spatio-temporal information can be encoded into poisson spike trains which essentially emulate firing activity of a neuron population that is presynaptically connected to the reservoir. These input spike trains then evoke unique reservoir firing

activity.

The conversion of image data to spiking input simply associates an input neuron with every pixel and defines its spiking rate according to that pixel intensity value via a poission distribution given by

$$Pois(\lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{1}$$

where λ is the rate and k is the number of occurrences. Simply put, the higher the number λ, the more spiking events will occur over a period of time. This is the method employed by [18, 21] for LSM input, and by [32] for general spiking image input.

Speech recognition, on the other hand, innately occupies the temporal domain, and therefore it need not be spreadout over time via poisson rates, but rather can associate spikes with loudness thresholds at every moment in time (temporal) to a spectrum of neurons associated with different frequency ranges (spatial). This is the same way the brain encodes sounds into neuronal firing [33], and both [17,18] use a preprocessing filter [34] to convert regular audio to this spiking format.

While the the temporal domain is native to speech, and artificially added to images, video recognition is naturally spatio-temporal and is therefore suitable to spike encoding, which can be accomplished in several ways. For video event recognition in [20] a pre-trained ResNet-50 neural network is used to extract features at each frame, which are then reduced to 100-dimensions using PCA before finally being fed as spiking input to the LSM. For video event prediction, [19] uses a real-time dynamic vision sensor (DVS) [35], which is itself designed to record visual data in terms of spatio-temporal events that are equivalent to spikes.

3.3.2 Reservoirs

A liquid state reservoir is no more than a RSNN and is therefore defined by is topological structure, its information processing rules, and a number of other minor hyperparameters like thresholds, baselines, refractory and delay periods, etc. Structure (section 4) defines the architecture of synaptic connections between neurons in the reservoir network and processing rules (section 5) refers to how information is transmitted around the network and whether or not the network itself changes in response to these transmissions.

At their inception, LSMs were implemented with a 15x3x3 cortical column geometric topology [1] with dynamic synapses [44] that included a 4:1 ratio of excitatory and inhibitory neurons that had their own respective sub-hyperparameters (see sections 5 and 7 for details). Excitation means that a firing neuron will encourage the firing of other neurons it feeds into, and inhibition means the opposite. This original proposed geometric structure is often repeated in literature either precisely or nearly so, [14–17, 19], along with the inhibitory/excitatory behavior. However, alternative structures such as random [25] and small-world [23] have been explored, both of which may improve

performance with careful optimization for a given task. Processing rules, on the other hand, have been the subject of prolific research, especially (section 5.3) [13, 17, 20, 24], and less-so STSP (section 5.2) [22]. The prior implements long-term learning through synaptic weight change and the latter short-term attention through ephemeral changes in synaptic variables.

Honing in on the perfect reservoir for a given task has been approached with evolutionary algorithms [25, 27] and through Separation Driven Synaptic Modification (SDSM) [26] where weight changes are made based on separability of state vectors as related to distances of input vectors. While these methods produce improved performance, they are by no means generic and again come with overhead to tune for any given task. This characterizes an overall trend in LSM literature on reservoir refinement–that improvements can always be made for any given task in many clever ways, but these methods of improvements do not necessarily organize themselves into a generalizable playbook for reservoir design. Some resounding attention on certain methods may indicate robustness, such as STDP and cortical-based topologies.

3.3.3 Readout Maps

While the ability to learn a linear mapping from a non-linear dynamical system is a key revelation of reservoir computing, this in no way limits the variety readouts of that can be utilized. Essentially any form of supervised learning can be applied at the readout layer.

In [1], parallel perceptrons are used in the form of the *parallel-delta* rule [46], whereby universal continuous functions in the range $[0, 1]$ can be approximated by the fraction of active perceptrons in response to incoming reservoir liquid states, with learning through gradient descent. However, it has since been found that straightforward logistic regression on single liquid states extracted at regular intervals can already improve performance [14]. Similarly, linear regression can be employed for the approximation of continuous functions, such as for the object trajectory prediction of [19]. Single or multilayer activation-based neural networks can as well be used for learning with backpropagation. This is the technique used in [20] for video event classification, where liquid states from multiple sequentially connected reservoirs are first filtered through an attention learning mechanism for a single layer neural network.

In all the aforementioned cases, external mechanisms are exploited for learning, whereas [16] uses a secondary population of spiking neurons with a simple Hebbian learning rule for winner-take-all classification on speech recognition (one neuron per class, with whatever neuron spikes most as the winner), which is also used in [17]. By using spiking neurons for the readout layer, advantages of neuromorphic hardware can be gained by implementing the entire system in spikes [8, 17].

3.3.4 Other Considerations

Aside from vanilla implementations of LSMs, more specialized and complex architectures may be implemented. It has been shown that by stringing together reservoirs in hierarchical series, temporal characteristics at different time scales may be captured [20]. Ensembles of many small reservoirs (as little as 4 neurons each) have been effectively implemented by [18].

3.4 Challenges

Despite promising performance on a number of machine learning tasks, the issue remains that for any given task, and given the specific circumstances in which that task is entrenched, there exists no generally optimal reservoir, as is dictated by the "no free lunch principle" [28, 29]. However, this is not to say that some reservoirs are not generally better than others for common machine learning, especially given that specialized reservoirs may employ learning rules that allow for favorable adaptation to new tasks.

4 Structure - *Network Topology*

The connective architecture of a network defines its topology. There are a number of common methods for generating certain types of connectivity, and within each of these methods there exist a high number of unique possible networks on the order of $2^{n(n-1)/2}$ [36]. Reservoir computing already performs well with random connectivity [2], but there is still much research to be done in terms of optimizing reservoir dynamics as a function of topology, taking information from biology and network theory.

4.1 Random Topology

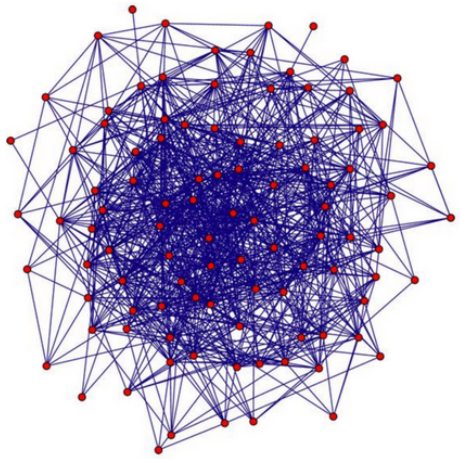


Figure 3: Example of a random graph topology, reprinted with permission from [42]

Random graphs (Fig. 3) essentially sample from all possible graphs with a uniform distribution defined by $p^m(1-p)^{N-m}$ where $N = \binom{n}{2}$. For Gilbert random graphs [37], any two nodes (or neurons) n_i and n_j have some probability p of being connected, including self-connection.

for i in range(neurons) **do**
for j in range(neurons) **do**
$P(p)\cdot$ connect(n_i,n_j)

This p value may therefore determine the internal density (or sparsity) of a reservoir simply by virtue of it defining the frequency with which connections are made. An Erdős-Rényi random graph has a similar effect, but rather constrains the total connections to the defined density [40], rather than allowing for some variance around this value.

In terms of using random graphs to determine synaptic connectivity for reservoirs, it should be noted this approach results in no particular adherence to spatial aspects of input. This is to say that spatial-temporal properties simply become temporal properties in a random reservoir. Adjacency of features in input channels are dispersed randomly across the graph and do not remain grouped in their original order.

4.2 Geometric Topology

Unlike random graphs, geometric topologies maintain a certain amount of spatio-topic coherence to the input. This is accomplished by first generating a geometric structure of nodes in Euclidean space, such as a cube (see Fig. 4) or the canonical $15 \times 3 \times 3$ neo-cortical column [1, 15], and creating edges between nodes on the basis of an exponential decay as a function of distance, such that the probability of any two neurons being connected is defined by

$$P(a, b) = C \cdot \exp^{-(\frac{D(a,b)}{\lambda})^2} \tag{2}$$

where C is a connectivity coefficient, $D(a, b)$ the Euclidean distance between a and b, and λ the connectivity parameter defining the typical distance and frequency of connected neurons. Such is the topological approach in LSM's founding paper [1].

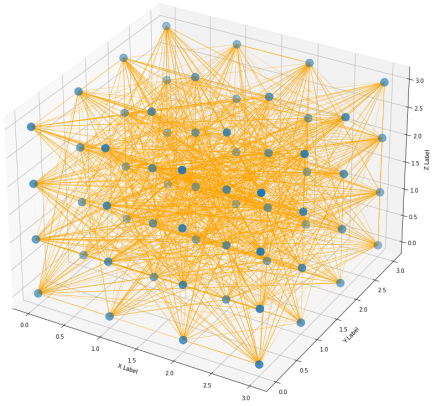


Figure 4: Example of 4x4x4 geometric topology

Geometric topologies are by far the most ubiquitous in LSM literature [1, 8, 14–17, 26]. Ostensibly, this is because of the biological plausibility of this structure in the cortical column [1]. Note, however, this spatio-topic method typically constrains information flow locally in terms of neuron location, although this may be moderated with the λ parameter. For certain types of information processing, it is imaginable that one would want some ratio of information (such

as spiking events) to reach across the network at length, thereby distributing essential features across a greater network area.

4.3 Small-World Topology

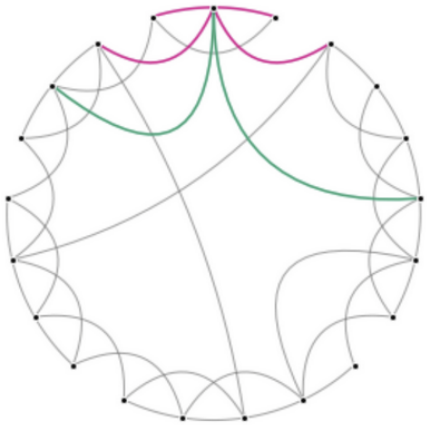


Figure 5: Example of ring lattice connectivity for small world topology of $\beta = 0.25$

Whereas random graphs implement no spatial consideration and geometric graphs bias toward local spatial properties, *small-world topologies* (Fig. 5), here considered in the flavor of the revised Watts-Strogatz model [38], provide a network generation method by which the prevalence of local and long distance connections may be moderated with the parameter β via

$$p_s = \frac{k}{k + \beta(N - 1 - k)} \tag{3}$$

$$p_l = \beta \cdot p_s \tag{4}$$

for a ring-lattice graph. When $\beta = 0$, k is the number of neighbors a node is connected to symmetrically on either side. As β increases, local connections defined by p_s untether to make random long distance connections defined by p_l, until eventually, at $\beta = 1$ an Erdős-Rényi random graph emerges very much akin to the aforementioned Gilbert random graph.

It is important to here realize that between the locally connected graph at $\beta = 0$ and the random graph at $\beta = 1$, there exists a spectrum of connectivity ranges, which may result in *scale-free* network properties where statistical outcomes at local scales may be as well observable at global scales [39]. This is because the geometry of local firing patterns may resonate at greater scales, due to a certain percentage of long range-connections. It may be expected that in some cases, this would be advantageous for the representational abilities of an

LSM reservoir and has been investigated by [23]. However, significant analysis on this subject remains to be done.

5 Rules - *Information Processing*

The nature by which information may propagate through a reservoir structure (or any structure for that matter) must be defined by some set of rules. Typically, spiking neural networks run on *leaky integrate and fire* (LIF) neurons, and sometimes incorporate plasticity rules that may evoke temporary or permanent change in the synaptic connections or neurons themselves.

5.1 Leaky Integrate and Fire Neurons

At any given moment, a neuron may have some membrane potential $v(t)$ which is constantly *leaking* and *integrating* according to

$$\frac{dv}{dt} = \frac{-v}{\tau_{mem}} + \sum_i \delta(t - t_i) \tag{5}$$

$$v \leftarrow w_{ji} \cdot s(n_j) \tag{6}$$

where τ is the membrane time constant that defines the rate of decrease for the membrane potential v, n_j is the j^{th} neuron in a population (i is the i^{th}), and the Dirac term in equation 5 simply describes a presynaptic spiking event that will increase the value of v by an approximately discrete jump in proportion to the weighting coefficient of that synapse (equation 6), as described in the above update rule (see also Fig. 6). All incoming spikes are *integrated* into the membrane potential. If spikes coming from upstream increase the membrane potential of a neuron faster than it is leaking, and if these spikes manage to instantiate an increase up to some threshold, the post-synaptic neuron *fires*, sending spiking events moderated by unique synaptic weights to all of its own downstream neurons.

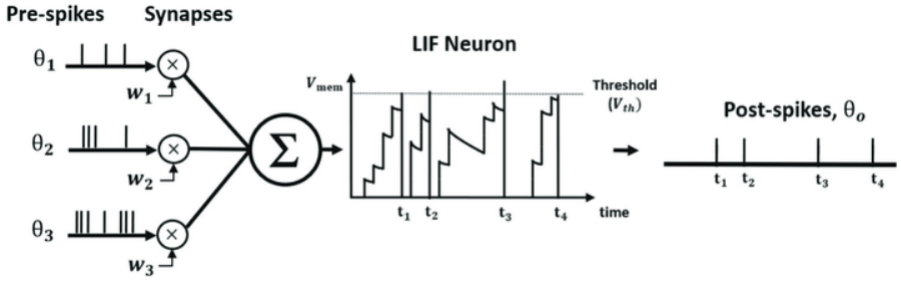


Figure 6: Leaky-Integrate and Fire neurons, reprinted with permission from [52]

This "rule" of information processing is already sufficient (in synchrony with a complex structure) for complex behavior with rich representational abilities. Nonetheless, in LSM literature, LIF neurons are often only the building blocks for more complicated neural coordination through what are commonly known as *learning rules.*

5.2 Short Term Change – STSP

The above mentioned neural parameters may be adapted over time to the benefit of unsupervised *learning*. By learning, it is meant that the network is changing in response to input, which may better equip itself to more meaningfully process unseen input in the future. Network change is accomplished by changes in synaptic weights or parameters, which can alter future firing patterns, and, if the change is lasting, even change the network topology (if a weighting connection becomes zero, this connection is essentially severed). Learning via self-change can be a temporary or permanent affair. In the case of *Short-Term Synaptic Plasticity* (STSP) [43], the synaptic parameters of calcium buildup u and resource availability x define a pair of coupled differential equations as follows:

$$\frac{dx}{dt} = \frac{1-x}{\tau_D} - ux \sum_i \delta(t - t_i) \tag{7}$$

$$\frac{du}{dt} = \frac{U-u}{\tau_F} + U(1-u) \sum_i \delta(t - t_i) \tag{8}$$

$$v \leftarrow \frac{(w \cdot x \cdot u)_{ji}}{U_{ji}} s(n_j) \tag{9}$$

Where x defines the fraction of resources required to fire and u is the calcium buildup at that synapse. U is the baseline calcium value. Biologically speaking, calcium buildup facilates neural firing by increasing the influence of spikes on membrane potential [43, 44]. This mechanism constitutes a short term change in synaptic (and therefore neuron) behaviour, whereby recently active synapses are more likely to become active again as calcium increases. The calcium time constant is greater than that of the resources. Therefore resources replenish faster than calcium depletes and so there is a tendency toward facilitation. However, given enough firing in a short amount of time, resource depletion (equation 7) overrides this facilitation (equation 8). See Fig. 7 for a visualization of this relationship. The update equation 9 for the membrane of a specific neuron n_i indicates that for any pre-synaptic firing of some neuron n_j, post-synaptic membrane potential of neuron n_i is updated according to the specific synaptic parameter values between the two. Due to their own time constants, changes in these synaptic variables will always decay back to baseline if no firing occurs and these changes are therefore short-term and temporary. Importantly, $\tau_F = 150ms$ is greater than $\tau_F = 20ms$ and thus calcium buildup decays more slowly than the fraction of resources required to fire, resulting in overall facilitation.

Notably, [50] found that through these temporary synaptic update mechanisms, 'activity-silent' information can be maintained in this ephemeral network of facilitory connections for a short-period (≈1ms) of time. So great is the priming effect of calcium build-up, a network response associated to a given input may be re-elicited with a random noise stimulus. In some sense, this results in an attentional effect on presently seen input (anticipation of continuity) and therefore may be useful to reservoirs in certain task domains–especially of small feature time scales. It is also worth considering that while the original LSM pa-

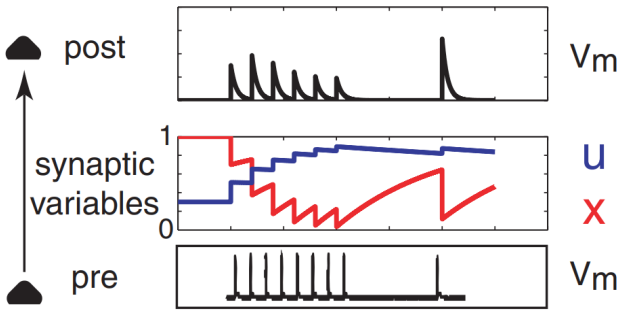


Figure 7: The relationship of required firing resources x and calcium buildup u, borrowed from [43]

per [1] uses similar synaptic parameters as STSP, they are implemented under the biologically derived [44] paradigm and are mostly depressing. All-facilitory STSP has received little to no attention from the LSM literature thus far, and may benefit from its sharp responsivity to input.

5.3 Learning Through Long Term Change – STDP

Of course for many learning tasks, it may be useful to elicit some permanent change in the network and in fact this has already been shown to be a beneficial mechanism in LSMs [**?**,**?**,17], most often taking the form of *Spike Timing Dependent Plasticity* (STDP) [45], which is best summarized by the quote "Neurons that fire together wire together" -Donald Hebb. As seen in Fig. 8, the more closely in time two neurons fire, the greater the change in their shared synaptic weight. If they fire in the correct order (pre-then-post), they are assumed to be correlated and the synaptic change is that of facilitation through an increase in their synaptic weight. Should they fire in the wrong order, the weight is decreased. These updates take the simple form of

$$\Delta w_+ = F_+(w) \cdot \exp^{\frac{\Delta t}{\tau_+}} \tag{10}$$

$$\Delta w_- = F_-(w) \cdot \exp^{\frac{\Delta t}{\tau_-}} \tag{11}$$

Where $F(\cdot)$ may be additive (the generic choice for reservoirs) or multiplicative with respect to the existing weight value. Δt is the difference in firing time and τ_+, τ_- are the respective time constants for positive and negative firing differences.

This canonical form of Hebbian learning is biologically plausible and effective in a number of spiking neural network domains–an elegant implementation of unsupervised local learning. It can even be used for a spiking readout map on reservoirs [16, 17, 20]. Note, that there are some additional choices of what neighborhood of neurons to consider for updates, such as nearest neighbor or

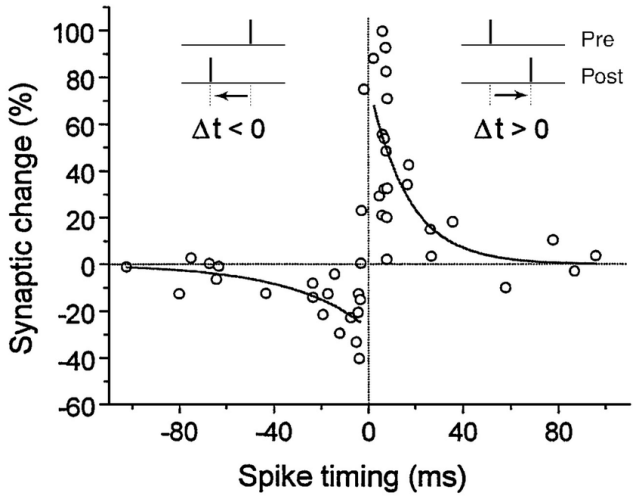


Figure 8: Explanatory diagram of typical STDP update behavior, reprinted with permission from [53].

even global (all pairing), the latter of which is employed in the present study on the basis of its higher performance for reservoirs of relevant size found in [17].

6 Representations

Implicit in liquid states that correspond to some input is a *representation* of that input. Representations can be better or worse and this is a function of how well meaningful characterstics of the input are preserved. For functional purposes, it is also essential that these meaningful characteristics be interpretable by the readout mechanisms of the system.

6.1 Separation and Approximation Properties

Returning to *separation* and *approximation*, the viability of an LSM for a classification task is entirely defined by its ability to separate input into the correct categories. To accomplish this, categories of different classes should be represented *separately* in different regions of state space. Moreover, a readout mechanism should be able to *approximate* these separations through its linear supervised learning procedure. To help illustrate this, Fig. 10 demonstrates how three classes might be organized in a 3-dimensional state space. This may be equivalent using only three neurons and their firing rates for a given time window to represent each input at a given moment in time. Clearly, if a linear classifier is to be used, it stands to benefit from different classes occupying different regions in state space. Thus, this illustration embodies well-organized representations of input in terms of this classification task. Different task-defined classes occupy different linearly separable regions in state space. Separation and approximation properties would be high here because classes are separately represented and linearly separable.

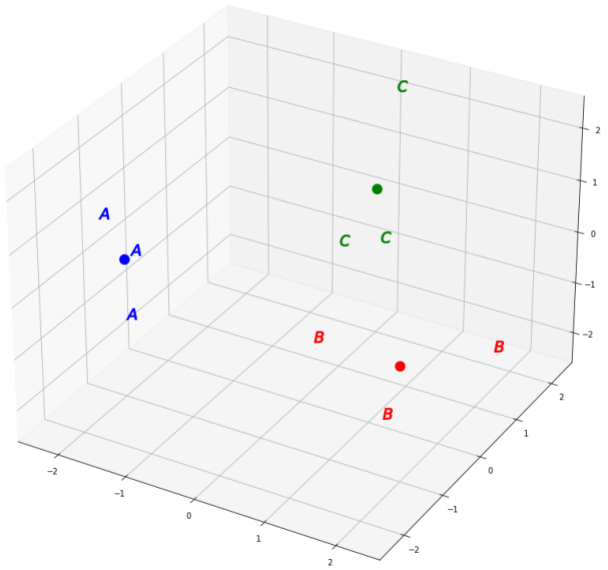


Figure 9: Instructive representation example of well organized representations of three classes. Letters are classed inputs, dots are centroids, and colors are clusters.

On the other hand, if input is highly separate, but not organized by class, as in Fig. **??**, then the *coherent* separation property is low because a linear classifier cannot be learned. Another way of phrasing this, is that intra-class clustering must be greater than inter-class clustering. Items from the same class should be closer to one another than items of another class. Thus an organized state space should emerge that is isomorphic to the organization of the input classes. This is a phenomena explored by [1, 15, 17, 18, 26].

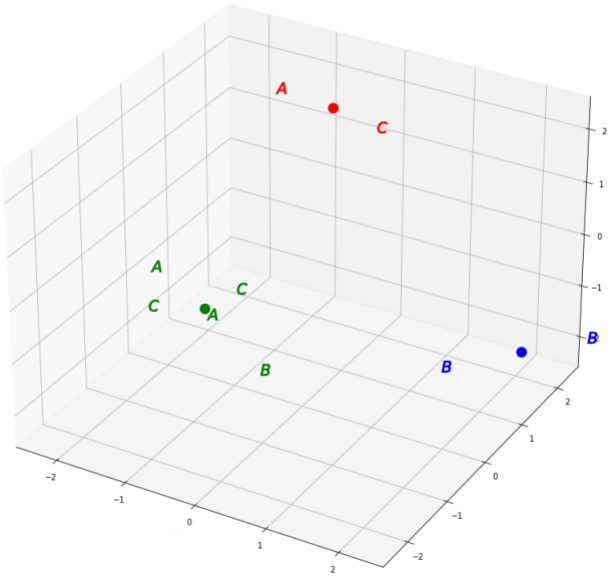


Figure 10: Instructive representation example of poorly organized representations of three classes. Letters are classed inputs, dots are centroids, and colors are clusters.

6.2 Distance Measures

While these low dimensional examples are instructive, LSMs usually use dimensionality much greater than what can be visually imagined for representation. As soon as more than three neurons are in play, visualizability is lost. However, in higher dimensions, more complex sets of features can be used to define separability [29]. To assess the quality of separation and approximation properties in higher dimensions, relative euclidean distances of representations in the full dimensionality can be considered, such as [15], where separability is defined by

$$Sep_{\Psi}(O) = \sum_{m=1}^{N} \sum_{n=1}^{N} \frac{||\mu(O_m) - \mu(O_n)||_2}{N^2} \tag{12}$$

or the refined version of [26]

$$Sep_{\Psi}(O) = \sum_{m=1}^{N} \sum_{n=1}^{N} \frac{||\mu(O_m) - \mu(O_n)||_2}{N^2 + N \sum_{m=1}^{N} \rho(O)_m} \tag{13}$$

Where $Sep_{\Psi}(O)$ is the separation of representations across input samples at a given moment in time. N is the number of classes. $\mu(O_m)$ is a average (centroid) state vector at that time for class m. So essentially for equation 12, the total separation is defined by the sum of differences between class centroids, divided by the number of classes squared. While equation 12 only makes use of

class-centroid spread (inter-class separation), equation 13 better encaspulates the clusterization of same-class representations (intra-class spread) by inversely correlating the variance $\rho(O)_m$ of positions for same-classed items in reservoir quality assessment.

6.3 PCA

Aside from blindly working in higher dimensions, dimensional reduction through principal Component Analysis (PCA) [**?**] can as well be applied to both visualized high-dimensional data and distill the most significantly defining featured by focusing only on the dimensions of maximum variance. This technique has been used to assess reservoir quality and performance in [17,18]. In both cases it has been found that reservoirs which demonstrate separability best in PCA space tend to perform better for classifications tasks such as spoken letter recognition and image recognition.

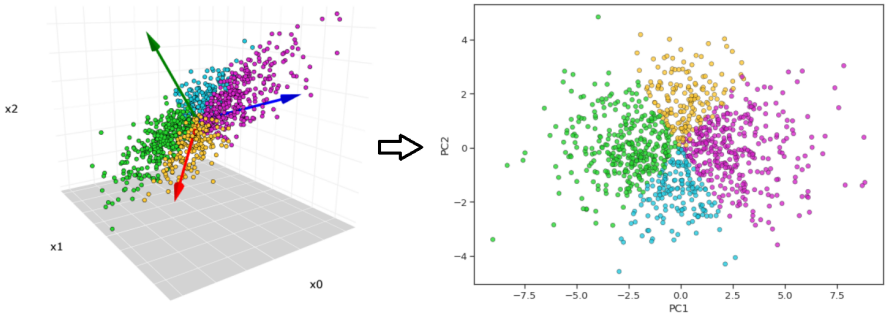


Figure 11: Example of dimensional reduction through PCA, adapted from tutorial [48].

Intuitively, PCA reduces the dimensionality of data to its directions of maximum variance. Therefore, ideally, the remaining compressed directions are not linearly dependent. In some sense, this impresses the most informative possible representation of the high-dimensional data to lower dimensions. Referring to Fig. 11 (left), it can be seen that the directions of maximum variance are defined by the red and blue arrow and therefore these define the plane onto which the date is compressively projected on to in 11 (right). This is accomplished by first centering (normalizing) the data about the origin by subtracting from all points the mean of the data. Next, the singular value decomposition SVD is computed and only the desired number of principal components are extracted from the first columns of the matrix. Apply these columns to the normalized cloud of high dimensional data$\mathbb{R}^n$ to transform it the desired dimensionality $\mathbb{R}^m$, such that the transformation

$$\mathbb{R}^n \Rightarrow \mathbb{R}^m \tag{14}$$

is achieved via the mapping

$$x \Rightarrow f_1(x), ..., f_m(x) \tag{15}$$

where $f_i(x)$ is the the corresponding column of the SVD matrix U_i acting on the the normalized data $\bar{\mathbf{x}}$.

In the case of [17], PCA is applied to an a spoken letter speech recognition task by reducing all state-vectors for a given moment in time across all input responses to 3-dimensions as visualized in Fig. 12, where $s^i(t)$ is the liquid state response vector at time t for input i and $R(t)$ is the collection of these responses across all samples. A random moment in time is selected and PCA preserving only the first three components of maximum variance across all samples is executed such that representations can be visualized in three dimensions.

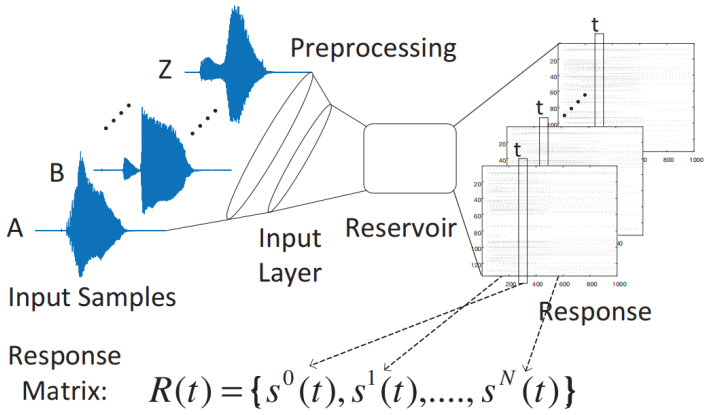


Figure 12: Example of PCA procedure on liquid state responses for speech recognition, reprinted with permission from [17].

7 Methods

We here try to focus on a few key inquiries with a strict experimental design. An informed sampling of the large space of possible liquid state reservoirs is made by testing different combinations of learning rules, topologies, and other hyperparameters on two classification tasks–image and spoken digit recognition. The quality of these many different reservoirs are assessed in two ways.

- How well do they perform on these tasks in terms of correct classification?
- How well are their representations in state space organized?

Based on these results, the correlation of representational dynamics to performance may be considered. In summary, the procedure is as follows:

1. Generate X amount of unique LSMs (varying meta and hyperparameters)
2. Test on classification tasks in both visual and audio domains
3. Assess classification performance
4. Assess representational dynamics
5. Consider how dynamics and performance are related

It should be noted that the most substantial aspect of this experimental design are the performance results, as they can straightforwardly apply to the applications of new reservoir configurations on real tasks. Dynamical behaviors and their relationship to performance is a more tentative investigation that may only yield indications of where further attention may be directed in terms of reservoir quality assessment. In a sense, performance assessment entails an engineering experiment, while dynamical considerations relate more to an undercurrent of reservoir theory, for which results would have less explicit applications.

7.1 Experimental Design

All LSM simulations are coded from scratch using the *Brian Neural Simulator* [49]. Python packages are used for logistic regression (SciPy) and for small-world node pairing. Everything else is hand-crafted and can be found in the project repository (NeuroMachines).

The visual domain is tested with the MNIST handwriting dataset [30] and the auditory with the Heidelberg spoken digit dataset [31]. For both input modalities, two training samples from three class are shown to reservoir neural populations of only 135 neurons before testing classification performance on single examples from each class. The amount of training data and reservoir population size both function as ways of moderating task difficulty. The prior is essentially and axiom of all machine learning (that performance betters with more training data) and the latter of specifically reservoir computing [17]. These knobs can be intuitively 'tuned' to produce results that spread out performance of unique reservoir configurations sufficiently to meaningfully rank

performance as a function of reservoir properties. Because simple image and speech recognition are not particularly difficult for LSMs, limited training data and medium-small reservoirs (in terms of literature standards) are used. These are tuned to the speech recognition task (the more difficult of the two, due to more complex temporal properties) and carried over to image recognition.

This approach had a secondary benefit of being more computationally cheap than using large reservoirs on many hundreds of examples. MNIST trials lasted 140ms and Heidelberg 700ms. It is reiterated here that our primary interest is in the more spatio-temporally rich spoken digit dataset, and the MNIST only serves as a point of reference.

It should also be note that two "meta-parameters" are explored. *Input feed* can either be *reset* or *continuous*, meaning that input samples can be fed to freshly re-initialized reservoir conditions each time, or all samples are fed to the reservoir in a continuous stream. *X-atory* behaviour refers to whether the network is randomly initialized with all positive weights, or if the excitatory/inhibitory parameters of [19] are adopted, which employ specified initial weighting assignments that include negative values. In this latter case, each neuron is associated with either an excitatory or inhibitory tag. Excitatory-to-excitatory connections (EE) have a synaptic weighting value of 0.5, EI 2.5, IE -2, and II -2. Transmission delay for all connections is 0.8ms, except for EE which is 1.5ms. Refractory period for spiking is 2ms for all but EE, which is 3ms. We will generally refer to this group of parameter settings as *EI parameters*. These parameters are biologically plausible [44] and allow for more symmetrical membrane potentials around zero because in many cases, the potential is brought to negative values through inhibition. This naturally results in less spiking (and therefore more sparsity), which may be more energy efficient. This may also result in more unique firing activity by preventing saturation.

This experimental design comes with a number of particular consequences. Firstly, it can be found which reservoir configurations perform the very best in terms of classification. Secondly, it can be found which key parameters (like learning rules) are most commonly found among the top performers given other parameter settings, which may indicate general robustness. Finally, it can be seen how these findings hold across both modalities of input.

7.1.1 Neuromorphic MNIST Handwriting Dataset

We test on the computer vision task of image recognition for hand-written digits. Note, this task is primarily defined by spatial features. Images are converted in accordance to the procedure found in [32]. 28x28 pixel images are unraveled into an input vector $\in \mathbb{R}^{784}$ defined by the pixel intensity value at each index. This vector determines the Poisson spiking rates across 784 input channels (see Fig. 13). However, now there is spatial structure defined by the original image incorporated into the arrangement of these spike trains. Running the simulation with these spike train inputs for some amount of time loosely simulates light-intensity spiking data being processed by the brain [32] and in this way a static image can be impressed on the reservoir over a period of time.

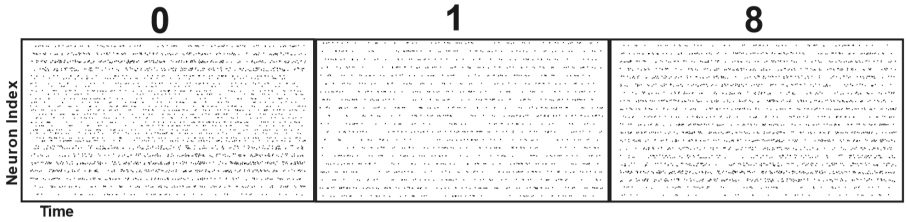


Figure 13: Raster plot for the handwritten digits 0, 1, and 8 (left-to-right). Note, these are three unlabled raster plots, where the y-axis are the 784 input channels and the x-axis is time. Dots indicate a spiking event at that channel for a given moment in time. Notice features of the digits are discernable even to the human eye. The digits 0 and 8 involve more markings when written and therefore can be seen to have more spiking events. The gap in the center of the number 0 is as well visible in its firing pattern.

7.1.2 Heidelberg Spoken Digit Dataset

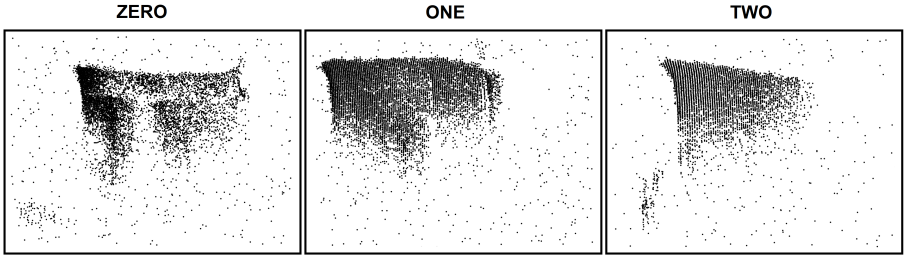


Figure 14: Cochlear implant recordings of spoken digits "zero", "one", and "two." Note, these are three unlabled raster plots, where the y-axis are the input channels (ordered in terms of frequency) and the x-axis is time. Dots indicate a spiking event at that channel for a given moment in time.

Finally, our system is tested on the Heidelberg Spoken Digit Dataset [31] where spoken digits are recorded via a cochlear implant in a spiking form similar to how the brain might receive audio information. Neuron indices correspond to increasing frequency values. If a certain decibel intensity is reached for a given frequency, the associated neuron will spike. Of course, any sound could in theory be preprocessed to adopt a spiking form, such as done in [17].

Importantly, the above mentioned input modalities are *spatio-temporal* tasks, for which RSNNs have a native advantage due to their time-embedded spiking nature [1, 19]. By providing the reservoir network with information that has structure in both the time and space domains, reservoir information processing abilities can be more thoroughly assessed. While the neuromorphic MNIST dataset is really defined by its spatial features that are stretched out over time,

the spoken digits have spatial grouping defined by frequency adjacency in the recordings, and temporal continuity defined by the duration of the different phonemes in the spoken digit. Thus we can think of the tasks in the following way:

- Spiking MNIST → Mostly spatial
- Spoken digit → Spatio-temporal

Therefore we can contextualize the results accordingly for each of these domains. Our representational dynamics will be exclusively characterized in terms of the Heidelberg dataset because it is the more spatio-temporally rich of the three.

7.2 Reservoirs

In total, 1008 different experiments are run for each dataset, varying hyper (table 2) and meta (table 3) parameter settings, given a set of static parameters (table 1). 'Neurons' refers to the number of neurons in the reservoir population τ_{mem} refers the leaking rate from equation 5. The remaining table 1 parameters are self-explanatory and moreover, detailed in the LIF subsection in section 5.

Parameter	**Value**
Neurons	135
τ_{mem}	30ms
Baseline	13.5 mV
Threshold	15 mV
Refractory	0ms
Delay	1.5ms

Table 1: Static Parameters

Table 2 includes a number of nuances, here explained. *Input density* defines the probability of connection between any input channel to any reservoir neuron. In total, there should be roughly that percentage of connections made out of all possible input-to-reservoir connections. Similarly, *reservoir density* defines the probability of any two reservoir neurons making a recurrent connection (including self-connection). It is important to here realize the the reservoir density parameter *is equivalent to the random topology* p_{random} *parameter.* Therefore, while the β and *dimensions* parameters are only varied when their respective topologies are in play (small-world and geometric) the p_{random} is varied for *all* topologies, because it also defines reservoir-density. In some sense, geometric and small world topologies can be thought of as levels of organization used to wire a random topology in a non-random way. For every three random topology experiments run, there are nine small-world and nine geometric topology experiments. For example, each β parameter is run for each of the reservoir-density

(and therefore p_{random}) parameter. Results for random topology are normalized to $\times 3$ for fair comparison.

Parameter	**Values**
Input density	0.1, 0.2, 0.3
Reservoir density	0.1, 0.2, 0.3
Learning	LIF, STDP, STSP, LSTP
Topology	random, geometric, small-world
p_{random}	0.1, 0.2, 0.3
β	0.0, 0.33, 0.66
Dimensions	$15 \times 15 \times 3$ $9 \times 5 \times 3$ $27 \times 3 \times 1$

Table 2: Hyperparameters. p_{random}, β, and dimensions are specific to their respective topologies of random, small-world and geometric. p_{random} is equivalent to reservoir density and therefore for every three random topology experiments, there are nine small world and geometric experiments.

Finally, the meta-parameters simply refer to those settings that apply to the whole LSM. EI includes excitatory and inhibitory behavior. Input feed toggles between resetting and continuous input sample feeding to the LSM.

Parameter	**Values**
EI	True, False
Input feed	reset, continuous

Table 3: Meta-parameters

7.2.1 Learning Rules

STSP is implemented with the typical time constant parameters from section 5.2. STDP is adapted in accordance to the Brian simulator guidelines [49], such that instead of explicitly calculating an exponential function on every spike-timing difference for every neuron pairing, firing traces a are used as synaptic parameters between each pair,

$$\frac{da_{pre}}{dt} = \frac{-a_{pre}}{\tau_{pre}} \tag{16}$$

$$\frac{da_{post}}{dt} = \frac{-a_{post}}{\tau_{pre}} \tag{17}$$

which are constantly decaying. $\tau_{pre} = \tau_{post} = 20ms$. However, updates are made for pre and post-synaptic firing such that

$$a_{pre} \leftarrow A_{pre} \tag{18}$$
$$a_{post} \leftarrow A_{post} \tag{19}$$

where $A_{pre} = 0.01$ and $A_{pre} = -A_{pre} * \tau_{pre}/\tau_{post} * 1.05 = -.0105$. At each firing event, the weight between that neuron and any other is updated according to

$$w \leftarrow A_{pre} \tag{20}$$
$$w \leftarrow A_{post} \tag{21}$$

where weight values are clipped between 0 and 0.01 to prevent saturation. This is a common technique, recommended by [49] and similarly employed by [16,17].

In addition to previously described learning rules, a novel rule that combines STSP *and* STDP dynamics for long and short term learning is also tested under the name Long Short-term Plasticity (LSTP). Because the dynamics of STSP and STDP do not directly interfere with each other, the above mentioned formulations for STSP and STDP are simply applied together in one all-encompassing learning rule, which may draw advantages from both short and long-term learning, though with some added update complexity.

Given the other most favorable settings, additional ivestigations are executed for each learning rule. The role of synaptic parameters (calcium buildup, firing traces, etc) are observed for different performance levels and firing behaviours. To test reproducibility, 100 reruns of the experiments are performed for each learning rule with new random initializations, therefore testing for performance variance. Moreover, temporal certainty (classification confidence over time) is tested for each of these settings.

7.3 Readout

As aforementioned, the present research is concerned with better understanding reservoir dynamics in order to characterize reservoir quality. That being said, a generous readout map is designed, such that we can assume a sufficient amount of information is being consumed from the liquid states to approach some pseudo-ideal classification performance. This is in contrast to the many clever spiking readout maps found in the literature, such as [16], that are native to fully spiking hardware and therefore more energy efficient and much faster. By instead opting for an external logistic regression model similar to [14, 19], we reduce our bias toward reservoirs that may simply be better suited for some lean readout map in particular.

In order to implement an external readout map using the classical machine learning logistic regression algorithm, spikes must first be translated into numerical data. This is achieved by one-hot encoding spikes in 1ms time windows for a given liquid state across all neurons, as shown in Fig. 15. If a neuron i spikes in some window $[t, t + 1ms]$ then its neuron index is assigned the value

of 1 at the time t window index

$$S_{[i,t]} \leftarrow 1 \tag{22}$$

$$x(t) = [x_0(t), x_1(t), ..., x_N(T)] \tag{23}$$

$$S = [x(0), x(1), ..., x(T)] \tag{24}$$

where S is the response matrix for the entire sample, x is the state vector for all neuron indices $[1, ..., N]$ at some time t, and T is the length of the sample. If multiple spikes occur for that neuron in that window, the value is simply incremented by that amount again and therefore the data is not strictly one-hot-encoded, but rather binned.

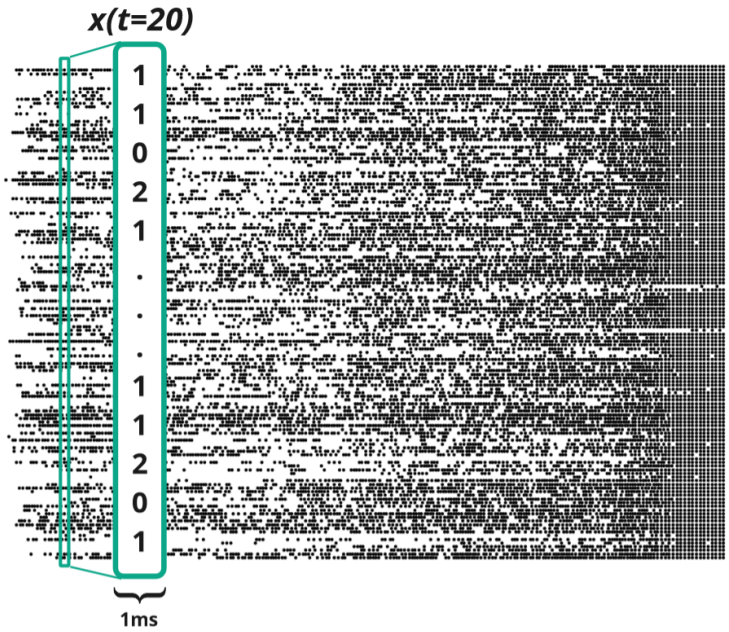


Figure 15: Example of counting spikes within a given 1ms time window (at t = 20ms) at each neuron index. Note the raster plot in the background is an example of a liquid state response to Heidelberg speech input.

Now ready with prototypical machine learning data, an even further advantage is given to the system by partitioning the data into groups of 20 windows (each 1ms) and *then* feeding them into a generic logistic regression algorithm with the appropriate labels. Once weights are learned for these "chunks," future unseen chunks may be classified appropriately. Chunk size can be used to moderated performance efficacy. The larger the chunks, the easier the classification task.

7.4 Representation Analysis

7.4.1 PCA

In order to navigate the unseeably high-dimensional representation state space, PCA is applied. This allows for the visualization of higher-dimensional spaces at a palpable and familiar 3D scale,

$$x(t) \in \mathbb{R}^{135} \rightarrow s(t) \in \mathbb{R}^3 \tag{25}$$

but it must be said that these are not necessarily lower-resolution versions of the full-dimensionality. Fundamental nuances of the data may be lost and we therefore use this technique here with tentative caution, drawing conclusions only about the relative compressed dynamics–not inherent characteristics–of the full data. We are not alone in this approach, however, and PCA for reservoir analysis can be found in [17, 18]. This approach allows for the characterization of different dynamical patterns in PCA space, which may be related to performance measurements and could give insight as to how dynamics relate to reservoir quality for our classification task of interest (The Heidelberg spoken digit dataset).

7.4.2 Separation Measures

Similar to [15,26], relative euclidean distances are used to assess separation and representational organization in full-dimensional state-space. This is accomplished by summing the distances of class-centroids and subtracting from that the average within-class distances according to

$$Sep_{org}(X) = \sum_{t=1}^{T} \sum_{c=1}^{C} \sum_{r=1}^{R_c} ||\mu(x^c(t)) - \mu(x^r(t))||_2 \tag{26}$$

Where X is the response tensor across all time T and samples. Each sample makes one of nine total one-hot-encoded response matrices S from equation 24. C is the number of classes and R is the number of replicas in each class. $x^i(t)$ refers to a response state-vector for some sample i. Thus, $Sep_{org}(X)$ describes the average inter-class distance minus intra-class distance over the entire trial. In this way, not only is separation considered, but also coherent organization (how well inputs of the same class are represented more similarly than those of other classes). We use the term coherent separation to refer to the degree to which inter-class separation supercedes intra-class separation as defined by $Sep_{org}(X)$ in equation 26. This can also refer to such instances in PC space. This method is selected over equation 12 and 13 because we found it to correlate more with performance and because it sums over the entire trial, allowing for easier comparison across experiments.

8 Classification Performance Results

As a pure machine learning task, performance on correct classification of input patterns for image and speech data are recorded and compared. Remember that the LSM paradigm used here employs aggregate classification, whereby at any given moment, the class that has been guessed the most so far is the classification at that moment. Performance is determined by the classification at the final time step for a given trial. A useful way of tracking performance over time is to measure the *certainty*, whereby the ratio of the current number of correct class-guesses to the most-guessed class is taken. When the most-guessed class *is* the correct class, this number is one.

$$C(t) = \frac{\text{Correct guesses so far at time } t}{\text{Most guessed class so far at time } t} \tag{27}$$

Ties are given favor of correctness. To compare performance across reservoirs, the final certainty across all classes is taken.

8.1 Image

Hyperparameter	Value: Occurrences
Learning	LIF: 26, STSP: 64, STDP: 46, LSTP: 46
Topology	rnd: 25, geo: 77, smw: 80
Input density	0.1: 45, 0.2: 67, 0.3: 70
Reservoir density	0.1: 69, 0.2: 57, 0.3: 56
p_{rand}	0.1: 8, 0.2: 11, 0.3: 6
Dimensions	[15, 3, 3]: 26, [9, 5, 3]: 23, [27, 5, 1]: 28
β	0.0: 22, 0.33: 26, 0.66: 32
EI	False: 114, True: 68
Input feed	reset: 61, continuous: 121

Table 4: Hyper Parameter Occurrences in Top 182 Performers for image recognition. Random topology and all values for p_{rand} should be normalized to $\times 3$ for a fair comparison to other parameters. This is because p_{rand} essentially defines reservoir density and was varied for the other two topologies as well. Note that the top 182 performers are considered here because they all achieved perfect classification *across the entire sample* (a 182-way tie for first-place).

LSMs performed extremely well on the spiking MNIST image classification task. Over 500 configurations achieve correct final classification for all three classes by the final time step, and 182 configurations achieve 100% certainty for the entire trial (see Table 4). Experimental design could have been altered to better tease apart performance, but it was decided for comparison purposes to keep parameters the same as for speech recognition, as this was the principle task of interest.

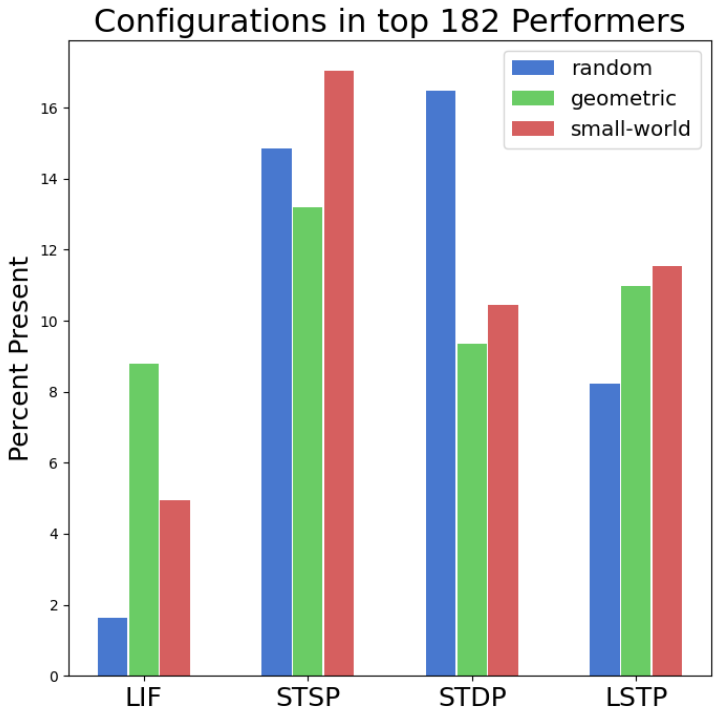


Figure 16: All configurations that achieved perfect certainty across entire trial for MNIST handwriting recognition computer vision task

Fig. 16 displays the quantity of learning rules and topologies that achieve correct classification by accross the whole trial. It can be seen that STSP is the most commonly occurring learning rule and it's most commonly combined topology is that of small-world. However, preferred topology varies per learning rule. Interestingly, LSTP, which contains the same short-term synaptic learning as STSP, also pairs best with small-world. Random topologies are noticeably preferred by STDP and rejected by LIF.

8.2 Speech

As the spatio-temporal Heidelberg spoken digit dataset is our primary classification task of interest, more in depth results are presented here. Firstly, raw performance is given, followed by synaptic dynamics, temporal certainty, firing totals, and reproducibility.

8.2.1 Performance

By design, performance is well spread out for the Heidelberg spoken-digit recognition task. Many reservoirs perform well and even achieve perfect classification

with as little as two training examples per class. Examples of liquid state responses for reservoirs that achieve perfect classification can be seen in Fig. 17. A typical high performance plot can be seen in Fig. 18, where we can see after an initial wash-out period, during which time the reservoir is getting "acquainted" with its testing sample (moving away from its random voltage and/or weight initializations), the readout quickly becomes certain which sample is which (in this case the spoken digits "zero", "one", and "two").

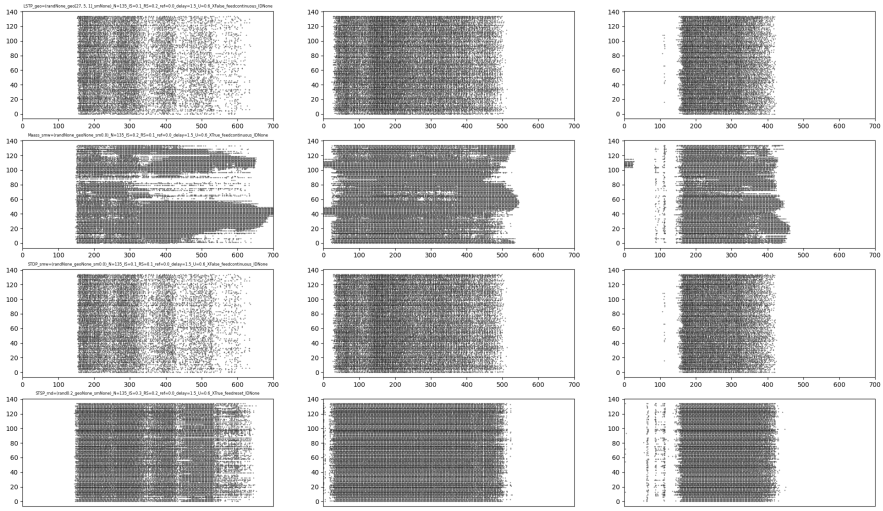


Figure 17: Four perfectly-classifying reservoir configurations and their liquid state responses for each class of the Heidelberg dataset. Columns = ["Zero", "One", "Two"] input classes, rows = [LSTP, LIF, STDP, STSP] learning rules from top to bottom. Note the LIF configuration includes inhibitory behavior.

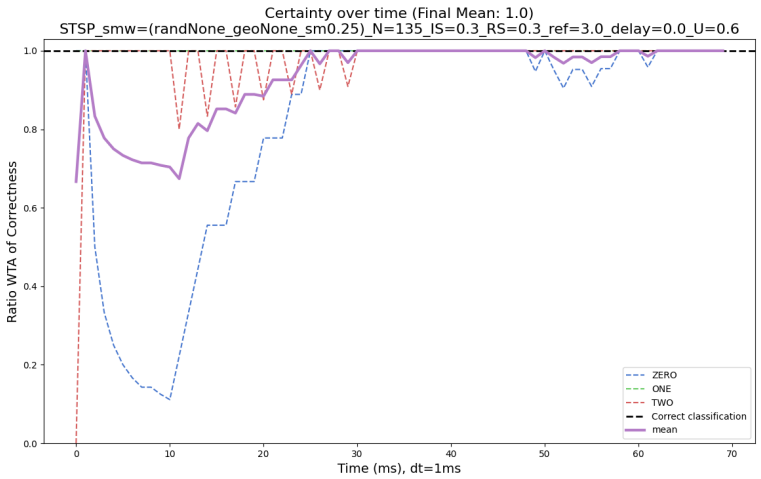


Figure 18: An example of a high performing reservoir for speech recognition. During the 700ms trial (70 moments of classification), certainty increases to 100% even after several brief periods of decrease.

Hyperparameter	Value: Occurrences
Learning	LIF: 17, STSP: 37, STDP: 23, LSTP: 23
Topology	rnd: 6, geo: 46, smw: 48
Input density	0.1: 50, 0.2: 26, 0.3: 24
Reservoir density	0.1: 42, 0.2: 30, 0.3: 28
p_{random}	0.1: 2, 0.2: 2, 0.3: 2
Dimensions	[15, 3, 3]: 20, [9, 5, 3]: 14, [27, 5, 1]: 12
β	0.0: 19, 0.33: 16, 0.66: 13
EI	False: 65, True: 35
Input feed	reset: 28, continuous: 72

Table 5: Hyper Parameter Occurrences in Top 100 Performers for speech recognition. Note that rnd topology and all values for rndp should be normalized to $\times 3$ for a fair comparison to other parameters. This is because rndp essentially defines reservoir density and was varied for the other two topologies as well.

However, not all reservoir configurations performed so effectively. As shown in Fig. 19, there is significant diversity in reservoir performance. This is a virtue of the task difficulty, limited training examples, and limited neuron population sizes. This performance spread is a favorable outcome, because it allows for a clean analysis of performance rankings so as to better understand what sort of hyperparameters might contribute to effectiveness.

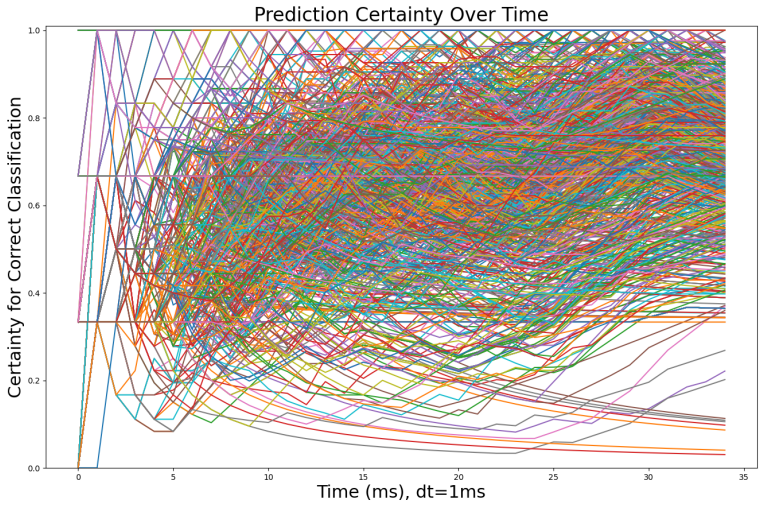


Figure 19: The certainty performance of all 1008 reservoir configurations. Note that most reservoirs perform better than random, while some are worse than guess, and a considerable number achieve perfect classification.

Through this large hyperparameter sweep, we can glean certain favorable configuration types. Analyzing the top 100 performers (see Fig. 20 (left) and Table. 5) in terms of final average certainty across all classes, we see which combinations of learning rules and topologies perform best given the other remaining parameters of the sweep (density, input feed, and excitatory/inhibitory behaviour). We find that STSP notably outperforms other learning rules in terms of functioning with a greater variety of other hyper-parameters, especially in combination with a small-world topology. It is also notable that the novel LSTP learning rule performs competitively well with the typical choice of STDP, though it involves more complex learning behavior.

With respect to the influence of inhibitory behaviour and input feed type on different learning rules and topologies, we can refer to Fig. 20 (right). It can be seen that STSP better exploits all meta conditions, where as LIF configurations are indifferent to these changes. All *learning* rules prefer a continuous input feed. This is also the case for all topologies (not shown). Small-world topologies better exploit all meta-conditions such that performance is more robust to all permutations of these conditions than other topologies.

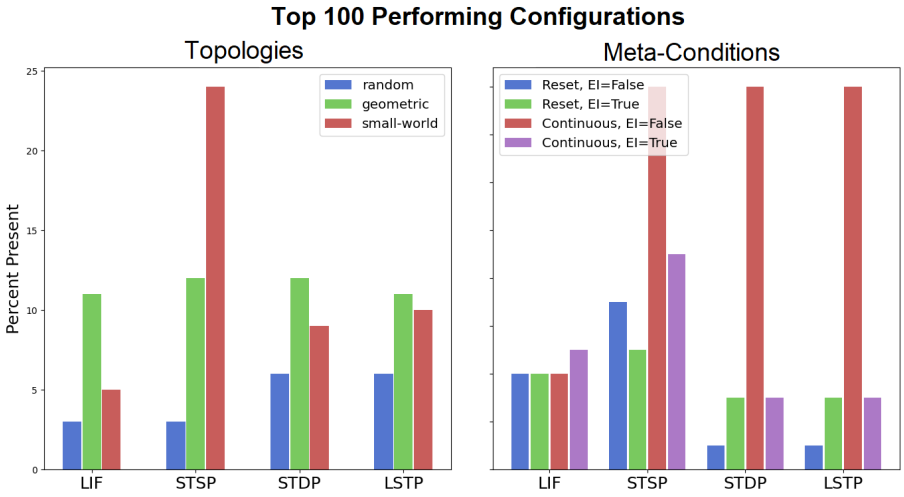


Figure 20: Occurrences of reservoir configurations of specified combinations for learning rules with topologies (left) and meta-parameters (right) in the top 100 out of 1008 LSM configurations.

8.3 Synaptic Dynamics and Firing

As described in section 5, (learning) rules define the behaviour of a system over some structure. We here visualize the connection of synaptic parameters to system behaviour (firing). Because we have already seen that small-world is a consistently performing topology across learning rules, and that continuous input feed and non-EI settings are generally favored, we will here isolate our investigation of synaptic parameters to these settings over each learning rule for high (randomly selected above 90% final certainty, table 6) and low (randomly selected below 50% final certainty, 6) performing configurations, in search of what constitutes good performance. Because learning rules displayed the most varied performance, they become our primary variable of interest. Fig. 21 displays reservoir firing, mean voltage, and synaptic parameter change, all in response to input for high and low-performing small-world continuous non-EI configurations over different learning rules. The values have been heavily normalized to superimpose to the range of 135 reservoir neurons on the y-axis.

Learning Rule	β	Input density	Reservoir Density	Performance (%)
High-Performing				
LIF	0.66	0.3	0.1	96
STSP	0.66	0.2	0.2	100
STDP	0.0	0.1	0.3	98
LSTP	0.33	0.1	0.2	92
Low-Performing				
LIF	0.0	0.2	0.1	52
STSP	0.66	0.3	0.1	60
STDP	0.33	0.3	0.2	54
LSTP	0.0	0.3	0.2	55

Table 6: High and low-performing small-world continuous non-EI configurations for each learning rule and their respective performances.

The relevance of input signal (purple) can here be seen. All input samples are 700ms, but the time that the stimuli of the spoken digit causes firing is shorter, usually in the range of about 100-600ms. The way different reservoirs respond to the non-firing 'down-time' in an input signal varies. High-performing LIF has some response to input (slight drops in firing between signals, greater voltage-change amplitude, some shuffling in firing pattern), but is fairly repetitive and constant. Low-performing LIF appears to essentially fire constantly and without change. Voltage-change amplitude is less and there is no discernable change in firing pattern (which groups of neurons are firing). High-Performing STSP maintains firing even during the interim time between input signals, most likely due to its calcium buildup and residual voltage levels. The firing is dynamic and still changes during the input transitions. Low-performing STSP does not maintain firing between input signals and the firing during the input signal is repetitive (non-dynamic). High-performing STDP and LSTP also have dynamic firing (changing over time and in response to input), but do not maintain firing between input signals). The a_{pre} and a_{post} synaptic parameters from equations 18 and 19 result in dynamic weight change that is clearly correlated with the presence of an input signal. Low performing STDP and LSTP have repetitive, non-dynamic firing, with less dynamic weight-updates and more sudden changes in a_{pre} and a_{post} that only appear to respond to the presence/absence of input and not the subtle within-input changes.

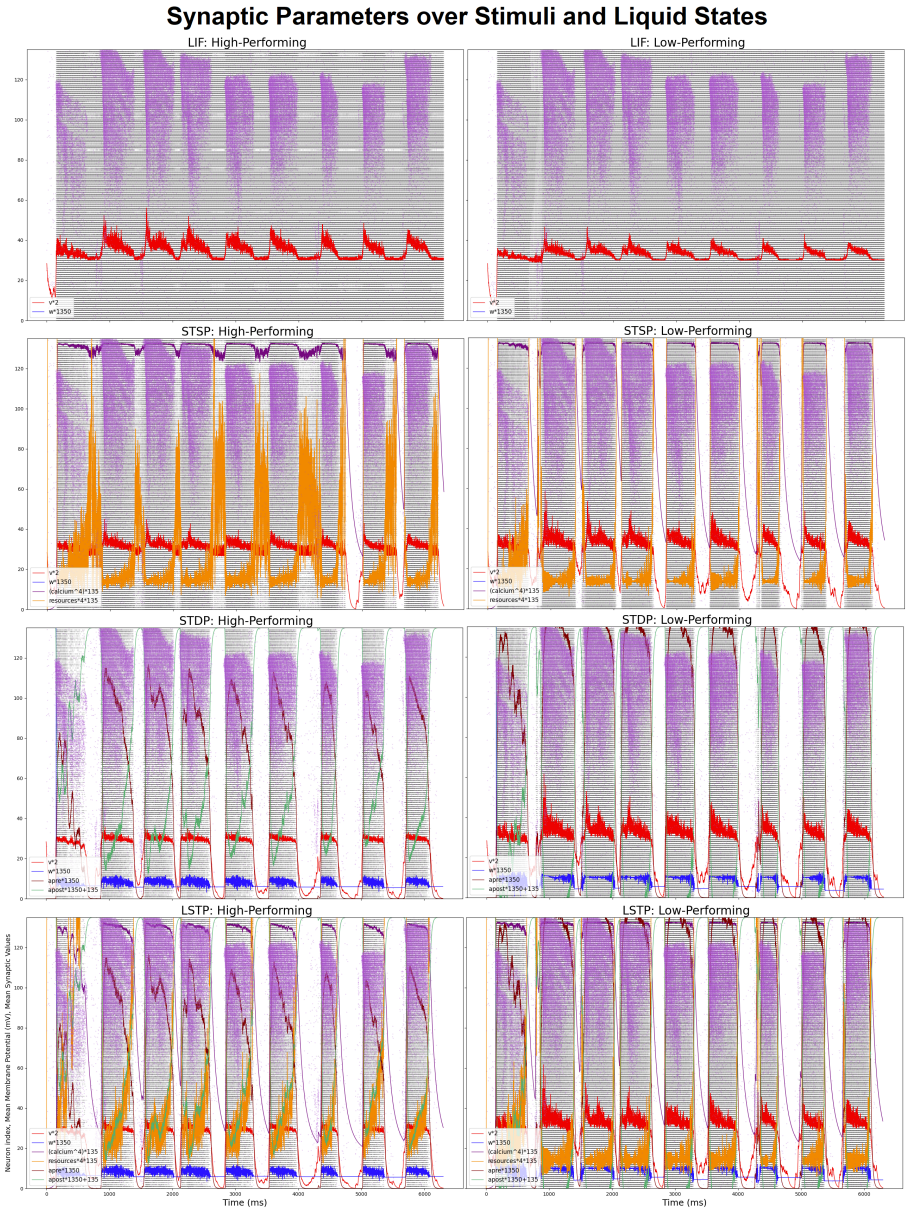


Figure 21: High (left) and low (right) performing configurations for continuous non-EI trials over entire 6300ms trials. Top-to-bottom: LIF, STSP, STDP, LSTP. Values have been heavily normalized (by factors show in legend) to superimpose to the range of 135 reservoir neurons on the y-axis. Purple dots indicate spoken digit input firing (normalized by a factor of 0.19286 to fit 700 input channels over 134 reservoir neurons), black dots indicate reservoir response firing. Dark blue is always mean membrane potential in mV for the whole reservoir and orange is always the mean weighting values.

To formalize differences in total firing as a function of learning rules, we here include Fig. 22. It becomes apparent that LIF configurations tend toward only two extremes (saturation and sparsity) in a bi-modal distribution. While STSP is second both in terms of highest and lowest firing amount, it is also distributed across the entire range of total firing. STDP and LSTP tend toward a much tighter range between about 150000 and 250000 spikes. These results cohere to Fig. 21, where it can be seen that only STSP and LIF configurations fire between input stimuli.

Figure 22: Total firing in response to all input for all reservoir configurations over total firing rankings. Because of the strong correlation of firing between STDP and LSTP, their yellow and green performance markers may be difficult to distinguish.

8.4 Reproducibility and Temporal Certainty

A necessary question that follows from the above performance results is *how reproducible are they?* Ideally, each configuration would be run hundreds of time on many different training/testing sets, but for the sake of limited compute, we here again look only the same high-performing small-world continuous non-EI LSM configurations as seen in Fig. 21 (table 6). Each configuration is run 100 times with random, unseeded initializations of voltages, weights, and small-world connectivity (random reconnecting via the β parameter from equations 3 and 4). Mean classification certainties (equation 27) for all test input samples from each class over all 100 reruns are tracked over time at along with the standard deviations. Fig. 23 illustrates an overall rise in certainty for all learning rules as the length of time the configuration has been exposed to the input stimuli increases. However, STDP and LSTP, which appear to be strongly correlated in both certainty and variance, experience a dip in certainty at around 600ms, which is about the time the input stimuli is removed (when the speaking of digit is complete, but the sample goes on for another 100ms, see Fig. 14). Note that STDP and LSTP here have different β parameters and therefore different connectivity, so their correlation must come from their shared long-term learning dynamics. LIF flatlines around the end of the input stimuli, and, interestingly, STSP continues to rise. This is not surprising because in Fig. 21 we already saw that high-performing STSP reservoirs may continue

to dynamically fire even after the input stimuli is withdrawn. Like with total firing, STSP and LIF again both exhibit much higher variance than the learning rules with long-term learning. STSP performs on average lower than STDP and LSTP, even by the end of the trial where it has an advantage through continued firing, but because of its greater variance, it at least sometimes performs higher than any other configuration ever did. This would explain why STSP achieved the highest results in the overall sweep–its variance spectrum sometimes reaches to higher performance than what is possible for other configurations at the final time step.

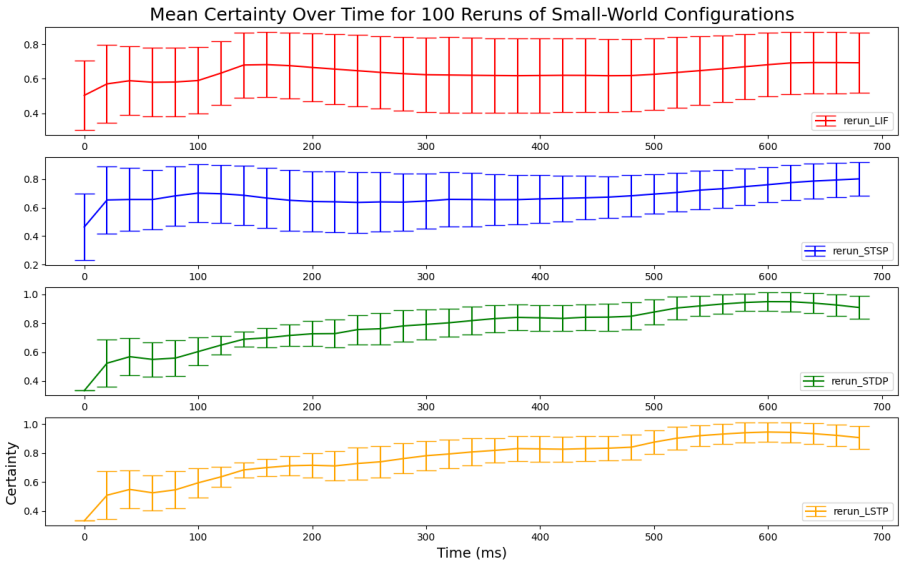


Figure 23: Mean C(t) (equation 27) for all test sample of each class for 100 reruns of high-performing small-world continuous non-EI configurations for each learning rule.

9 Representation Analysis

While tracking certainty is a useful metric for performance, it leaves much hidden about the inner-workings of a reservoir during a trial period. To gain insight on the dynamics of the system, state-space is observed in both compressed and full dimensionality for the Heidelberg speech recognition task.

9.1 PCA

The advantage of PCA for reservoirs is it allows the visualization of representations in three dimensions. This does not directly translate to empirical results, but rather acts as tool to gain intuitive understanding about state space. These results are therefore presented in this context.

9.1.1 Static Representations

At any given moment in trial time, the firing activity for a given input will occupy some position in state space. These high dimensional $\in \mathbb{R}^{135}$ are reduced to three dimensions based on variance across all inputs at that time, as described in section 7.4. Therefore, a point in 3D space corresponds to the relative representational position of a a given input at a given time in the reservoirs' state space. An example of this visualized compression may be seen in Fig. 24. This well-organized (organized by class) distribution of input samples in compressed PC state-space is indicative of many high-performing reservoirs, but not most. Reservoirs with this sort of organization in PC space are only found to perform well (never poorly), but many reservoirs with seemingly scrambled organization (not spatially organized by class) still perform well, and this is of course due to the loss of information in PC compression: meaning that reservoirs poorly organized in PC space may still be well-organized (and therefore linearly separable) in full-dimensions.

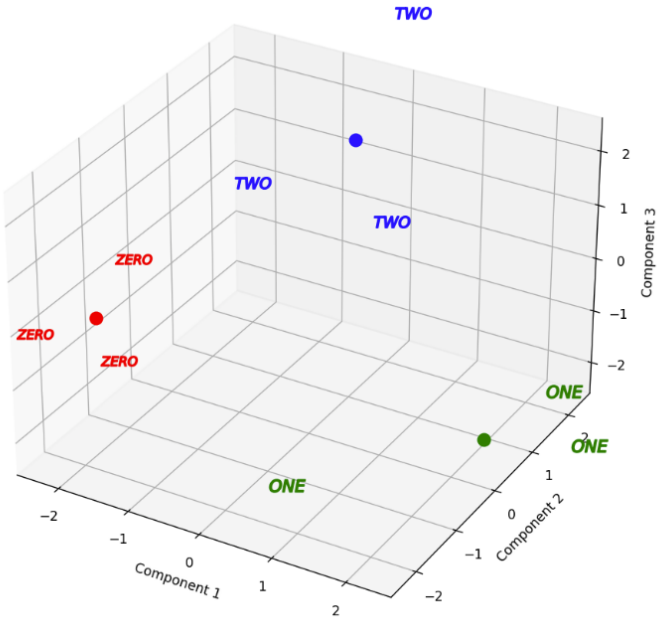


Figure 24: An example of a well-organized (high coherent separation) reservoir for speech recognition.

9.1.2 Representations Over Time

Consider the centroids (dots) from Fig. 24. These mark only the average position of each class representation at a specific time window $[t, t + 1ms]$. The trials, however, last $700ms$. To visualize these relative representational positions in PC space *over time*, the path of these centroids throughout the the duration of a trial may be traced. This process results in a typology of four archetypes of dynamical behaviour, shown in Fig. 25, which we assign the following names and qualitative definitions:

1. Pseudo-Noise: Appear to be random scribbles (may be better organized in higher dimensions)
2. Perfect Cyclical: Precise repeating of a few representational positions
3. Evolving Cyclical: A repetition of relative state representational positions that is drifting or wobbling in state space over time
4. Dynamic Encoding: Class representations follow distinct, non-repeating paths

Note that pseudo-noise paths are by far the most common across all configurations and performance levels. The remaining three are more rare, but tend

to be high-performing in terms of classification, so long as their class-centroid paths do not become entangled.

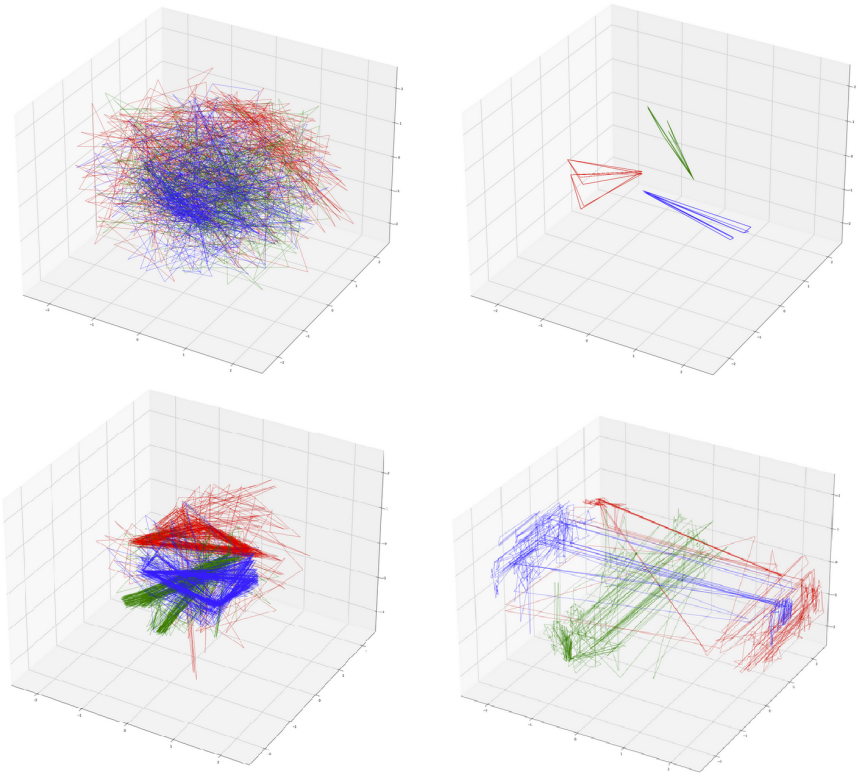


Figure 25: Archetypal PC paths of class centroids from 150ms to 700ms. Top Left: pseudo-noise, top right: perfect cyclical bottom left: evolving cyclical, bottom right: dynamic encoding

9.2 Organized Separation

As indicated by high-performing reservoirs that appear disorganized in terms of both static and temporal representations in PC space, there must be far more information in full-dimensional state space. Therefore, as detailed in sections 7.4, relative euclidean distances in full-dimensional space can be used to determine separation and organization in a more robust way. What is lost is of course visualizability, but nonetheless we may find ways to peak into high-dimensional behaviour. Fig. 26 demonstrates how average *organized separation* in terms of equation 26 is correlated both with performance and input-density. Note the fairly clear delineation of input density with distance, where lower-connectivity usually equates with less coherent organization. Furthermore, it seems that

below some threshold $Sep_{org} = 1000$ that performance tends to degrade. However, above this threshold organized separation makes less of a difference, and in fact, apparently an input density of 0.1 with slightly lower Sep_{org} performs most often in the top rankings. Also see that each input density seems to be fragmented into two different distance segments. All other reservoir variables were tested and the EI meta-parameter was found to be the culprit, as seen in Fig. 27, where EI being set to true results in lower organized separation for a given input density.

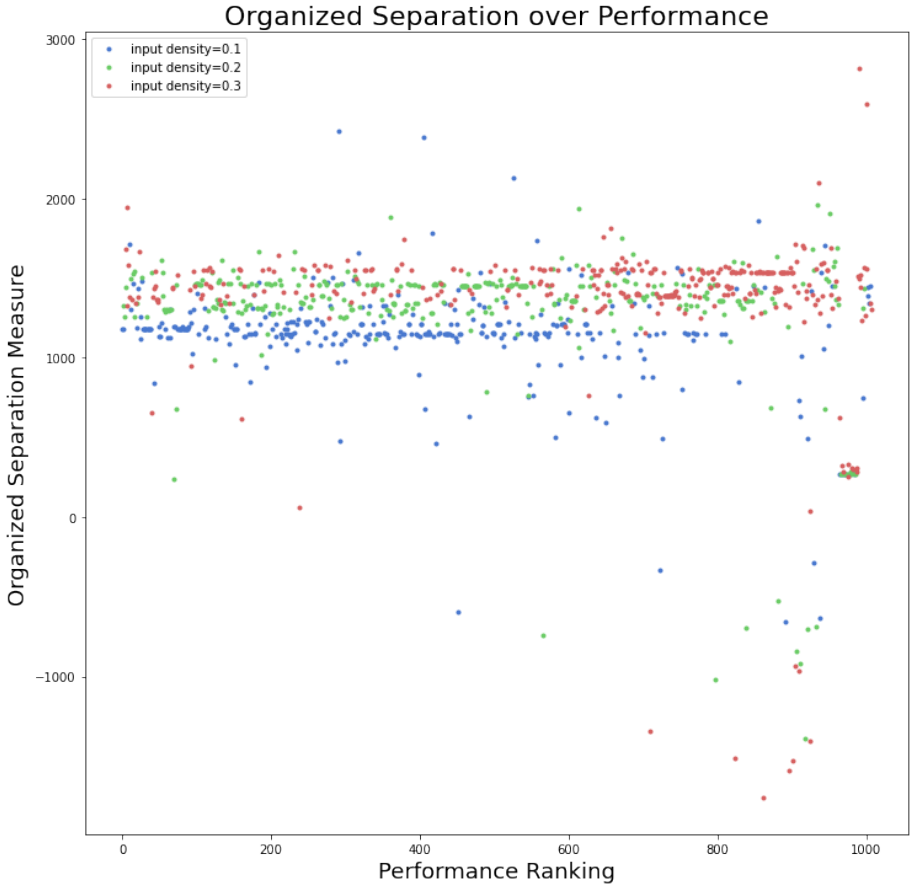


Figure 26: Coherent separation over performance ranking as defined by 26 for different levels of input density. input density=0.3 indicates a 30% chance of any input channel being connected to any reservoir neuron.

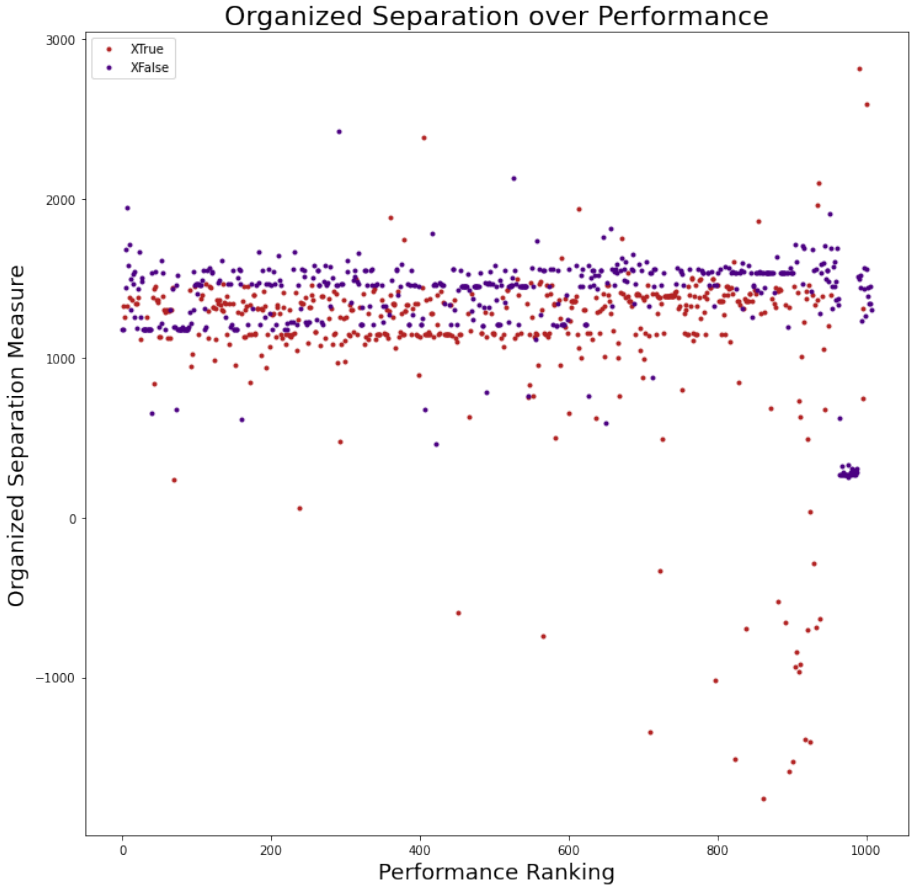


Figure 27: Distance over performance ranking for different EI behaviour settings. Brugandy dots corresponds to the EI settings [19] detailed in subsection 7.1 where inhibitory connections are included. Cobalt dots indicate all excitatory connections.

10 Discussion

Here results and analysis are interpreted and contextualized in the literature. It should be restated that due to their complexity, fundamental theory for LSMs is a difficult and vague territory. Results must always be seen in the exact circumstances of experimental design. We therefore here also include a caveats section, seeking to be aware of these limitations.

10.1 Learning Rules

10.1.1 Performance

The most salient finding in terms of performance, across all other parameter settings for both input domains, is that STSP is the strongest performing learning rule, occurring most often among the top performers for either task, given all other parameter settings. This is the case both with and without the EI parameters, the inclusion of which involve inhibitory behaviour that emulates the parameters of [44] used in [1]. In fact, STSP more often performed better without any inhibition, and thus distinguishes itself from the from the synaptic dynamics of [1, 14, 15, 18–20].

It is here speculated that STSP's efficacy is related to the time scale of information flow for this given task. STSP can be thought of as an attention mechanism, biasing firing activity to stay focused on a stream of related input. By being primed to have firing patterns related to recently seen input, and by responding very strongly to repeated input (in a short time frame), STSP essentially *focues* on a given input pattern. The parameters (from equations 8 and 7) of a 150ms time constant for calcium buildup decay and a 20ms time constant for resource replenishment are on the time scale order of both the image and speech recognition tasks. Poisson firing rates do not change at all for the MNIST task, and occur only for a short period of 120ms. While the Heidelberg spoken digit task involves trials of 700ms, the changes in firing rate that comprise the audio encoding of the digits occur more on the order of ≈20ms, as seen in Fig. 14. It is therefore not surprising that the learning rule best equipped for *short-term* change performed best on these tasks.

It may here be speculated that for a task involving greater time scales, the long-term learning of STDP may stand to benefit. Future research may be proposed to test whether the combined short and long-term learning of LSTP may be more robust to changes in time scale.

10.1.2 Synaptic Dynamics

The extended investigation of the role of learning on firing behaviour and performance visualized in Fig. 21 yields a number of insights.

LIF firing is sustained throughout all stimuli for low and high-performing reservoirs, but in no obviously meaningful way. This suggests that when an LIF reservoir performs well, it is simply a 'lookup table' whereby different stimuli hit and sustain some firing pattern without adjustment until a new stimuli

arrives. This behaviour would be represented by the 'perfect cyclical' dynamical behaviour in PC space. When an LIF reservoir performs poorly, it must be that these cycles overlap in representational space and are therefore not linearly separable. It can be seen that even sans-stimulus, the potentiation sustained by calcium buildup for high-performing STSP is sufficient to prolong input-related firing, allowing for more informative state vectors across the whole trial. We here use the word informative in the sense that firing patterns that are changing over time contain more information than firing patterns that simple repeat and therefore may be more easily distinguished by logistic regression learning if these patterns are unique to their respective classes. This would correspond to 'evolving-cyclical,' 'dynamic-encoding,' or 'pseudo-noise' in terms of our dynamical patterns typology because these are the dynamics that involve continuous change as a function of input. For 'pseudo-noise,' it would have to be the case that there is good coherent separation in higher dimensions of classification to be successful. It is likely the case that these dynamic firing patterns are unique to each input stimuli for STSP because its firing patterns are provoked by the stimuli and then further potentiated by calcium buildup. When STSP performs low, between-stimulus firing is absent and it is therefore probably a consequent of insufficient calcium levels for firing-persistence. For STDP and LSTP, between-stimulus firing is neither present in high or low-performing reservoirs. The Hebbian correlation established by STDP does not necessarily evoke stimulus-absent firing–only increasingly similar firing patterns for related input. Where informational density can be observed for these long-term learning rules is in the dynamics of the weight updates, caused by a_{pre} and a_{post}, which are more responsive overtime to input stimuli for high-performing configurations. This results in liquid state responses that contain more uniquely identifying information over the entire sample.

These observations are consistent with our performance results. STSP would be expected to have an advantage, as it continues to fire in response to stimulus throughout the entire input window of 700ms. For both resetting and continuous input feed, this would result in more abundant legible state vectors (instead of empty or redundant) for training the aggregate classifier. Because classification is made by summing predictions made across the entire sample, the total amount of information-rich state-vectors (even when there is no stimulus) will importantly contribute to final classification performance. It could be that with preprocessing of signals to only include the spoken-digit audio component (the actual stimulus), STDP performance would relatively improve.

10.1.3 Reproducibility

This effect is reflected as well in Fig. 23, where we see that the STDP and LSTP peak around the time the input stimuli stop at 600ms (note there is some variation for each input sample), while STSP continues to improve. However, at peak performance around 600ms, STDP and LSTP performance is comparable to that of STSP and moreover with less variance. The top performing STSP configurations are likely catching the higher end of the variance spectrum (this

is bound to happen sometimes for 252 experiments), and if enough compute were available for rerunning *every* experiment 100 times, then perhaps STSP would perform less highly. It is also important to here see that LIF is unreliably variant and may *only* achieve high-performance through luck-of-the-draw. It may be the case that with the right preprocessing method (to prune away any non-stimuli time from input samples) or recognition algorithm (to automatically determine when a stimuli begins and ends) that STDP and LSTP would on average outperform STSP.

10.1.4 STDP and LSTP

It is also of interest that STDP and LSTP appear to be tightly coupled in performance (Fig. 20), firing (Fig. 22), and temporal certainty/reproducibility (Fig. 27). Given that LSTP contains the dynamical synaptic parameters of both STSP and STDP, this is cause for question. It is here speculated that the short-term mechanisms of the STSP component do not keep up with the weight changes of the STDP component. STSP temporarily potentiates recent firing patterns by factoring in calcium buildup contributions to the membrane potential integration term in equation 9, but because this contribution is multiplicative with the weighting value at that synapse, and because that weighting value is changing through the Hebbian mechanisms of equations 20 and 21, it maybe be that this reinforcing mechanism does track the moving target. Colloquially, the attention mechanism of STSP may not be able to focus on the moving target of a learning reservoir population. Perhaps with more careful parameter tuning, and with differential terms that would more sensibly couple these dynamics, that short and long-term learning could better harmonize to the benefit of performance.

10.2 Topologies

While in total, small-world is the most commonly occurring topology in top performing reservoirs given all other hyperparameter settings (including in combination with STSP), the finding is not particularly pronounced, with 48, 46, and 18(weighted), being the speech-recognition occurrences of small-world, geometric, and random topologies in the top 100 configurations respectively. That topology does not make a strong difference in performance is consistent with the findings of [41] in ESN literature.

However, small-world here does perform marginally better in *both* input domains, and is by far the best topology for STSP in speech recognition and given STSP's overall superior performance, this is a finding of interest. We here again refer to the short-term nature of these classification tasks. Both tasks include spatial organization and short-term temporal dynamics. Therefore, it may be reasonable to expect that because small-world topology allows for the immediate dissemination of information across both local and global scales, it may have an advantage over random topologies, which do not preserve spatial information, and geometric topologies, which only pass information on local

spatial scales. It could be that given tasks with less spatial organization, random topologies may perform better, or given longer tasks, that geometric topologies may have more time to fully propagate information. It could be that small-world topologies pair particularly well with STSP for this specific task because it is a short-term task to which STSP can be highly responsive to, especially with the ability to propagate information locally and globally instantly and with preserved spatial organization.

10.3 Density

Input density (the probability p of any input channel connecting with any reservoir neuron) has inverse trends in the two task domains. In both cases, higher input density (lower sparsity) results in greater coherent separation. However, for the MNIST task, as this p increases, performance increases (see Table 4), whereas for speech recognition it is the opposite. We here venture to say that this is due to that the information encoded in the handwritten digits of the MNIST occupies relatively few input channels (all blank pixels are equivalent) and therefore a higher input density is necessary to capture the relevant features. For the speech recognition task, which involves already dense firing across many neurons, this may only serve to saturate the reservoir, provoking so much firing that it is difficult to tell apart reservoir responses to different input. Homeostatic adaption (automatic change in threshold to keep firing within a given range) may then benefit higher density reservoirs.

It is also of interest that for only a change in the combinations of input and reservoir density, performance is completely changed for each learning rule in table 6. This indicates a strong influence of density-balance on performance. However, general formula for the balance we observed and it appear to be unique for every reservoir configuration. There may be more subtle dynamics at play that warrant further investigation into density-balance, as already suggested by [18].

10.4 EI Meta-parameter

For both input domains, reservoirs that included inhibitory behaviour (EI = True) occurred less frequently in the top performing reservoirs given all other hyperparameters. This is in contrast to the general trend in literature to always use inhibition [1, 14–20]. Reasons for the literature being biased in this way may include biological plausibility [44], habit [51], or energy-saving sparsity [8]. Excluding inhibition always results in more firing by virtue of equation 5 and therefore would be more costly for hardware implementation. Firing always requires energy. The desired balance between energy cost and performance accuracy will of course be case-based.

10.5 Input Feed Meta-parameter

A constant stream of input without re-initialization proved to be the better method for both task domains. This is consistent with the literature [1,14,17,18]. The result is logical as it allows greater divergence (and therefore more unique firing patterns) from the initial conditions. Moreover, this embodies a more realistic approach to online neuromorphic systems, which will not know when one input ends and another begins. If, for whatever reason, there is some resting-state for a system that would define some standardized initial conditions, then it is worth noting that pure LIF neurons are here found to be ambivalent input-feed type. This is likely because they are not learning neurons and therefore have little to benefit by having more time with the input.

10.6 Representations

Much work is left to be done in terms of representation analysis. Findings here serve only to say that there indeed may be a useful typology of dynamical behaviour in state space and that it may be correlated with good reservoir performance. Our PCA static plots indicate that coherent separation occurs even in compressed dimensions. This aligns with the separability of PC representations found in [17, 18]. The class-centroid path plots indicate that this organization may be preserved over time to the benefit of performance. We do here speculate that these dynamical patterns correspond to the different firing dynamics of Fig. 21, but learning this relationship would require understanding these dynamical patterns in high-dimensions, which should in theory be possible to formalize.

PCA clearly leaves much mystery about behaviour in higher dimensions, but our findings using full-dimensional state space through equation 26 are consistent with the notion that coherent separation is related to better performance, as seen in 26. Therefore, like [15, 26], our definition of separation, here called *coherent separation*, does show to be related to performance, especially below the the apparent threshold at about $Sep_{org} = 1000$ below which performance tends to fall off. This threshold may approximate the requisite organized separation for linearly separability for the complexity of this particular task, above which greater Sep_{org} does not contribute to higher classification.

All said, it stands to reason that in both arenas, compressed and fully-dimensional, future research may lead to ways of deliberately biasing toward well-organized state space and particular dynamical behaviours that would increase classification performance.

10.7 Conclusions

In total, a number of conclusions can be drawn from the here presented work for these particular experimental conditions.

1. LSMs are suitable for both image and speech recognition, even given limited training data.

2. STSP performs best given all other hyperparameters on both image and speech recognition.
3. Small-world performs *marginally* best given all other hyperparameters on both image and speech recognition.
4. STSP and small-world perform best together.
5. STSP and LIF are more likely to fire between input stimuli
6. STSP and LIF have a greater perfromance variance
7. STDP and LSTP peak in performance at the end of the stimuli and have a much lower variance than STSP and LIF
8. The novel LSTP learning rule is competitive with the canonical STDP and outperforms LIF.
9. The dynamics of STDP have a greater influence that STSP on LSTP
10. Inhibitory weight initializations have a negative impact on performance but improve firing sparsity.
11. Continuous input feed is better for both input domains.
12. There appear to be specific archetypes of representational dynamics in PCA space that may correlate with performance and learning rules.
13. Coherent separation appears to correlate with performance, input density, and inhibitory/excitatory behaviour.

That all of the above listed findings hold for both the MNIST image recognition task *and* the Heidelberg spoken digit classification task may suggest some generalizability and more robust confirmation of this is left to future (more expansive) research.

10.8 Caveats

Given the complexity and sensitivity of reservoir computing, it is essential to contextualize findings with and awareness of possible caveats, which here are listed below.

- Results may be specific to any of the reservoir parameters held constant (such as population size).
- Results may be specific to the particular training samples used.
- Results may be specific to time scale of these particular tasks.
- High performing reservoirs may be the 'lucky ones', performing at the favorable end of their variance spectrum.

10.9 Future Works

It may be evident to the reader that the here presented research is somewhat sprawling in scope. The reason for this is simple: *dynamical systems are not simple*. The nature and presentation of the input, the reservoir and all of its many hyperparameters, and the particular readout mechanism employed all contribute to an explosion of possible liquid state systems, each of which may respond in totally different ways to any given input or reservoir initialization. It is not surprising then, that LSM literature is filled with fruitful findings, but without much in the way of axiomatic truths for good system design. Thus the study of LSMs themselves is somewhat constrained to a reservoir computing approach, where many complex interactions and transformations happen among researchers, and lessons are learned from the literature gestalt. Hopefully the work here has contributed something to this gestalt, especially in terms of learning rules, topologies, and representional organization. From this work, the author here speculates on a number of future possibilities.

- A more thorough sweep of all variables presented here, along with those held constant (such as reservoir size, membrane time constants, and synaptic variables) along with a greater cross-validation scheme, and repeatability of results.
- Preprocessing and/or pruning methods for input to work with only the most active moments of input stimuli.
- Synthetic learning rule that better harmonizes short and long-term learning through the coupling of STSP and STDP dynamics.
- Implementation of here presented systems on state-of-the-art neuromorphic hardware.
- Work on enforcing well-organized representations to the benefit of performance.
- Video recognition with each frame being presented on the order of 10 to 100ms as done with the image recognition task here.
- Hierarchical and modular reservoirs–it may be interesting to have the first few reservoirs process with STSP for short-term spatio-temporal features and with the later reservoirs using STDP for features of greater scale.
- Learning rules that learn to adapt to different task types by coming pretrained with certain triggers that respond to characteristics of different modalities and spatio-temporal scales.
- A rigorous mathematical definition of dynamical archetypes in full-dimensional space.
- Ivestigation into many-class experiments with definable and contintuous class differences, and how this could be achieved to develop a representational space that is meaningfully continuous and isomorphic to the input.

These above suggestions are only the most related to the work presented here, and at a greater scale LSMs should have a rich potential for future work, especially in the domain of video recognition, representation learning, and edge computing. This project has created at least as many questions as it has answered, and therefore it is the author's view that there is abundance of discoveries left to be made for reservoir computing and liquid state machines.

References

[1] Maass, Wolfgang, Thomas Natschläger, and Henry Markram. "Real-time computing without stable states: A new framework for neural computation based on perturbations." Neural computation 14.11 (2002): 2531-2560.

[2] Jaeger, Herbert. "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note." Bonn, Germany: German National Research Center for Information Technology GMD Technical Report 148.34 (2001): 13.

[3] Schrauwen, Benjamin, et al. "Compact hardware liquid state machines on FPGA for real-time speech recognition." Neural networks 21.2-3 (2008): 511-523.

[4] Roy, Subhrajit, Amitava Banerjee, and Arindam Basu. "Liquid state machine with dendritically enhanced readout for low-power, neuromorphic VLSI implementations." IEEE transactions on biomedical circuits and systems 8.5 (2014): 681-695.

[5] Wang, Qian, Yingyezhe Jin, and Peng Li. "General-purpose LSM learning processor architecture and theoretically guided design space exploration." 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS). IEEE, 2015.

[6] Schürmann, Felix, Karlheinz Meier, and Johannes Schemmel. "Edge of chaos computation in mixed-mode vlsi-a hard liquid." Advances in neural information processing systems 17 (2004).

[7] Nikolić, Danko, et al. "Temporal dynamics of information content carried by neurons in the primary visual cortex." Advances in neural information processing systems 19 (2006).

[8] Liu, Yu, Yingyezhe Jin, and Peng Li. "Online adaptation and energy minimization for hardware recurrent spiking neural networks." ACM Journal on Emerging Technologies in Computing Systems (JETC) 14.1 (2018): 1-21.

[9] Mead, Carver. "Neuromorphic electronic systems." Proceedings of the IEEE 78.10 (1990): 1629-1636.

[10] Misra, Janardan, and Indranil Saha. "Artificial neural networks in hardware: A survey of two decades of progress." Neurocomputing 74.1-3 (2010): 239-255.

[11] Seo, Jae-sun, et al. "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons." 2011 IEEE Custom Integrated Circuits Conference (CICC). IEEE, 2011.

[12] Merolla, Paul, et al. "A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm." 2011 IEEE custom integrated circuits conference (CICC). IEEE, 2011

[13] Lazar, Andreea, Gordon Pipa, and Jochen Triesch. "Fading memory and time series prediction in recurrent networks with different forms of plasticity." Neural Networks 20.3 (2007): 312-322.

[14] Verstraeten, David, et al. "Isolated word recognition with the liquid state machine: a case study." Information Processing Letters 95.6 (2005): 521-528.

[15] Goodman, Eric, and Dan Ventura. "Spatiotemporal pattern recognition via liquid state machines." The 2006 ieee international joint conference on neural network proceedings. IEEE, 2006.

[16] Zhang, Yong, et al. "A digital liquid state machine with biologically inspired learning and its application to speech recognition." IEEE transactions on neural networks and learning systems 26.11 (2015): 2635-2649.

[17] Jin, Yingyezhe, Yu Liu, and Peng Li. "SSO-LSM: A sparse and self-organizing architecture for liquid state machine based neural processors." 2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH). IEEE, 2016.

[18] Wijesinghe, Parami, et al. "Analysis of liquid ensembles for enhancing the performance and accuracy of liquid state machines." Frontiers in neuroscience 13 (2019): 504.

[19] Kaiser, Jacques, et al. "Scaling up liquid state machines to predict over address events from dynamic vision sensors." Bioinspiration biomimetics 12.5 (2017): 055001.

[20] Soures, Nicholas, and Dhireesha Kudithipudi. "Deep liquid state machines with neural plasticity for video activity recognition." Frontiers in neuroscience 13 (2019): 686.

[21] Ivanov, Vladimir, and Konstantinos Michmizos. "Increasing Liquid State Machine Performance with Edge-of-Chaos Dynamics Organized by Astrocyte-modulated Plasticity." Advances in Neural Information Processing Systems 34 (2021): 25703-25719.

[22] Azambuja, Ricardo de et al. "Short-term plasticity in a liquid state machine biomimetic robot arm controller." 2017 International Joint Conference on Neural Networks (IJCNN) (2017): 3399-3408.

[23] Hazan, Hananel, and Larry M. Manevitz. "Topological constraints and robustness in liquid state machines." Expert Systems with Applications 39.2 (2012): 1597-1606.

[24] Xue, Fangzheng, Zhicheng Hou, and Xiumin Li. "Computational capability of liquid state machines with spike-timing-dependent plasticity." Neurocomputing 122 (2013): 324-329.

[25] Reynolds, John JM, James S. Plank, and Catherine D. Schuman. "Intelligent reservoir generation for liquid state machines using evolutionary optimization." 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 2019.

[26] Norton, David, and Dan Ventura. "Improving liquid state machines through iterative refinement of the reservoir." Neurocomputing 73.16-18 (2010): 2893-2904.

[27] Liu, Chuang, et al. "Optimizing the Neural Structure and Hyperparameters of Liquid State Machines Based on Evolutionary Membrane Algorithm." Mathematics 10.11 (2022): 1844.

[28] David H. Wolpert, The supervised learning nofreelunch theorems, in: Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications, WSC 2006, 2001, pp. 25–42.

[29] Lukoševičius, Mantas, and Herbert Jaeger. "Reservoir computing approaches to recurrent neural network training." Computer Science Review 3.3 (2009): 127-149.

[30] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, 29(6), 141–142.

[31] Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, Friedemann Zenke, December 6, 2019, "Heidelberg Spiking Datasets", IEEE Dataport, doi: https://dx.doi.org/10.21227/51gn-m114.

[32] Diehl, Peter U., and Matthew Cook. "Unsupervised learning of digit recognition using spike-timing-dependent plasticity." Frontiers in computational neuroscience 9 (2015): 99.

[33] Lyon, Richard. "A computational model of filtering, detection, and compression in the cochlea." ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing. Vol. 7. IEEE, 1982.

[34] Slaney, Malcolm. "Auditory toolbox." Interval Research Corporation, Tech. Rep 10.1998 (1998): 1194.

[35] Lichtsteiner, Patrick, Christoph Posch, and Tobi Delbruck. "A 128×128 120 dB 15μ s latency asynchronous temporal contrast vision sensor." IEEE journal of solid-state circuits 43.2 (2008): 566-576.

[36] Bollobás, Béla, et al. Probabilistic combinatorics and its applications. Vol. 44. American Mathematical Soc., 1991.

[37] Gilbert, Edgar N. "Random graphs." The Annals of Mathematical Statistics 30.4 (1959): 1141-1144.

[38] D.J. Watts and S.H. Strogatz, dynamics of small-world networks," Nature 393, 440 (1998).

[39] A.-L. Barabasi and R. Albert, of scaling in random networks," Science 286, 509 (1999).

[40] Erdős, Paul, and Alfréd Rényi. "On the evolution of random graphs." Publ. Math. Inst. Hung. Acad. Sci 5.1 (1960): 17-60.

[41] Liebald, Benjamin. "Exploration of effects of different network topologies on the ESN signal crosscorrelation matrix spectrum." Bachelor of Science (B. Sc.) thesis, International University Bremen, spring (2004).

[42] Antoniou, Ioannis E., and E. T. Tsompa. "Statistical analysis of weighted networks." Discrete dynamics in Nature and Society 2008 (2008).

[43] Mongillo, Gianluigi, Omri Barak, and Misha Tsodyks. "Synaptic theory of working memory." Science 319.5869 (2008): 1543-1546.

[44] Markram, Henry, Yun Wang, and Misha Tsodyks. "Differential signaling via the same axon of neocortical pyramidal neurons." Proceedings of the National Academy of Sciences 95.9 (1998): 5323-5328.

[45] G.-q. Bi and M.-m. Poo, "Synaptic modification by correlated activity: Hebb's postulate revisited," Annual review of neuroscience, vol. 24, no. 1, pp. 139–166, 2001.

[46] Auer, Peter, H. Burgsteiner, and Wolfgang Maass. "The p-delta learning rule for parallel perceptrons." submitted for publication (2001).

[47] Pearson, Karl. "LIII. On lines and planes of closest fit to systems of points in space." The London, Edinburgh, and Dublin philosophical magazine and journal of science 2.11 (1901): 559-572.

[48] Cheng, Casey. "Principal Component Analysis (PCA) Explained Visually with Zero Math." Medium, Towards Data Science, 22 Mar. 2022, https://towardsdatascience.com/principal-component-analysis-pca-explained-visually-with-zero-math-1cbf392b9e7d.

[49] Marcel Stimberg, Romain Brette, Dan FM Goodman (2019) Brian 2, an intuitive and efficient neural simulator eLife 8:e47314 https://doi.org/10.7554/eLife.47314

[50] Wolff, Michael J., et al. "Revealing hidden states in visual working memory using electroencephalography." Frontiers in systems neuroscience 9 (2015): 123.

[51] Kuhn, Thomas S. The structure of scientific revolutions. Vol. 111. University of Chicago Press: Chicago, 1970.

[52] Lee, Chankyu Sarwar, Syed Panda, Priyadarshini Srinivasan, Gopalakrishnan Roy, Kaushik. (2020). Enabling Spike-Based Backpropagation for Training Deep Neural Network Architectures. Frontiers in Neuroscience. 14. 119. 10.3389/fnins.2020.00119.

[53] Madadi Asl, Mojtaba Valizadeh, Alireza Tass, Peter. (2018). Propagation Delays Determine the Effects of Synaptic Plasticity on the Structure and Dynamics of Neuronal Networks. 10.13140/RG.2.2.28601.88166.