MesoNet: A Spatio-Temporal Neural Architecture for Massively Energy Efficient Unsupervised Learning

Krishna Bhatt (krishbhatt2019@gmail.com)
BASIS Independent Silicon Valley Grade 10
1290 Parkmoor Ave, San Jose, CA 95126

Abstract

The demand for energy-efficient and scalable unsupervised learning in low-power edge computing applications—spanning robotics, space exploration, and autonomous systems—has grown significantly. Traditional artificial neural networks (ANNs) struggle with high energy consumption due to their numerous matrix operations. To remedy this, spiking neural networks (SNNs), inspired by biological neurons, offer great promise with immense energy efficiency and event-driven computation, which can be further exploited by neuromorphic chips. However, the commonly used STDP learning algorithm struggles with accuracy and scalability due to its local nature. Against this backdrop, I present MesoNet, a novel neural architecture for improved low level generalization and enormous energy gains over modern ANNs. To achieve such goals, I conduct a novel, in-depth bifurcational study of SNN learning dynamics, analyzing network behaviors surrounding a key learning rate parameter during convergence. Leveraging these insights, I develop a distributed plasticity model that modulates the learning rate through bio-realistic plasticity evolution preventing undesirable convergence patterns. Having the effect of improving convergence speeds and the learning capability of the network, I incorporate this innovation with a split-and-merge architecture to form hierarchical representations of data and achieve phenomenal feature abstraction and scalability. I demonstrate the improved performance on a series of datasets (MNIST, CIFAR-10, and CIFAR-100) that indicate a 21% improvement of accuracy over previous state-of-the-art SNNs and 47% computation reduction from ANNs, showing great promise for future development of SNNs as a viable alternative to ANNs as well as the practicality of bifurcational study in the neuromorphic space.

Table of Contents

No outside assistance was received in the creation of this paper.

1. Introduction

As AI applications permeate into various parts of technology, unsupervised learning is gaining increased awareness in the edge computing use-case. The need for online learning is expanding into real-world applications such as automated driving, robotics, space exploration, and the internet of things (Yazici et al., 2023). However, modern unsupervised algorithms based on artificial neural networks (ANNs) are computationally hungry and incur massive energy costs (Pfeiffer & Pfeil, 2018) becoming a real obstacle for when energy usage is limited. Concurrently, spiking neural networks (SNNs), regarded as the third generation of neural networks (Maass, 1997), are being designed to explore the intricacies of intelligence and brain function by replicating the behavior of biological neurons (Roy et al., 2019) through incorporation of biology-inspired features. These networks are built upon two key elements: spiking neurons and synapses, which are organized into hierarchical layers to form cohesive networks. Recently, SNNs have gained substantial attention in the research community due to their unique advantages, such as low power consumption (Roy et al., 2019), event-driven processing (Petro et al., 2019), online learning, and massive parallelism (Pfeiffer & Pfeil, 2018).

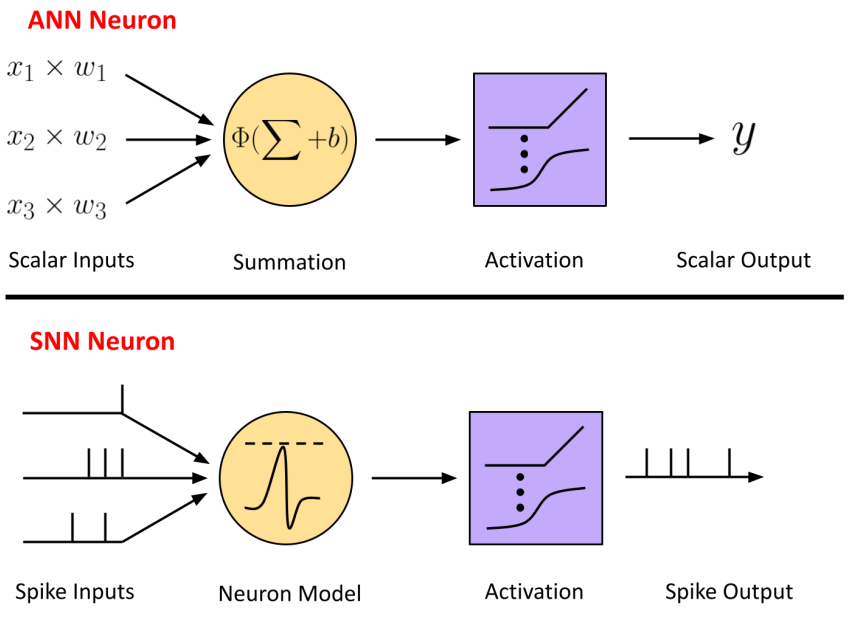

Figure 1. Compared to traditional ANNs, which process information synchronously in a serialized and vectorized manner incurring enormous memory and computational costs through matrix multiplication, SNNs process information asynchronously through sparse biological *spikes* with spike timings encoding information.

These abilities stem from their sparse spike-based computation and communication, which can be exploited to achieve higher computational efficiency by specialized neuromorphic hardware (Han & Roy, 2020). SNNs have potential to solve many problems regarding unsupervised edge computing due to these characteristics. However, as a nascent field of study, SNNs have yet to achieve accuracy comparable to state-of-the-art (SOTA) ANNs (Lu & Sengupta, 2024).

Overview of SNN Architectures. Currently SNNs can be broadly divided into three major categories: *supervised, unsupervised,* and *converted*. Supervised SNNs denote the network's use of labelled datasets to guide training, specifically through following the gradient of their loss function. Various strategies have been developed to do this including backpropagation through time and use of surrogate gradients (Neftci et al., 2019) as the spiking nature of neurons makes them non-differentiable; however none have yet met the performance of classical ANNs. Unsupervised SNNs denote the algorithmic training of SNNs without the use of labelled datasets. Algorithms such as spike-timing dependent plasticity (STDP) and DECOLLE (Kaiser et al., 2020) adjust weights based on biological synaptic plasticity to learn the intrinsic structures of input samples. Converted SNNs utilize pretrained ANNs and transfer weights (Gao et al., 2023) to SNNs to circumnavigate training difficulties. This paper will focus on the unsupervised class of SNNs for their implications in online learning and edge computing.

Spike Timing Dependent Plasticity. Most modern unsupervised SNNs are trained through spike-timing dependent plasticity due to its computational resourcefulness, proven effectiveness, and bio-plausibility (Bell et al., 1997). Intuitively based upon the common adage, "neurons that fire together wire together," STDP-based learning rules modify synaptic weights interconnecting a pair of pre and post-synaptic neurons based on the degree of correlation between the respective spike timings. This process of STDP builds on biological concepts of Long Term Potentiation

(LTP) and Long Term Depression (LTD) (Song et al., 2000) , and can be described as the following:

$$w_a = \sum_{pre} \sum_{post} W(t_{post} - t_{pre})$$

$$W(\Delta t) = \begin{cases} A_{pre} \times e^{-\frac{\Delta t}{\tau}} & \Delta t > 0 \\ A_{pre} \times e^{\frac{\Delta t}{\tau}} & \Delta t < 0 \\ 0 & else \end{cases}$$

where t_{pre} and t_{post} are the spike timings of the pre and post-synaptic neurons respectively, A_{pre} and A_{post} are the learning rates governing plasticity, and τ is a learning time constant. LTP can be defined by $W(\Delta t)$ when Δt is positive and LTD when Δt is of a negative value. However, the locally-based learning in individual synapses means that STDP has various limitations: In a preliminary study by Diehl and Cook (2015), standard conductance-based STDP was utilized to train a fully connected shallow network to a sub-optimal performance level of 95% accuracy on the toy dataset MNIST with an extremely slow learning speed of 900,000 iterations to complete training. Additionally, catastrophic forgetting (Panda et al., 2017), a phenomenon observed in continual learning and STDP where knowledge of old tasks are lost when new ones are learned is frequently observed within STDP based neural networks. As such, the performance of STDP-trained networks seems to be limited beyond basic tasks like handwritten digit recognition. To address some of these issues, convolutional SNNs, similar to their deep learning counterparts, were proposed to learn hierarchical input representations (Kheradpisheh et al., 2017) and improve scalability. However, it remains to have lower performance than SOTA ANNs on most popular benchmark datasets. As described by Rathi et al. (2022), STDP trained SNNs face two primary issues, that if surpassed, “would enable a new generation of low-cost

(area/power/resource requirement), local and unsupervised learning framework, as opposed to their non-spiking counterparts (ANNs).” These two problems include:

(a) The scalability of convolutional SNNs remains to be unknown without the typical issues of information loss.

(b) It is not yet known if STDP alone can enable deeper layers to learn complex input representations and generalize on low-level features and clusters.

Dynamic Bifurcational Analysis. Concurrently, Dynamic Bifurcation Analysis, a mathematical framework for studying how the behavior of a system changes due to parameters, has found relevance in exploring machine learning. It provides a strong intuition into the underlying dynamics of neural network states (Sussillo & Abbott, 2009) and thus has become a key research focus regarding network convergence during learning (Dauphin, Pascanu, Gulcechre, et al., 2014). Mathematically, it is used in dynamical system theory to investigate critical transitions in nonlinear systems, where small changes in control parameters cause the system to shift from one qualitative state to another (Onuki, 2002), (Kuznetsov, 2013). These transitions, known as bifurcations, mark important structural changes in a system’s dynamics such as moving from stability to oscillations to chaos. The existence of bifurcations can be described by the simple ordinary differential equation:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x, \lambda)$$

where x $\in \mathbb{R}^n$ represents the state variables of the system and $\lambda \in \mathbb{R}^m$ is a parameter vector. The function f : $\mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ describes the dynamics of the system. The equilibrium points of the system occur when $f(x, \lambda) = 0$. A bifurcation occurs when the parameter vector λ leads to a qualitative change in the number or stability in the equilibrium points. These changes can

manifest as sudden transitions between network states, oscillations, or chaotic behavior (Strogatz, 2018). As described by Beer (1995), various forms of neural networks can be naturally modeled as nonlinear dynamical systems which exhibit complex behaviors, including fixed points, limit cycles, and even chaotic dynamics, depending on their architecture and parameter settings. These qualities would make this system ideal for analyzing unintuitive spatiotemporal correlations within SNNs.

Contributions. In this paper, I will be focused on designing a STDP-based SNN that improves in all learning efficiency (speed), network sensitivity and stability, and high-level generalizations to create a spatiotemporal neural architecture for energy efficient unsupervised learning. To sum up, I will propose a plasticity based STDP learning architecture comprising of three main contributions:

(1) Provide a theoretical framework describing a system of STDP synapses as a bifurcation dynamical system, extending the work of Zhang et al. (2021) proving a LIF Neuron as a bifurcational dynamical system to .

(2) Propose a distributed plasticity architecture, MesoNet, based on an Inception-like split and merge architecture (Ghantiwala, 2023) and demonstrate its performance in generalization on complex recognition tasks going past MNIST.

(3) Provide a preliminary computational cost estimate of the MesoNet framework against conventional SNN methods and SOTA unsupervised ANN systems to demonstrate computational efficiencies.

1.1 Related Works

Prior to this work, there have been multiple investigations into the scalability of SNNs. Diehl and Cook (2015) introduced the first computationally-feasible implementation for double-layer STDP by leveraging event-driven simulation to reduce computational costs. Similarly, Kheradpisheh et al. (2017) proposed a hierarchical SNN framework for visual pattern recognition, demonstrating that layered architectures have scope for improved scalability and performance. Building on unsupervised learning, Masquelier and Thorpe (2007) demonstrated that STDP can learn invariant visual features for spatio-temporal input streams. More recently, Panda et al. (2017) introduced a forgetting mechanism that combines STDP with controlled forgetting for improved feature extraction and high level generalization. However, few studies developed a calculable approach to improved convergence and feature extraction for learning algorithms. Two studies, Zhang et al. (2021) and Zhang et al. (2024) provided further insight into the system dynamics of LIF neurons through bifurcation analysis, however they were limited to the spike-timing dynamics of the neurons in a supervised setting. Outside the realm of SNNs, studies regarding deep ANNs (Dauphin, Pascanu, Gulcechre, et al., 2014) (Saxe et al., 2013) have explored the internal dynamics of learning regarding stability and dynamical isometry. This paper will extend these previous works in attempting to bridge the gap between theory and practice for unsupervised SNNs to develop a system for improved learning based upon theoretical observation.

1.2 Preliminaries

As mentioned previously, SNNs contain three critical components that describe its function: the neuron model, input encoding, and learning connection. Here I will describe these components of the proposed MesoNet.

Neuron Voltage Model. In this work, I utilize the leaky-integrate-and-fire (LIF) neuron model for its prevalence in SNN applications as well as for the ease with which it can be analyzed. Here we can define a general form of the LIF neuron as follows:

$$\tau_m \frac{\mathrm{d}V}{\mathrm{d}t} = -V(t) + Rf(I(t))$$

where $I(t) = (I_1(t), \ldots, I_M(t))$ denotes the M-dimensional input signals, $V(t)$ indicates the membrane potential of the neuron at time t, τ_m and R are hyperparameters for the membrane time constant and membrane resistance, respectively, and f is the aggregation function such that $f(I(t)) = \sum_{j=1}^{M} W_j I_j(t)$, where W_j is the learnable weight connection corresponding to the j-th input channel. Based on the *spike response model* scheme (Gerstner, 1995), the LIF neuron has a general solution with the boundary condition $V_{rest} = 0$ as follows:

$$V(t) = \sum_{j=1}^{M} W_j [\int_{t'}^{t} \exp(\frac{t' - s}{\tau_m}) I_j(s) \mathrm{d}s]$$

where t' denotes the last firing time such that $t' = \max \{ s \mid V(s) = V_{firing}, s < t \}$. An output stimulus $S(t)$ is then generated whenever the membrane potential $V(t)$ reaches a certain threshold V_{firing} (firing threshold). This procedure can be formulated with a spike excitation function:

$$f_e : V \rightarrow S \text{where} S(t) \stackrel{\text{def}}{=} \left\lfloor \frac{V(t)}{V_{firing}} \right\rfloor$$

After firing, the membrane potential is reset to a lower value V_{rest} (resting voltage). Some researchers take the delayed-firing behaviors of neuroscience by adding an absolute refractory period or a refractory kernel to define a time period of inactivity after each spike (Dumont et al., 2017). However, for simplicity, I will not be considering such modes in this paper.

Neural Input Encoding. Compared to ANNs, SNNs use spikes for representation of data, and as such, real input data should be converted into a spiking version through neural encoding prior to entering the network. There are two popularized forms of neural encoding in current literature: *rate-based* and *temporal*. In temporal encoding, data is encoding through the temporal distance between firing events ((Han & Roy, 2020). In rate based coding, information is represented through the mean firing rate of a neuron over a given time period. In poisson rate coding, a subcategory of the latter, the normalized analogue value is compared to a randomly generated number from a poisson distribution. The neuron fires if the value is greater than the number and otherwise remains inactive (Quiroga et al., 2005). Although temporal encoding is considered more energy efficient for their "data-rich" sparse spiking, it faces issues with noise amplification for the reduced number of spikes. Rate-based encoding faces few of such issues and can also be practically recorded by a Dynamic Vision Sensor (Anumula et al., 2018), making it the optimal algorithm for the purposes of this paper.

Basic Connection. Putting these two building blocks along with the previously discussed STDP model, we can develop a basic 1:1 isolated connection that represents the simplest dynamic within the network. This basic form can be visualized as in Figure 2.

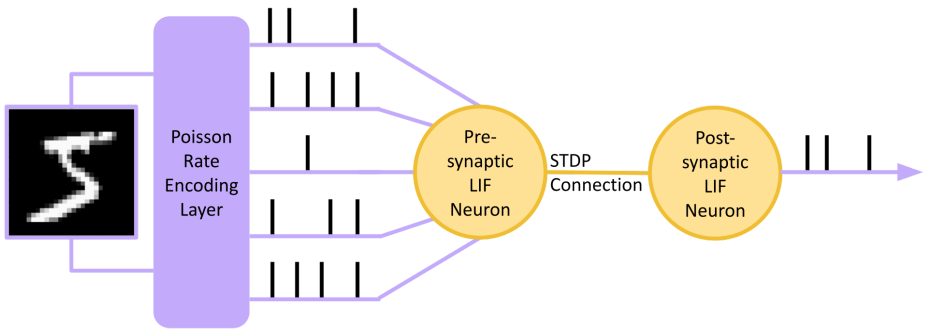

Figure 2. Visualization of the Basic 1:1 connection

This formation will prove to be useful later in scaling local dynamics to network-order representations.

Eigenvalues and Abstract Equations. For a system of first-order linear differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}\begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} + \begin{pmatrix} f_1(t) \\ \vdots \\ f_n(t) \end{pmatrix}$$

the compact algebraic formulation would be

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} = A\begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \text{ with } A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

These equations are related to the observation variables $u_1, \ldots, u_n$. Therefore, the eigenvalues (spectrum values) of the matrix (operator) **A** lead to the evolution rules of the dynamical system. The eigenvalues $\lambda_1, \ldots, \lambda_n$ can be computed as solutions to the the characteristic equation:

$$\det(A - \lambda I) = 0$$

2. Theoretical Findings and Methods

2.1 Bifurcational Dynamics of SNNs

In order to begin the bifurcation analysis of STDP synapses, I need to first formalize it as a bifurcation dynamical system. From the STDP learning rules described in Section 1, we can reformulate the equation as a ordinary differential equation (ODE) such that:

$$\frac{\mathrm{d}w}{\mathrm{d}t} = -\alpha w + f(\Delta t)$$

where α is the natural decay rate of the weight set to zero in standard STDP schemes, w is the weight of the synapse at time t, $f(\Delta t)$ is the STDP learning rule described as $W(\Delta t)$ in Section 1,

and Δt is defined as $t_{post} - t_{pre}$ where t_{post} and t_{pre} are the post and pre-synaptic spike timings respectively. This can be reduced to the following:

$$f(\Delta t) = \text{sgn}(\Delta t) \cdot \eta \cdot \exp(-\text{sgn}(\Delta t) \frac{\Delta t}{\tau_{learn}})$$

where the sgn function is defined by $|x| \div x$, η is the learning rate simplified as $A_{pre} - A_{post}$ and τ_{learn} is the learning time constant. In this section, I will investigate the bifurcation properties of these hyperparameters in basic 1:1 connections. Now I present my first theorem as follows:

Theorem 1. *Given the initial condition* $u(0) = u_{rest}$ or $u(t') = u_{rest}$ and $t_{pre} = 0$, *the dynamical system of weight updates given by a full layer of basic 1:1 connections is a bifurcation dynamical system, and the control parameters* α *and* η *are the corresponding bifurcation hyper-parameters*.

Proof. I start by analyzing the basic 1:1 connection by decomposing the STDP ODE into a function of w by eliminating the Δt parameter. First we assume that $t_{pre} = 0$ and as such $\Delta t = t_{post} - 0$. From there, we can determine the spike timing of the post-synaptic neuron as using the SRM model of LIF neurons as described in Section 2.1. This can be re-written as the following, assuming $t' = 0$:

$$\int_0^t \exp(\frac{-s}{\tau_m}) \times I_{syn}(s) \mathrm{d}s = \frac{V_{thresh}}{W_j(t)}$$

We can then estimate the value of the integral as a series by discretizing the time domain and decomposing I_{syn} which would give use the following:

$$\sum_{s=0}^{t} \exp(\frac{-s}{\tau_m}) \cdot W_j k(s) = \frac{V_{thresh}}{W_j}$$

where k(s) are spike inputs from the presynaptic neuron, for the purposes of this proof will remain constant as 1. Solving for the geometric series, we arrive at:

$$\Delta t(w) = t_{post} - 0 = -\tau_m \ln(1 - \frac{V_{thresh}}{w_j^2}(1 - \exp(-\frac{1}{\tau_m}))) - 1$$

where t_{post} is the spike time of the post-synaptic neuron given a constant input stream. A bifurcation is known as a critical point in a system at which a small smooth change in parameters leads to a sudden topological change in behavior (Onuki, 2002). To now determine the characteristics of such a bifurcation, I can calculate the Jacobian of dw/dt which introduces the gradient of the system. Formally defined, suppose a function $f: \mathbb{R}^n \to \mathbb{R}^m$ is a function which takes a point $x \in \mathbb{R}^n$ and produces vector $f(x) \in \mathbb{R}^m$ for which each of its partial derivatives exists on $\mathbb{R}^n$. The Jacobian matrix of f is defined such that its $(i, j)^{th}$ entry is $(\partial f_i / \partial x_i)$. (*Jacobian | Definition of Jacobian in English by Oxford Dictionaries*, n.d.). The Jacobian of matrix dw/dt, though use of chain rule, is defined $J(w)$ as:

$$J(w) = \frac{\mathrm{d}}{\mathrm{d}w}(-\alpha w + f(\Delta t(w))) = -\alpha - \frac{2\eta\tau_m}{\tau_{learn} w} \cdot \exp(\frac{-\mathrm{sgn}(\Delta t)\Delta t}{\tau_{learn}})$$

Based on this equation, both the weight update ODE and its Jacobian are dominated by the learning rate η, the decay rate α, and the time constants τ_m and τ_{learn}, determining the behavior of its eigenvalues. This means that the determination of the learning rates and decay rates are critical in deciding convergence behavior and the stability of the system. If $J(w)$ is computed as being greater than 0, this would signify a stable system for convergence of weights. If it is greater than such, it could represent an unstable system with issues of divergence, oscillations, etc. (Bengio & Delalleau, 2011). Next, I extend these

findings to a layer of STDP synapses of basic 1:1 connections. By referencing each synapse within the same context we get:

$$\frac{\mathrm{d}w}{\mathrm{d}t}_N(w,t) = -\alpha|w| + f(\Delta t(w))$$

where w is a N-dimensional vector that the i-th element represents the weight of the i-th STDP synapse. Similarly, the Jacobian matrix can be calculated as:

$$J_N(w) = -\alpha - \frac{2\eta\tau_m}{\tau_{learn}|w|} \cdot \exp\left(\frac{-\mathrm{sgn}(\Delta t_N)\Delta t_N}{\tau_{learn}}\right)$$

Similar to the isolated 1:1 connection, the network Jacobian is directly related to the learning rate η as well as the decay rate α and other time constants. A bifurcation occurs when $J(w)$ as well as the eigenvalue of the system p crosses its critical point 0 (occurs when α crosses its critical point α_{crit} or η crosses its critical point η_{crit}). Therefore $\mathrm{d}w/\mathrm{d}t$ coincides with a bifurcation dynamical system associated with η. This completes the proof.

Based on the results above, the decay rate α, learning rate η, and time constants τ_{m} and τ_{learn} play a crucial role in the convergence of weights across a layer of STDP synapses, directly impacting how weights are determined and the learning efficiency/capability of the network through the Jacobian.

In conventional STDP architectures, all of these hyperparameters are preset and static such that given a hyperparameter x, $x_1 = x_2 = \ldots = x_{\mathrm{N}}$. Conventionally, α is set to 0, η is set as a positive parameter, and τ_{m} and τ_{learn} are given as positive. As such, tying the entire network's synapse functionality to pre-set hyperparameters (η, τ_{m}, τ_{learn}) greatly weakens the representation ability of the network. Additionally, the network's dependence on various parameters would lead to the emergence of various fixed points and, in conjunction, a variety of undesirable behaviors such as

saddle-nose bifurcations and transcritical bifurcations (Mondal et al., 2021), which would disrupt convergence and introduce patterns such as oscillations, divergence, and destroyed fixed points. A much better manner in controlling relationships between fixed points would be to dynamically set the learning rate η such that bifurcation behavior can be controlled in order for the network to converge rapidly on accurate generalized features. However, further insight is required into bifurcations related to information transfer rather than simply weights to craft such an algorithm. Therefore, I introduce my second theorem:

Theorem 2. *Given the initial condition* $u(0) = u_{\text{rest}}$ or $u(t') = u_{\text{rest}}$ and $t_{\text{pre}} = 0$, *the dynamical system of multi-layer fully connected STDP layer is a bifurcation dynamical system, and the control parameters* α *and* η *are the corresponding bifurcation hyper-parameters*.

Proof. For determining the evolution of STDP synapses' states we can begin by defining a Lyapunov-like energy function for a singular basic 1:1 connection. A Lyapunov energy function $L(w)$ can be defined for an ODE, $w' = u(w)$ where $w \in \mathbb{R}^n$ if it satisfies the following criterion (Hirsch et. al):

(a) $L(w) > 0$ for all $w \neq 0$, and $L(0) = 0$. This indicates that $L(w)$ is analogous to energy, and the equilibrium point $w = 0$ has zero "energy."

(b) Along the trajectories of the system, the derivative of $L(w)$, $L'(w) = (\partial L / \partial w) \times u(w)$, must be less than or equal to zero, indicating Lyapunov stability, or strictly less than zero, indicating asymptotic stability.

In the scenario of modeling the network synaptic "energy", we can consider two major components: the information flow of the synapses as kinetic energy, and the structural

organization of the network as the potential energy. To determine information flow, we can begin by utilizing the mutual information flow metric proposed by Guo et al. (2024):

$$I(s;r) = H(r) - H(r|s)$$

where $I(s; r)$ is the mutual information flow, r is the spike response, s is the input spike train, and H is a measure of entropy. However, we can approximate the information flow using deterministic relationships such as Δt and how they evolve under the influence of synaptic weights w, as described in Theorem 1 as $\Delta t(w)$. This substitution can be made as Δt represents the functional coupling between pre and post-synaptic spikes, and thus, the sensitivity of Δt to changes in w reflects the degree to which weights influence the temporal structure of spikes, representing information flow. This gives us the following as the total sensitivity of the network:

$$S_N(w) = \sum_{j=1}^{N} \frac{\partial \Delta t_j}{\partial w_j} = \sum_{j=1}^{N} \frac{2\tau_m}{w}$$

where $S_N(w)$ is the function for network sensitivity of w and N is the number of basic 1:1 connections within the network. This can then be coupled by multiplying the weight update equation by w, as typical of Lyapunov energy functions (Vu & Turitsyn, 2015), to produce:

$$L_k(w,t) = (\frac{1}{2}w\cdot(-\alpha w + f(\Delta t(w))) - \frac{2\tau_m}{w})$$

where $L_k(w)$ is the kinetic energy function of a single 1:1 connection. We can measure potential energy for the neuron to retain information as its divergence from an optimal ratio of excitatory to inhibitory neurons (Alreja et al., 2022). Excitatory neurons are defined as

exc = $\{n \in N \mid \eta(n) > 0\}$, and by opposite function, inhibitory neurons are defined as *inh* = $\{n \in N \mid \eta(n) < 0\}$. Thus, we can define the total energy function of a network as:

$$L_N(|w|,t) = \sum_{i=1}^{N} L_k(w_i,t) + (E_i + \beta I_i)^2$$

where N is the number of neurons in the network, $|w|$ is a vector of all synaptic weights wherein w_i is the synaptic weight of the i-th neuron, E_i is the aggregate excitatory output ($w \times S_i(t)$) of the i-th neuron, I_i is the aggregate inhibitory output of the i-th neuron, and β is the pre-set inhibitory to excitatory (I:E) activity ratio. This network energy function satisfies all requirements previously defined, thus L_N is a Lyapunov-like energy function. Direct calculation shows that:

$$\frac{\mathrm{d}L_k}{\mathrm{d}t} = \frac{1}{2}(\alpha w - \mathrm{sgn}(\Delta t)\cdot\eta\cdot\exp(\frac{-\mathrm{sgn}(\Delta t)\Delta t}{\tau_{learn}})) + \eta\cdot\exp(\frac{-\mathrm{sgn}(\Delta t)\Delta t}{\tau_{learn}}) + \frac{2\tau_m}{w^2}$$

Thus we see that the kinetic energy is dominated by the learning rate, and in the case of a non-zero decay rate, α as well. Considering the potential energy of the system, the learning rate directly influences the prominence of excitatory or inhibitory neurons through its polarity and magnitude. A bifurcation of system energy would occur when L_N and its eigenvalues formulated from ($\partial L_k / \partial w$) crosses its critical point 0 (occurs when α crosses its critical point α_{crit} or η crosses its critical point η_{crit}). An eigenvalue p greater than zero would lead to an unstable manifold while one less than zero would lead to stability. Therefore, a system of a multi- layer STDP SNN comprised of a basic 1:1 connections coincides with a bifurcation dynamical system associated with η. This completes the proof.

This work extends Theorem 1 in proving that the values of the decay rate α and learning rate η play a direct role in determining the energy of the network, and thus, how it evolves over time in relation to information flow and network architecture. If α is set to 0 as per convention, and η is preset as $\eta > 0$, the system of a multi-layer fully-connected STDP SNN would be a diffusion system where kinetic energy and thus information flow will increase layer after layer and potential energy remains static through the network. Again, rather than statically initializing hyperparameters before runtime, a much more efficient method would be to dynamically adjust such rates such that SNNs can autonomously determine the apt mode at each synapse (ie. dissipating, conservative, or diffusive), to selectively retain information. In summary, traditional algorithms with fixed decay and learning rates impede the flexibility of SNNs, including their ability to selectively retain information and effectively converge to apt weights. To achieve an autonomously adaptive system, the learning and decay rates should be learnable and diverse.

2.2 Distributed Synaptic Plasticity Model

In order to develop a learning algorithm capable of addressing the short-comings of STDP regarding generalization and convergence, learning rates should be made learnable and diverse. For the purposes of this paper, I will set α as 0 as traditionally set across SNNs (Araki & Hattori, 2023) as alterations beyond this goes beyond the scope of this paper. An intuitive method of accompœlishing variable learning rates would be to parameterize and train them in parallel with the synaptic connection weights. However, this would introduce various new complexities as the STDP learning scheme is dependent on the learning rate such that both cannot be optimized at the same time.

An alternative method to develop learning rates would be to dynamically adjust them over time as a resultant of weight updates. In this learning scheme, weight updates would be distributed

amongst upstream synapses under a relevance filter that asserts blame of mismatched spike timings on the product of external synapses in addition to its own weight. On every externally sourced weight update, a given synapse would increment its learning rate, thus associating η with a quantified measure of "sureness" of the synapse's weight connection. Over time, this learning rate would converge to zero under a lack of external weight updates. Figure 3 visualizes this.

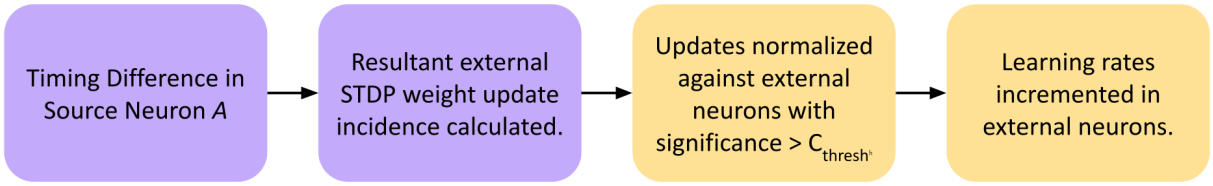

Figure 3. Flowchart describing the DSP algorithm

This distributed synaptic plasticity (DSP) model provides various advantages unique to its function that would improve MesoNet's functionality. Firstly, DSP extends past the local restraint of conventional STDP algorithms and allows blame for spike-timing differences to be shared across related neurons which would improve global generalization of higher order features as well as improve convergence. Secondly, DSP is an inherently greedy algorithm, prioritizing weight updates for external synapses that provide the greatest change, which would indicate improved convergence speeds. Thirdly, DSP provides a bio-plausible function to learning by mimicking interneuron coordination described by Luo (2021).

Formal Definitions of DSP. In this section, I will detail each redesigned aspect of the distributed synaptic plasticity model. Formally, the STDP equation can be reformulated to calculate the external incidence as the following:

$$w_{ext} = k \cdot \sum_{pre} \sum_{post} W(t_{post} - t_{pre})$$

where w_{ext} is the external weight update incidence, and k is a constant determining the neuron's external effect. For all synapses downstream of source synapse A, the significance value, an indicator for how the synapse contributes to A, is be calculated as such:

$$C_d = \frac{\eta_d}{1 + \gamma \cdot r(A, d)}$$

where C is the significance value of downstream neuron d such that $d \in D$ and D is the set of all downstream neurons, γ is the distance scaling constant, and $r(A, d)$ is the functional distance between source neuron A and d. This functional distance is defined as:

$$r(i, j) = \min_{\text{paths}} \sum_{k \in p} \left(\frac{1}{W_k}\right)$$

where *paths* is the set of all paths p from source neuron i to target neuron j, k denotes each edge or synapse of the path, and W_k is the weight of synapse k. This process is implemented through Dijikstra's algorithm, a prevalent path-finding algorithm, which solves for the shortest distance in $O(v + e \log v)$ time (Aramendia, 2023)**.** Upon the computation of the external weight incidence w_{ext}, each downstream neuron part of set $k = \{k \in D, \eta_k > \eta_{thresh}\}$ where η_{thresh} is the minimum plasticity threshold has its weight update computed as such:

$$\Delta w_d = \eta_d \left(\frac{C_d}{\sum_k C_k} \cdot w_{ext}\right)$$

where Δw_d is the computed weight update. However, outside of these distributed weight updates, the learning rate can be modeled as its own system by the following ODE:

$$\frac{\mathrm{d}\eta}{\mathrm{d}t} = -\frac{\eta_d}{\tau_p}$$

where τ_p is the stabilizing time constant. Thus, plasticity decays over time with a lack of weight updates. Upon a weight update, η_d is updated as follows:

$$\Delta\eta_d = \frac{\Delta w_d}{\eta_d^2}$$

Ensuring that Δw_d is divided by η_d^2 prevents any runaway growth where weight updates lead to increased plasticity, leading to further weight updates.

Bifurcation Dynamics of DSP. In addition to the intuitive positive effects DSP would have on a SNN. Its characteristics also make it favorable when considering the network's new bifurcation dynamics. To demonstrate this, I propose my third theorem:

Theorem 3. *Given the initial conditions* $\eta_d(0) = 1$, $u(0) = u_{rest}$ or $u(t') = u_{rest}$, $t_{pre} = 0$, and the hyperparameters λ_{ij} *are all greater than zero, a SNN consisting of DSP-based 1:1 connections has at most* 2^{N-1} *bifurcation solutions for which there exist no saddle-nose or transcritical bifurcations.*

Proof. I start this theorem by firstly reformalizing the dynamical system describing the basic 1:1 connection based on the distributed synaptic plasticity model. Although introducing new components, the fundamentals remain the same and can be described as such:

$$\frac{\mathrm{d}w}{\mathrm{d}t} = f(\Delta t(w), \eta)$$

where w is the weight of the synaptic connection, t is the time of operation, η is the learning rate, Δt is the spike-timing difference between post and pre-synaptic neurons, and $f(\Delta t(w), \eta)$ is the STDP function defined as such:

$$f(\Delta t, \eta) = \text{sgn}(\Delta t) \cdot \eta \cdot \exp(-\text{sgn}(\Delta t) \frac{\Delta t}{\tau_{learn}})$$

where $sgn(x)$ defines the sign function $|x| / x$, and τ_{learn} is the STDP learning time constant. Δt remains defined as it was in Theorem 1. However, the function f now takes η as a learnable parameter rather than a pre-initialized constant. It's dynamics assuming a constant spike input would be modeled as such:

$$\frac{\mathrm{d}\eta}{\mathrm{d}t} = -\frac{\eta}{\tau_p} + \frac{1}{\eta^2}\frac{\mathrm{d}w}{\mathrm{d}t}$$

where τ_{p} is the stabilizing time constant. Following the steps from Zhang et al. (2021), we can then find the solutions to the system**.** Firstly, I can obtain an algebraic representation of the energy function from Theorem 2 as:

$$\frac{\mathrm{d}L_k}{\mathrm{d}t} = P_\lambda \cdot (\frac{1}{2}\exp(\frac{-\text{sgn}(\Delta t)\Delta t}{\tau_{learn}})(-\text{sgn}(\Delta t)+1)) + \frac{2\tau_m}{w^2}$$

with $P_\lambda = A + B_\lambda$ where A is a diagonal array of learning rates η across all neurons and the eigenvalues of matrix B_λ are $\beta_1, \ldots \beta_{\text{n}}$. In this manner, the eigenvalues of P_λ can be calculated as the sum of A and B_λ such that $p_{\text{i}} = -\eta + \beta_{\text{i}}$. We can now determine the bifurcation solutions under DSP based upon this solution for the eigenvalues. Taking a system of two 1:1 connection for simplicity, such that

$$A = \begin{pmatrix} \eta_1 & 0 \\ 0 & \eta_2 \end{pmatrix} \text{ and } B = \begin{pmatrix} 0 & \lambda_1 \\ \lambda_2 & 0 \end{pmatrix}$$

Let the discriminants $D_1 = 2\eta$ and $D_2 = \eta^2 - \lambda_1\lambda_2$. When $D_1^2 - 4D_2 = \lambda_1\lambda_2 \geq 0$, P_λ has two real eigenvalues such that

$$p_1 = \frac{-D_1 - \sqrt{D_1^2 - 4D_2}}{2} \text{ and } p_2 = \frac{-D_1 + \sqrt{D_1^2 - 4D_2}}{2}$$

where p_1 must be less than zero and p_2 is indefinite. If $\lambda_c = \eta^2$ is set as the critical threshold, the solutions to the kinetic energy function would be controlled by the pair of bifurcation eigenvalues

$$\begin{cases} p_1 = \frac{-D_1 - \sqrt{D_1^2 - 4D_2}}{2} < 0 \\ p_2 = \frac{-D_1 - \sqrt{D_1^2 - 4D_2}}{2} \begin{cases} < 0, \ \lambda_1\lambda_2 < \lambda_c; \\ = 0, \ \lambda_1\lambda_2 = \lambda_c; \\ > 0, \ \lambda_1\lambda_2 > \lambda_c \end{cases} \end{cases}$$

Based upon this framework, as long as the product of λ_1 and λ_2 is greater than zero, there is at least one non-trivial solution to the kinetic energy function. When both eigenvalues are negative (ie. $\lambda_1\lambda_2 < \lambda_c$), the entire system becomes energy dissipating wherein all synapses constrain information flow. When $\lambda_1\lambda_2 = \lambda_c$, the system becomes dissipative in relation to p_1 and conservative in relation to p_2. Finally, when $\lambda_1\lambda_2 > \lambda_c$, the system becomes dissipative in relation to p_1 and diffusive in relation to p_2. We can assert with this framework that for N synapses, there are at most 2^{N-1} solutions to the kinetic energy equation. With this understanding of bifurcation dynamics in relationship to eigenvalues, I can now investigate the effects of the independent plasticity dynamics. Considering the equation for background plasticity updates, we can develop an ODE through direct calculation solving for the second critical point p_2 as:

$$\frac{\mathrm{d}p_2}{\mathrm{d}t} = \frac{\eta}{\tau_m} + \frac{1}{4} \cdot \frac{2D_1(-\frac{2\eta}{\tau_m}) - 4(-\frac{2\eta^2}{\tau_m} - \lambda_1\frac{\mathrm{d}\lambda_2}{\mathrm{d}t} - \lambda_2\frac{\mathrm{d}\lambda_1}{\mathrm{d}t})}{\sqrt{D_1^2 - 4D_2}}$$

> Based on this solution, the dynamic development of the critical point is directly controlled by the learning rate parameter. Taking the case of an inhibitory synapses, such that $\eta < 0$, ($\partial p_2 / \partial t$) can be confidently declared as negative for both the first term through direct observation and the second term given that $D_1 = 2\eta$. This means that negative values of η will create a progressively dissipative system proportional to its value where information capacity and the energy of connection is lost layer by layer. The opposite is true for excitatory synapses, such that $\eta < 0$, I can declare ($\partial p_2 / \partial t$) as positive, meaning positive values of η will create an increasingly diffuse system proportional to its value. Critically, I can assert that the DSP algorithm avoids both saddle-nose bifurcations and transcritical bifurcations as described in Section 2.2 as critical points diverge with lack of spiking input over time, creating a stable network for fast convergence. This completes the proof.

These results show that the novel proposed DSP model has the capability to develop learnable and diverse eigenvalues describing the network while preventing any unfavorable behavior such as saddle-nose bifurcations and transcritical bifurcations that were demonstrated to have existed in the standard STDP scheme in Theorem 1. These characteristics effectively enable DSP to achieve faster and more accurate convergence across sample inputs and selectively retain information due to the trainable aspect of the eigenvalues to switch between modes (dissipative, conservative, diffusive).

Synapse Flipping. DSP effectively optimizes the kinetic energy of the system to ensure smooth convergence, however it does not address the potential forces pertaining to the network's architectural design. To review, the network's potential forces can be described as:

$$L_{Np}(|w|,t) = \sum_{i=1}^{N}(E_i + \beta I_i)^2$$

where L_{Np} describes the network level potential force, $|w|$ is a vector of all synaptic weights, N is the number of neurons in the network, E_i is the aggregate excitatory output ($w \times S_i(t)$) of the i-th neuron, I_i is the aggregate inhibitory output of the i-th neuron, and β is the pre-set inhibitory to excitatory (I:E) activity ratio. To ensure that the network approaches the optimal ratio β, I implement a synapse flipping algorithm such that if $\eta > \eta_{thrsh}$ and $w = 0$,

$$w \overset{\text{def}}{=} -\text{sgn}(w) \cdot r$$

where r is the weight reset value typically set as $w_{initial}$. This ensures a healthy balance of excitatory and inhibitory synapses on a need-based system. In summary, in this section I developed a Distributed Synaptic Plasticity and synapse flipping algorithm that takes its foundations from bifurcation dynamics in order to create a system of diverse eigenvalues that enables speedy and unimpaired convergence.

2.3 Implementation

In order to maximize improvement of feature extraction and accuracy, MesoNet is architecturally designed to leverage the characteristics of DSP-based unsupervised STDP. In order to accomplish this, I utilize the foundations of the model proposed by Meng et al. (2021) to create a Split and Merge Architecture derived from the highly popularized Inception model from the ANN setting (Szegedy et al., 2015). In this framework, input images split into multiple pathways to be processed by various convolutional filters (eg. 3×3, 5×5, 7×7) before being concatenated back together. This methodology has two major benefits (Ghantiwala, 2023): (1) The split-and-merge module can process multi-scale spatial information to improve feature detection.

(2) It improves the network's parallelism as various pathways can operate concurrently. This architecture can be summarized by Figure 4. Different from Meng et al. (2021)'s work, each partition then has various horizontally connected "neural processing cores" consisting of simple fully-connected DSP-STDP based LIF neurons. This has the effect of drawing district boundaries within abstractions for DSP's credit assignment to maintain intent within the network. Additionally, this enables further parallelism on neuromorphic hardware wherein neural cores correspond to the time-multiplexed neuromorphic cores on chips such as Loihi (Abderrahmane et al., 2019). Furthermore, each core contains "gated channels" for lateral information pass-through, similar to biology (Lai & Jan, 2006).

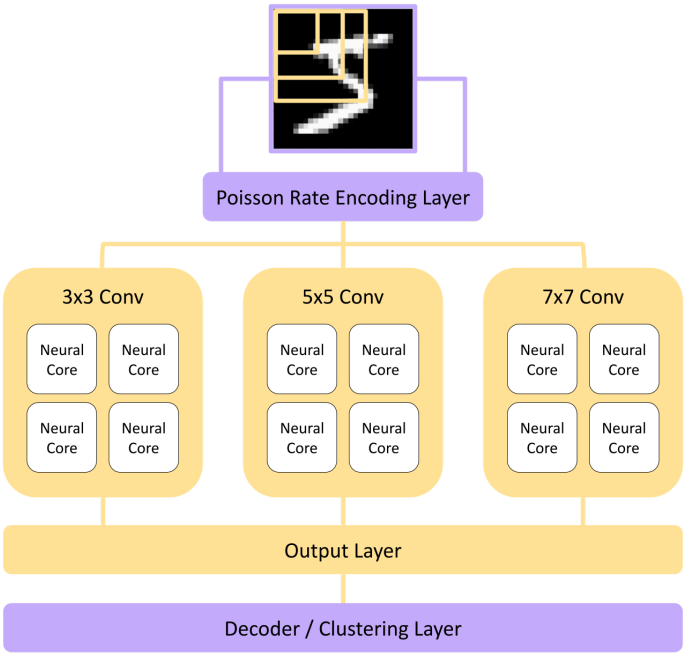

Figure 4. MesoNet's Inception-Inspired Split-And-Merge Architecture.

Network Simulation. I have implemented MesoNet utilizing Python 3.11 and the Brian2 simulator (Stimberg et al., 2019) that provides extensive control over spiking neural networks. Experiments were run on an Ubuntu machine with the Brian2CUDA extension that provides parallelization on the machine's GTX 1080 graphics card.

3. Experimental Findings

In this section, I conduct various experiments on benchmark datasets to evaluate the functional performance of MesoNet. These procedures were formulated to answer the following

investigative question: *How does the performance of MesoNet– specifically accuracy, learning efficiency, and computational cost– compare with state-of-the-art SNNs and ANNs in unsupervised image classification tasks?*

3.1 Experimental Set-Up

Datasets. For this study, I will be utilizing MNIST, CIFAR-10, and CIFAR-100 datasets, widely used benchmarks in the field of computer vision. MNIST (Lecun et al., 1998) is a collection of 70,000 grayscale handwritten digits (0-9), each with a resolution of 28x28 pixels, and serves as a fundamental dataset for evaluating algorithms. CIFAR-10 (Franky, 2022) contains 60,000 color images with 32x32 pixel resolution, divided into 10 diverse classes such as airplanes, birds, and automobiles. CIFAR-100 (GeeksforGeeks, 2024) extends CIFAR-10 to 60,000 color images of 32x32 resolution, but instead divided into 100 classes.

SOTA Contenders. For direct comparison, I used the basic 2 layer connection proposed by (Diehl & Cook, 2015) as a baseline and will be comparing MesoNet to the Adaptive Synaptic Plasticity model (Panda et al., 2017) and CSNN (Kheradpisheh et al., 2017). I will also be comparing MesoNet to the Invariant Information Clustering (IIC) algorithm, the SOTA unsupervised ANN at the time of writing proposed by Ji et al. (2019).

Parameters. The preprocessing stage for each of the datasets will be the same as those used by Pillow et al. (2005) as described in Section 1.2. Each static image will be transformed into a spike train using the poison encoding scheme. The decoder layer will take inputs from the previous convolutional output layer, as described in Figure 4 and determine the output label based upon a voting scheme as described by Meng et al. (2021). Table 1 describes the hyperparameters set for each of the SNN contenders.

Table 1. Hyperparameters for all SNN Contenders

Parameter	MNIST	CIFAR10	CIFAR100
Encoding Length	300	350	450
Learning Rate	0.01	0.005	0.001
Membrane Time Constant	2 ms	2 ms	2 ms
Synapse Time Constant	1 ms	1 ms	1 ms
DSP Distance	-	-	-

For MesoNet, I have initialized with the same parameters and a randomly generated initialization of weights. Additionally the DSP distance constant will be set to 3 for all three datasets. For IIC, I will be utilizing the same hyperparameters as given in their paper (Ji et al., 2019). The network topology for each model remained as published in their respective papers.

Training Procedure. In my testing, I used a standard procedure similar to the one used by Diehl & Cook (2015): A static image is presented for training to the SNN for 350ms before a 150ms phase which allows neurons to settle down to their original state. The networks were trained with a single pass through a training segment of each dataset (70%) such that there was in total one image per iteration (eg. 50,000 iterations for MNIST). After training, weights were frozen and accuracy was calculated based upon the remaining images (30%) in each dataset.

3.2 Classification Accuracy

In **Table 3**, I report the classification accuracy of the testing set of MNIST, CIFAR10, and CIFAR100 on the various models. As shown, MesNet achieved much improved accuracies across all datasets compared to its SNN counterparts, and shows particular promise with larger scale datasets such as CIFAR100, where it showed a 6.62% improvement over CSNN**.** The IIC ANN model continued to dominate across all datasets with an accuracy of 25.7% accuracy on

CIFAR100, however MesoNet takes an incredible step forward for SNNs making them a viable alternative for fully unsupervised works. These results will be further discussed in Section 4.

Table 2. Classification Accuracies for Experimental Contenders

Model	MNIST	CIFAR10	CIFAR100
Diehl & Cook 2015	90.22%	17.92%	1.20%
ASP	94.31%	36.41%	2.63%
CSNN	92.15%	39.08%	4.54%
IIC (ANN)	99.20%	61.70%	25.70%
MesoNet	96.70%	53.47%	11.16%

3.3 Learning Efficiency

In Figure 5, I report the MNIST testing results of networks across a varying number of training iterations.

Figure 5. Chart describing Accuracy v. Training Iterations for Models on MNIST

Once again, MesoNet achieves much improved results from its SNN counterparts, and most notable is its rapid initial convergence before it levels out. This signifies the effectiveness of the DSP algorithm at selectively incorporating important information, something that will be discussed further in Section 4. It is worth noting that there is still more improvement between Mesonet and IIC at this point, however MesoNet still demonstrates a considerable advance in SNN performance. Additionally, I observed that the ASP algorithm underwent various oscillations on its way to convergence, indicating a possible transcritical bifurcation embedded in its system dynamics that would be worth exploring in a future work.

Furthermore, we can consider the learning efficiency of neurons and synapses. In Figure 6, I report on the number of required neurons and synapses active within each SNN model. This was determined through a robustness study wherein neurons and synapses were randomly deleted until accuracy dropped below 3% of the original accuracy.

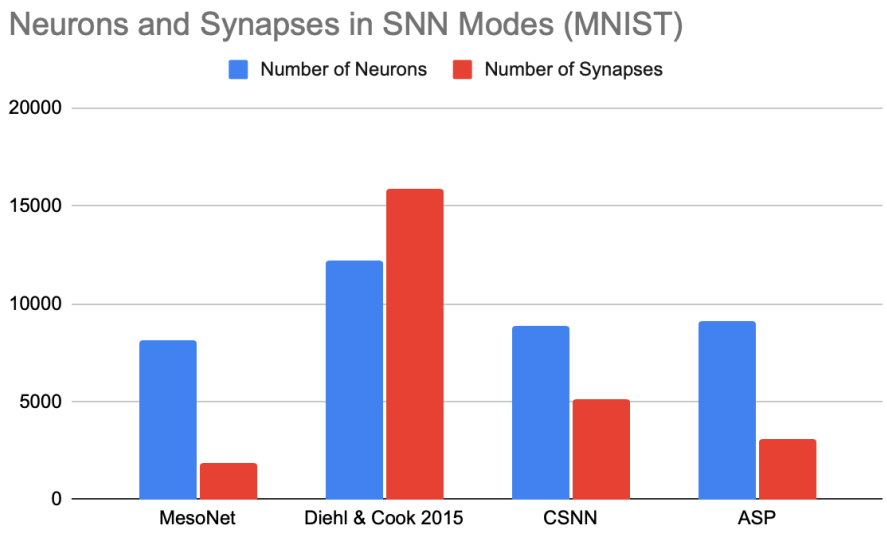

Figure 6. Required Neurons for various models running MNIST.

Generally, compared to models such as CSNN and ASP, MesoNet neuron population is statistically insignificant than others (MesoNet utilizes on average ~8100 neurons on MNIST). However, MesoNet only uses ~1900 average synapses, which is only 61% of the synapses ASP utilizes and 37% of the synapses CSNN employs, while performing better than both across all datasets in accuracy. This allocative efficiency of synapses points once again towards the improved DSP algorithm, which will once again be discussed in Section 4.

3.4 Computational Efficiency

Finally, I report on the number of CMOS operations incurred by each network over 5000 training iterations in Figure 7. This was implemented through a simple operation counter run on a CUDA enabled GPU. Here, there was no statistical trend amongst the various models as they all fell within ~5 million operations of one another which could be a product of simple randomness.

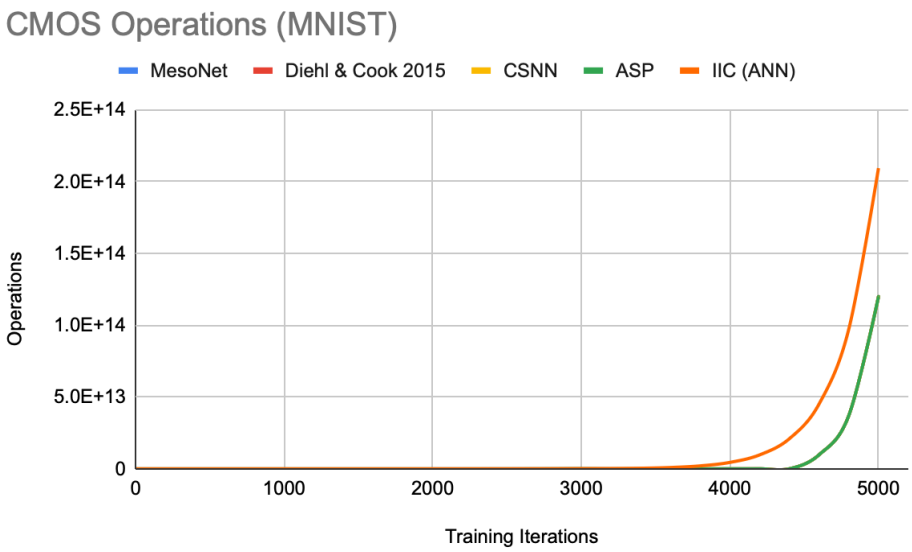

Figure 6. CMOS Operations for SNNs and ANNs over 5000 training iterations of MNIST

The most notable observation here is the radical difference between SNNs including MesoNet and the IIC model. The SNNs averaged to run about 47% of the total operations ran by the ANN,

once again proving the benefits of SNN's sparse, event-based computation. This massive reduction in operations coincides with a massive reduction in required compute and energy to run machine learning models.

4. Discussion and Conclusion

Demonstrated by the experiments in Section 3, the proposed MesoNet significantly improves upon the current state of the art fully-unsupervised SNNs in all scales of tasks, showing great promise specifically in scalability to more complex datasets. In terms of accuracy, MesoNet offers a ~21% improvement above the previous state of the art on CIFAR100 demonstrating the improved capability of the DSP module combined with the inception-like network to generalize on low level features. Specifically, DSP and synaptic flipping greatly improved the learning speed of MesoNet through its greedy updates, allocative efficiency through credit assignment system, and exploitation of bifurcation dynamics. This improvement was reflected once again within the number of synapses MesoNet required while training MNIST, which was only 37% of the synapses required for CSNN**,** a popularized SNN for learning at scale.

Concluding Remarks. In conclusion of this paper, I proposed a theoretical framework to describe a system of STDP synapses as a bifurcation dynamical system to then create a distributed synaptic plasticity model for improved feature generalization. Integrating this with a split-and-merge architecture, I developed MesoNet, a spatio-temporal neural architecture for massively energy efficient unsupervised learning. This network greatly improved the state of the art in generalizing on abstraction opening possibilities for improved scalability. Demonstrated through a variety of tests on popularized benchmarks, MesoNet greatly improved the learning capability and efficiency of SNNs while maintaining a massive energy gap to ANNs, furthering SNNs viability as an energy-efficient alternative to conventional neural networks. Additionally,

the bifurcational analysis conducted provides further insight into the functionality of SNNs to connect theory with practice in the field. Future work could focus on further scaling of the SNNs through incorporation of bio-plausible features such as self-forming recurrencies and inhibitory patterns.

Literature Cited

Yazici, İ., Shayea, I., & Din, J. (2023). A survey of applications of artificial intelligence and machine learning in future mobile networks-enabled systems. *Engineering Science and Technology an International Journal*, *44*, 101455. https://doi.org/10.1016/j.jestch.2023.101455

Pfeiffer, M., & Pfeil, T. (2018). Deep learning with spiking neurons: Opportunities and challenges. *Frontiers in Neuroscience*, *12*. https://doi.org/10.3389/fnins.2018.00774

Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Netw*, 10, 1659–1671. https://doi.org/10.1016/S0893-6080(97)00011-7

Roy, K., Jaiswal, A., & Panda, P. (2019). Towards spike-based machine intelligence with neuromorphic computing. *Nature*, *575*(7784), 607–617. https://doi.org/10.1038/s41586-019-1677-2

Petro, B., Kasabov, N., & Kiss, R. M. (2019). Selection and optimization of temporal spike encoding methods for spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(2), 358–370. https://doi.org/10.1109/tnnls.2019.2906158

Han, B., & Roy, K. (2020). Deep spiking neural network: energy efficiency through time based coding. In *Lecture notes in computer science* (pp. 388–404). https://doi.org/10.1007/978-3-030-58607-2_23

Lu, S., & Sengupta, A. (2024). Deep unsupervised learning using spike-timing-dependent plasticity. *Neuromorphic Computing and Engineering*, *4*(2), 024004. https://doi.org/10.1088/2634-4386/ad3a95

Neftci, E. O., Mostafa, H., & Zenke, F. (2019). Surrogate Gradient Learning in Spiking Neural Networks: Bringing the power of Gradient-Based Optimization to Spiking Neural Networks. *IEEE Signal Processing Magazine*, *36*(6), 51–63. https://doi.org/10.1109/msp.2019.2931595

Kaiser, J., Mostafa, H., & Neftci, E. (2020). Synaptic Plasticity Dynamics for Deep Continuous Local Learning (DECOLLE). *Frontiers in Neuroscience*, *14*. https://doi.org/10.3389/fnins.2020.00424

Gao, H., He, J., Wang, H., Wang, T., Zhong, Z., Yu, J., Wang, Y., Tian, M., & Shi, C. (2023). High-accuracy deep ANN-to-SNN conversion using quantization-aware training framework and calcium-gated bipolar leaky integrate and fire neuron. *Frontiers in Neuroscience*, *17*. https://doi.org/10.3389/fnins.2023.1141701

Bell, C. C., Han, V. Z., Sugawara, Y., & Grant, K. (1997). Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature*, *387*(6630), 278–281. https://doi.org/10.1038/387278a0

Song, S., Miller, K. D., & Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, *3*(9), 919–926. https://doi.org/10.1038/78829

Diehl, P. U., & Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, *9*. https://doi.org/10.3389/fncom.2015.00099

Panda, P., Allred, J. M., Ramanathan, S., & Roy, K. (2017). ASP: Learning to forget with adaptive synaptic plasticity in spiking neural networks. *IEEE Journal on Emerging and*

Selected Topics in Circuits and Systems, *8*(1), 51–64. https://doi.org/10.1109/jetcas.2017.2769684

Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., & Masquelier, T. (2017). STDP-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, *99*, 56–67. https://doi.org/10.1016/j.neunet.2017.12.005

Rathi, N., Chakraborty, I., Kosta, A., Sengupta, A., Ankit, A., Panda, P., & Roy, K. (2022). Exploring Neuromorphic computing Based on spiking Neural Networks: Algorithms to hardware. *ACM Computing Surveys*, *55*(12), 1–49. https://doi.org/10.1145/3571155

Zhang, S.-Q., Zhao-Yu Zhange, & Zhou, Z.-H. (2021). Bifurcation spiking neural network. *The Journal of Machine Learning Research*, *22*(1), 11459–11479. https://doi.org/10.5555/3546258.3546511

Ghantiwala, A. (2023, December 9). Understanding architecture of inception network & applying it to a Real-World dataset. *Medium*. https://gghantiwala.medium.com/understanding-the-architecture-of-the-inception-network-and-applying-it-to-a-real-world-dataset-169874795540

Sussillo, D., & Abbott, L. (2009). Generating Coherent Patterns of Activity from Chaotic Neural Networks. *Neuron*, *63*(4), 544–557. https://doi.org/10.1016/j.neuron.2009.07.018

Dauphin, Y., Pascanu, R., Gulcechre, C., Cho, K., Ganguli, S., & Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *ACM Computing Surveys*, *2*, 2933–2941. https://doi.org/10.5555/2969033.2969154

Onuki, A. (2002). *Phase Transition Dynamics*. https://doi.org/10.1017/cbo9780511534874

Y. A. Kuznetsov. Elements of Applied Bifurcation Theory. Springer, 2013.

S. H. Strogatz. Nonlinear Dynamics and Chaos. CRC Press, 2018.

Beer, R. D. (1995). A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence*, *72*(1–2), 173–215. https://doi.org/10.1016/0004-3702(94)00005-l

Gerstner, W. (1995). Time structure of the activity in neural network models. *Physical Review. E, Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, *51*(1), 738–758. https://doi.org/10.1103/physreve.51.738

Dumont, G., Payeur, A., & Longtin, A. (2017). A stochastic-field description of finite-size spiking neural networks. *PLoS Computational Biology*, *13*(8), e1005691. https://doi.org/10.1371/journal.pcbi.1005691

Quiroga, R. Q., Reddy, L., Kreiman, G., Koch, C., & Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature*, *435*(7045), 1102–1107. https://doi.org/10.1038/nature03687

Anumula, J., Neil, D., Delbruck, T., & Liu, S. (2018). Feature representations for neuromorphic audio spike streams. *Frontiers in Neuroscience*, *12*. https://doi.org/10.3389/fnins.2018.00023

Jacobian | Definition of Jacobian in English by Oxford Dictionaries. (n.d.). Oxford Dictionaries | English. https://web.archive.org/web/20171201043633/https://en.oxforddictionaries.com/definition/jacobian

Bengio, Y., & Delalleau, O. (2011). On the Expressive Power of Deep Architectures. In *Lecture notes in computer science* (pp. 18–36). https://doi.org/10.1007/978-3-642-24412-4_3

Masquelier, T., & Thorpe, S. J. (2007). Unsupervised Learning of Visual Features through Spike Timing Dependent Plasticity. *PLoS Computational Biology*, *3*(2), e31. https://doi.org/10.1371/journal.pcbi.0030031

Zhang, S.-Q., Chen, J.-Y., Wu, J.-H., Zhang, G., Xiong, H., Gu, B., Zhi-Hua Zhou, (2024). On the Intrinsic Structures of Spiking Neural Networks. *Journal of Machine Learning Research*. https://www.jmlr.org/papers/volume25/23-1526/23-1526.pdf

Saxe, A. M., McClelland, J. L., & Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1312.6120

Mondal, B., Ghosh, U., Rahman, M. S., Saha, P., & Sarkar, S. (2021). Studies of different types of bifurcations analyses of an imprecise two species food chain model with fear effect and non-linear harvesting. *Mathematics and Computers in Simulation*, *192*, 111–135. https://doi.org/10.1016/j.matcom.2021.08.019

M. W. Hirsch, S. Smale, and R. L. Devaney. Differential Equations, Dynamical Systems, and An Introduction to Chaos. Academic Press, 2012.

Guo, Y., Chen, Y., Zhang, L., Liu, X., Wang, Y., Huang, X., & Ma, Z. (2024). IM-Loss: information maximization loss for spiking neural networks. *ACM Computing Surveys*, *12*, 156–166. https://doi.org/10.5555/3600270.3600282

Vu, T. L., & Turitsyn, K. (2015). Lyapunov functions family approach to transient stability assessment. *IEEE Transactions on Power Systems*, *31*(2), 1269–1277. https://doi.org/10.1109/tpwrs.2015.2425885

Alreja, A., Nemenman, I., & Rozell, C. J. (2022). Constrained brain volume in an efficient coding model explains the fraction of excitatory and inhibitory neurons in sensory cortices. *PLoS Computational Biology*, *18*(1), e1009642. https://doi.org/10.1371/journal.pcbi.1009642

Supervised STDP Learning with Weight Decay for Spiking Neural Networks. (2023, November 10). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/10455911

Araki, H., & Hattori, M. (2023, November 10). *Supervised STDP Learning with Weight Decay for Spiking Neural Networks*. IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/10455911

Luo, L. (2021). Architectures of neuronal circuits. *Science*, *373*(6559). https://doi.org/10.1126/science.abg7285

Aramendia, A. I. (2023, November 2). What is Dijkstra's Algorithm, How does it Work? | Medium. *Medium*. https://medium.com/@alejandro.itoaramendia/a-guide-to-dijkstras-algorithm-all-you-need-99635dcd6d94

Meng, M., Yang, X., Bi, L., Kim, J., Xiao, S., & Yu, Z. (2021). High-parallelism inception-like spiking neural networks for unsupervised feature learning. *Neurocomputing*, *441*, 92–104. https://doi.org/10.1016/j.neucom.2021.02.027

Szegedy, C., Liu, N. W., Jia, N. Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *IEEE Conference on*

Computer Vision and Pattern Recognition, 1–9. https://doi.org/10.1109/cvpr.2015.7298594

Abderrahmane, N., Lemaire, E., & Miramond, B. (2019). Design space exploration of hardware spiking neurons for embedded artificial intelligence. *Neural Networks*, *121*, 366–386. https://doi.org/10.1016/j.neunet.2019.09.024

Lai, H. C., & Jan, L. Y. (2006). The distribution and targeting of neuronal voltage-gated ion channels. *Nature Reviews. Neuroscience*, *7*(7), 548–562. https://doi.org/10.1038/nrn1938

Stimberg, M., Brette, R., & Goodman, D. F. (2019). Brian 2, an intuitive and efficient neural simulator. *eLife*, *8*. https://doi.org/10.7554/elife.47314

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proc. IEEE 86, 2278–2324. https://doi.org/10.1109/5.726791

Franky. (2022, September 2). Once Upon a Time in CIFAR-10 - franky - Medium. *Medium*. https://franky07724-57962.medium.com/once-upon-a-time-in-cifar-10-c26bb056b4ce

GeeksforGeeks. (2024, May 22). *CIFAR 100 Dataset*. GeeksforGeeks. https://www.geeksforgeeks.org/cifar-100-dataset/

Pillow, J. W., Paninski, L., Uzzell, V. J., Simoncelli, E. P., & Chichilnisky, E. J. (2005). Prediction and Decoding of Retinal Ganglion Cell Responses with a Probabilistic Spiking Model. *Journal of Neuroscience*, *25*(47), 11003–11013. https://doi.org/10.1523/jneurosci.3305-05.2005

Ji, X., Vedaldi, A., & Henriques, J.F. (2018). Invariant Information Clustering for Unsupervised Image Classification and Segmentation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 9864-9873.