HETEROGENEOUS NEURONAL AND SYNAPTIC DYNAMICS FOR SPIKE-EFFICIENT UNSUPERVISED LEARNING: THEORY AND DESIGN PRINCIPLES

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper shows that the heterogeneity in neuronal and synaptic dynamics reduces the spiking activity of a Recurrent Spiking Neural Network (RSNN) while improving prediction performance, enabling spike-efficient (unsupervised) learning. We analytically show that the diversity in the integration/relaxation dynamics of neurons improves an RSNN's ability to learn more distinct input patterns (higher memory capacity), leading to improved classification and prediction performance. We further prove that heterogeneous Spike-Timing-Dependent-Plasticity (STDP) dynamics of synapses reduce spiking activity but preserve memory capacity. The analytical results motivate **h**eterogeneous RSNN (HRSNN) design using Bayesian optimization to determine heterogeneity in neurons and synapses to improve $\mathcal{E}$, defined as the ratio of spiking activity and memory capacity. The empirical results on time series classification and prediction tasks show optimized HRSNN increases performance and reduces spiking activity compared to a ho**m**ogeneous RSNN (MRSNN).

1 INTRODUCTION

Spiking neural networks (SNNs) (Ponulak & Kasinski, 2011) use unsupervised bio-inspired neurons and synaptic connections, trainable with either biological learning rules such as spike-timing-dependent plasticity (STDP) (Gerstner & Kistler, 2002) or supervised statistical learning algorithms such as surrogate gradient (Neftci et al., 2019). Empirical results on standard SNNs also show good performance for various tasks including spatiotemporal data classification, (Lee et al., 2017; Khoei et al., 2020; She et al., 2021), sequence-to-sequence mapping (Zhang & Li, 2020), object detection (Chakraborty et al., 2021; Kim et al., 2020), and universal function approximation (Gelenbe et al., 1999; Iannella & Back, 2001; Torikai et al., 2008; Farsa et al., 2015). An important motivation for the application of SNN in machine learning (ML) is the sparsity in the firing (activation) of the neurons, which reduces energy dissipation during inference (Kim et al., 2020; Wu et al., 2019; Srinivasan & Roy, 2019). Many prior works have empirically shown that SNN has lower firing activity than artificial neural networks and, hence, can improve energy efficiency (Kim et al., 2020; Wu et al., 2019; Srinivasan & Roy, 2019). However, there are very few analytical studies on *how to reduce the spiking activity of an SNN while maintaining its learning performance*. Understanding and optimizing the relations between spiking activity and performance will be key to designing energy-efficient SNNs for complex ML tasks.

In this paper, we derive analytical results and present design principles from optimizing the spiking activity of a recurrent SNN (RSNN) while maintaining prediction performance. Most SNN research in ML considers a simplified network model with a homogeneous population of neurons and synapses (ho**m**ogeneous RSNN (MRSNN)) where all neurons have uniform integration/relaxation dynamics, and all synapses use the same long-term potentiation (LTP) and long-term depression (LTD) dynamics in STDP learning rules. On the contrary, neurobiological studies have shown that a brain has a wide variety of neurons and synapses with varying firing and plasticity dynamics, respectively (Destexhe & Marder, 2004; Marder & Taylor, 2011; Gouwens et al., 2019; Hansel et al., 1995; Prescott et al., 2008; Machens et al., 2010). We show that ***optimizing neuronal and synaptic heterogeneity will be key to simultaneously reducing spiking activity while improving performance***.

We define the spike efficiency $\mathcal{E}$ of an RSNN as the ratio of its memory capacity $\mathcal{C}$ and average spiking activity $\tilde{S}$. Given a fixed number of neurons and synapses, a higher $\mathcal{C}$ implies a network can learn more patterns and hence, perform better in classification or prediction tasks (Aceituno et al., 2020; Goldmann et al., 2020; Vershynin, 2020); a lower spiking rate implies that a network is less active, and hence, will consume less energy while making inferences (Sorbaro et al., 2020; Rathi et al., 2021). We analytically show that a **H**eterogeneous **R**ecurrent SNN (HRSNN) model leads to a more spike-efficient learning architecture by reducing spiking activity while improving $\mathcal{C}$ (i.e., performance) of the learning models. In particular, we make the following contributions to the theoretical understanding of an HRSNN.

- We analytically show that for a finite number of neurons, increasing the heterogeneity among the neuronal dynamics increases $\mathcal{C}$ of HRSNN.
- We prove that heterogeneity in the synaptic dynamics reduces the spiking activity of neurons while maintaining $\mathcal{C}$.
- We connect preceding results to prove that simultaneously using heterogeneity in neurons and synapses, as in an HRSNN, improves the spike-efficiency of a network.

The theoretical results are used to develop an HRSNN architecture where a modified Bayesian Optimization (BO) is used to determine the optimal distribution of neuron and synaptic parameters to maximize $\mathcal{E}$. We empirically characterize HRSNN considering the tasks of (a) classifying time series (Spoken Heidelberg Digits (SHD)) and (b) predicting the evolution of a dynamical system (a modified chaotic Lorenz system) . HRSNN exhibits a better performance (higher classification accuracy and lower NRMSE loss) with a lesser average spike count $\tilde{S}$ than MRSNN.

1.1 Related Works

Inspired by the biological observations, recent empirical studies showed potential for improving SNN performance with heterogeneous neuron dynamics(Perez-Nieves et al., 2021; She et al., 2022; 2021). However, there is a lack of theoretical understanding of why heterogeneity improves SNN performance, which is critical for optimizing SNNs for complex tasks. She et al. (2022) have analytically studied the universal sequence approximation capabilities of a feed-forward network of neurons with varying dynamics. However, they did not consider heterogeneity in plasticity dynamics, and the results are applicable only for a feed-forward SNN and do not extend to recurrent SNNs (RSNN). The recurrence is not only a fundamental component of a biological brain (Soures & Kudithipudi, 2019), but as a machine learning (ML) model, RSNN also shows good performance in modeling spatiotemporal and nonlinear dynamics (Pyle & Rosenbaum, 2017; Gilra & Gerstner, 2017). Hence, it is critical to understand whether heterogeneity can improve learning in an RSNN. To the best of our knowledge, this is the first work that analytically studies the impact of heterogeneity in synaptic and neuronal dynamics in an RSNN. This work shows that only using neuronal heterogeneity improves performance and does not impact spiking activity. In fact, the number of spikes required for the computation increases exponentially with the number of neurons. Therefore, simultaneously analyzing and optimizing neuronal and synaptic heterogeneity, as demonstrated in this work, is critical to design an energy-efficient recurrent SNN.

2 Preliminaries and Definitions

Table 1 summarizes the key notations used in this paper. We now define the key terms used in the paper. Figure 1 shows the general structure of the HRSNN model with heterogeneity in both the LIF neurons and the STDP dynamics.

2.1 Heterogeneous LIF Neurons

We use the Leaky Integrate and Fire (LIF) neuron model in all our simulations. In this model, the membrane potential of the i-th neuron $u_i(t)$ varies over time as:

$$\tau_m \frac{dv_i(t)}{dt} = -\left(v_i(t) - v_{rest}\right) + I_i(t) \tag{1}$$

Table 1: Notations Table

Notation	Meaning	Notation	Meaning	Notation	Meaning	Notation	Meaning
HRSNN	Heterogeneous Recurrent Spiking Neural Network	$S_i(t)$	Spike train from neuron i at time t	z_j^t	Spike indicator Function	$\mathbf{r}(t)$	States of RSNN
MRSNN	Homogeneous Recurrent Spiking Neural Network	Δw	Synaptic Weight Update	v_{th}	Threshold voltage	$\mathbf{\Sigma}$	Covariance Matrix of $\mathbf{r}(t)$
τ_m	Membrane Time Constant	$\tau_\pm$	Synaptic Time Constants	$\mathcal{C}$	Memory Capacity	x_i	Model Input
$v_i(t)$	Membrane Potential	$\eta_\pm$	Scaling Functions	y_τ	Model Output with delay τ	$\mathcal{E}$	Spike Efficiency
v_{rest}	Resting potential	α	Constant Decay Factor	$\mathcal{R}$	Recurrent Layer in RSNN	S_R, S_M	Total Spike Count of HRSNN (R) , MRSNN (M)
$\tilde{S}$	Avg. Spike Count per neuron	W_{ji}	Synapse weight from neuron i to neuron j	$\mathcal{H}$	Heterogeneity in Neuronal Parameters	f_R^{ISI}, f_M^{ISI}	Spike frequency of HRSNN (R), MRSNN (M)

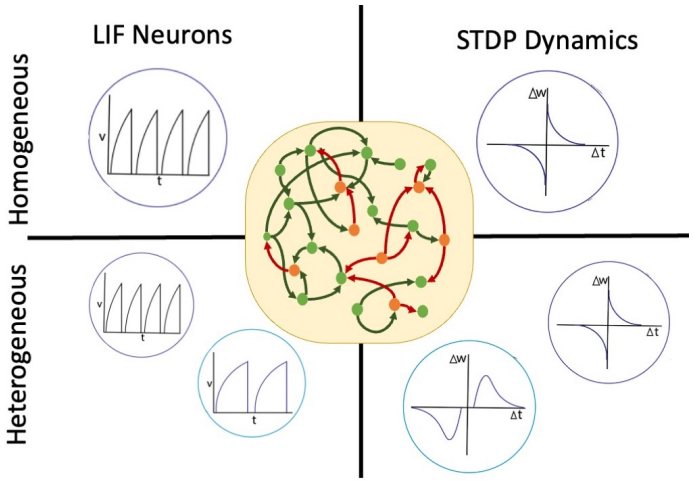

Figure 1: Concept of HRSNN with variable Neuronal and Synaptic Dynamics

where τ_{m} is the membrane time constant, v_{rest} is the resting potential and I_{i} is the input current. When the membrane potential reaches the threshold value v_{th} a spike is emitted, $v_i(t)$ resets to the reset potential v_{r} and then enters a refractory period where the neuron cannot spike. Spikes emitted by the j th neuron at a finite set of times $\{t_j\}$ can be formalised as a spike train $S_i(t) = \sum \delta\left(t - t_i\right)$. To implement the LIF model, we discretize time into multiples of a small-time step Δt so that spikes can only happen at multiples of Δt. (Cramer et al., 2020; Perez-Nieves et al., 2021) Thus, we can approximately solve Eq. 1 as

$$v_i(t+\Delta t) = v_i[t+1] = \beta\left(v_i[t] - v_0\right) + v_0 + (1-\beta) I_i[t] - \left(v_{th} - v_r\right) S_i[t] \quad \text{s.t. } \beta = \exp^{(-\Delta t/\tau_{\mathrm{m}})} \quad (2)$$

Let the recurrent layer of an RSNN be $\mathcal{R}$. We incorporate heterogeneity in the LIF neurons by using different membrane time constants $\tau_{m,i}$ and threshold voltages $v_{th,i}$ for each LIF neuron i in $\mathcal{R}$. This gives a distribution of time constants and threshold voltages of the LIF neurons in $\mathcal{R}$.

2.2 Heterogeneous STDP:

The STDP rule for updating a synaptic weight (Δw) is defined by Pool & Mato (2011):

$$\Delta w(\Delta t) = \begin{cases} A_+(w) e^{-\frac{|\Delta t|}{\tau_+}} \text{ if } \Delta t \geq 0 \\ -A_-(w) e^{-\frac{|\Delta t|}{\tau_-}} \text{ if } \Delta t < 0 \end{cases} \qquad s.t., A_+(w) = \eta_+\left(w_{\max} - w\right), A_-(w) = \eta_-\left(w - w_{\min}\right) \quad (3)$$

where $\Delta t = t_{\text{post}} - t_{\text{pre}}$ is the time difference between the postsynaptic spike and the presynaptic one, with synaptic time-constant $\tau_\pm$. In heterogeneous STDP, we use an ensemble of values taken from a distribution for $\tau_\pm$ and the scaling functions $\eta_\pm$.

2.3 HETEROGENEITY:

We define heterogeneity as a measure of the variability of the hyperparameters in an RSNN that gives rise to an ensemble of neuronal dynamics.

Entropy is used to measure population diversity. Assuming that the random variable for the hyperparameters X follows a multivariate Gaussian Distribution ($X \sim \mathcal{N}(\mu, \Sigma)$), then the differential entropy of x on the multivariate Gaussian distribution, is $\mathcal{H}(x) = \frac{n}{2}\ln(2\pi) + \frac{1}{2}\ln|\Sigma| + \frac{n}{2}$. Now, if we take any density function $q(\mathrm{x})$ that satisfies $\int q(\mathrm{x}) x_i x_j d\mathrm{x} = \Sigma_{ij}$ and $p = \mathcal{N}(0, \Sigma)$, then $\mathcal{H}(q) \leq \mathcal{H}(p)$. (Proof in Suppl. Sec. A) The Gaussian distribution maximizes the entropy for a given covariance. Hence, the log-determinant of covariance matrix bounds entropy. Thus, for the rest of the paper, we use the determinant of the covariance matrix to measure the heterogeneity of the network.

2.4 MEMORY CAPACITY:

We define the memory capacity $\mathcal{C}$ as a measure for the ability of the model to store and recall previous inputs fed into the network (Jaeger, 2001; Jaeger et al., 2001).
In this paper, we use $\mathcal{C}$ as a measure of the performance of the model, which is based on the network's ability to retrieve past information (for various delays) from the reservoir using the linear combinations of reservoir unit activations observed at the output. Intuitively, HRSNN can be interpreted as a set of coupled filters that extract features from the input signal, and the final readout selects the right combination of those features for classification or prediction. First, the τ-delay $\mathcal{C}$ measures the performance of the RC for the task of reconstructing the delayed version of model input $x(t)$ at delay τ (i.e., $x(t-\tau)$) and is defined as follows:

$$\mathcal{C} = \sum_{\tau=1}^{\infty} \mathcal{C}(\tau) = \sum_{\tau=1}^{\infty} \frac{\mathrm{Cov}^2\left(x(t-\tau), y_\tau(t)\right)}{\mathrm{Var}(x(t))\,\mathrm{Var}\left(y_\tau(t)\right)}, \tau \in \mathbb{N}, \tag{4}$$

where $\mathrm{Cov}(\cdot)$ and $\mathrm{Var}(\cdot)$ denote the covariance function and variance function, respectively. The $y_\tau(t)$ is the model output in this reconstruction task. $\mathcal{C}$ measures the ability of RSNN to reconstruct precisely the past information of the model input. Thus, increasing $\mathcal{C}$ indicates the network is capable of learning a greater number of past input patterns which in turn, helps in increasing the performance of the model.

Spike-Efficiency: *We define the spike-efficiency ($\mathcal{E}$) of an RSNN model as the ratio of $\mathcal{C}$ to the average total spike count per neuron $\tilde{S}$.*

$\mathcal{E}$ is an analytical measure used to compare how $\mathcal{C}$ and hence the model's performance is improved with per unit spike activity in the model. Ideally, we want to design a system with high $\mathcal{C}$ using fewer spikes. Hence we define $\mathcal{E}$ as the ratio of the memory capacity using $N_\mathcal{R}$ neurons $\mathcal{C}(N_\mathcal{R})$ to the average number of spike activations per neuron ($\tilde{S}$) and is given as:

$$\mathcal{E} = 1000 \times \frac{\mathcal{C}(N_\mathcal{R})}{\frac{\sum_{i=1}^{N_\mathcal{R}} S_i}{N_\mathcal{R}}}, \qquad S_i = \int_0^T s_i(t)dt \approx N_{\text{post}} \frac{T}{\int_{t_{ref}}^{\infty} t f_i^{ISI} dt} \tag{5}$$

where N_{post} is the number of postsynaptic neurons, f_i^{ISI} is the inter-spike interval spike frequency for neuron i, and T is the total time. The factor 1000 is multiplied for better interpretability.

3 HETEROGENEOUS RSNN: ANALYTICAL RESULTS

In this section, we show three key analytical results. First, we show that heterogeneity in neuronal dynamics increases memory capacity. The intuition is this type of heterogeneity increases the number of principal components captured from the input space, forming a better representation of the input and improving the model's performance, thus improving $\mathcal{C}$. Second, we will show that heterogeneity in the STDP dynamics reduces spike activation without impacting $\mathcal{C}$. The intuition behind this comes from the fact that it gives better orthogonalization among the recurrent network states and helps achieve a more efficient representation of the input space. This simultaneously lowers the higher-order

correlation in spike trains making the model more spike-efficient since the higher-order correlation progressively decreases the information available through neural population (Montani et al., 2009; Abbott & Dayan, 1999). Finally, we show that incorporating heterogeneity in both neuron and STDP dynamics allows us to increase the ratio of $\mathcal{C}$ to spike activity, which is defined as $\mathcal{E}$. A higher $\mathcal{E}$ implies the ability to improve performance while reducing spike counts.

3.1 Memory Capacity

The performance of an RSNN depends on its ability to retain the memory of previous inputs. To quantify the relationship between the recurrent layer dynamics and $\mathcal{C}$, we note that extracting information from the recurrent layer is made using a combination of the neuronal states. Hence, more linearly independent neurons would offer more variable states and, thus, more extended memory.

Lemma 3.1.1: *The state of the neuron can be written as follows:*

$$r_i(t) = \sum_{k=0}^{N_R} \sum_{n=1}^{N_R} \lambda_n^k \left\langle v_n^{-1}, \mathbf{w}^{\text{in}} \right\rangle (v_n)_i \, x(t-k) \tag{6}$$

where $\mathbf{v}_n, \mathbf{v}_n^{-1} \in \mathbf{V}$ are, respectively, the left and right eigenvectors of $\mathbf{W}$, and $\lambda_n^k \in \lambda$ belongs to the diagonal matrix containing the eigenvalues of $\mathbf{W}$; $\mathbf{a}_i = [a_{i,0}, a_{i,1}, \ldots]$ represents the coefficients that the previous inputs $\mathbf{x}_t = [x(t), x(t-1), \ldots]$ have on $r_i(t)$.

Short Proof: *(See Suppl. Sec. B for full proof)* As discussed by Aceituno et al. (2020), the state of the neuron can be represented as $\mathbf{r}(t) = \mathbf{W}\mathbf{r}(t-1) + \mathbf{w}^{\text{in}}\, x(t)$, where $\mathbf{w}^{\text{in}}$ are the input weights. We can simplify this using the coefficients of the previous inputs and plug this term into the covariance between two neurons. Hence, we write the coefficients of the inputs $\mathbf{a}$ as a function of the eigenvalues of $\mathbf{W}$ to get

$$\mathbf{r}(t) = \sum_{k=0}^{\infty} \mathbf{W}^k \mathbf{w}^{\text{in}} x(t-k) = \sum_{k=0}^{\infty} \left(\mathbf{V}\mathbf{\Lambda}^k\mathbf{V}^{-1}\right) \mathbf{w}^{\text{in}} x(t-k) \Rightarrow r_i(t) = \sum_{k=0}^{N_R} \sum_{n=1}^{N_R} \lambda_n^k \left\langle v_n^{-1}, \mathbf{w}^{\text{in}} \right\rangle (v_n)_i \, x(t-k) \tag{7}$$

. ■

Theorem 1: *For HRSNN, heterogeneity in the neuronal parameters $\mathcal{H}$ varies inversely to the correlation among the neuronal states measured as $\sum_{n=1}^{N_\mathcal{R}} \sum_{m=1}^{N_\mathcal{R}} \text{Cov}^2\left(x_n(t), x_m(t)\right)$ which in turn varies inversely with $\mathcal{C}$. Hence, the $\mathcal{C}$ increases as the heterogeneity in the parameters increases.*

Intuitive Proof: *(See Suppl. Sec. B for full proof)* Aceituno et al. (2020) showed that the $\mathcal{C}$ increases when the variance along the projections of the input into the recurrent layer are uniformly distributed. We show that this can be achieved efficiently by using heterogeneity in the LIF dynamics. More formally, let us express the projection in terms of the state space of the recurrent layer. We show that the raw variance in the neuronal states $\mathcal{J}$ can be written as $\mathcal{J} = \dfrac{\sum_{n=1}^{N_\mathcal{R}} \lambda_n^2(\mathbf{\Sigma})}{\left(\sum_{n=1}^{N_\mathcal{R}} \lambda_n(\mathbf{\Sigma})\right)^2}$ where $\lambda_n(\mathbf{\Sigma})$ is the nth eigenvalue of $\mathbf{\Sigma}$. We further show that with higher $\mathcal{H}$, the magnitude of the eigenvalues of $\mathbf{W}$ decreases and hence leads to a higher $\mathcal{J}$. Now, we project the inputs into orthogonal directions of the network state space and model the system as $\mathbf{r}(t) = \sum_{\tau=1}^{\infty} \mathbf{a}_\tau x(t-\tau) + \varepsilon_r(t)$ where the vectors $\mathbf{a}_\tau \in \mathbb{R}^N$ are correspond to the linearly extractable effect of $x(t-\tau)$ onto $\mathbf{r}(t)$ and $\varepsilon_r(t)$ is the nonlinear contribution of all the inputs onto the state of $\mathbf{r}(t)$. First, we show that $\mathcal{C}$ increases when the variance along the projections of the input into the recurrent layer is more uniform. Intuitively, the variances at directions $\mathbf{a}_\tau$ must fit into the variances of the state space, and since the projections are orthogonal, the variances must be along orthogonal directions. Hence, we show that increasing the correlation among the neuronal states increases the variance of the eigenvalues, which would decrease our memory bound $\mathcal{C}^*$. We show that heterogeneity is inversely proportional to $\sum_{n=1}^{N_\mathcal{R}} \text{Cov}^2\left(x_n(t), x_m(t)\right)$. We see that increasing the correlations between neuronal states decreases the heterogeneity of the eigenvalues, which reduces $\mathcal{C}$. We show that the variance in the neuronal states is bounded by the determinant of the covariance between the states; hence, covariance increases when the neurons become correlated. As $\mathcal{H}$ increases, neuronal correlation decreases. Aceituno et al. (2020) proved

that the neuronal state correlation is inversely related to $\mathcal{C}$. Hence, we claim that as $\mathcal{H}$ increases, $\mathcal{C}$ also increases. ■

3.2 Spiking Efficiency

Lemma 3.2.1: *Heterogeneity in LTP/LTD dynamics of STDP in HRSNN promotes sparsity in the neural firing as it leads to a lower variance for the distribution of membrane voltages in the neuronal population than MRSNN keeping the mean constant which can be represented as follows:*

$$\Delta\text{Var}[V_{\text{Heterogeneous}}] < \Delta\text{Var}[V_{\text{Homogeneous}}] \tag{8}$$

where V_X denotes the random variable for the membrane voltages for the LIF neurons of type X, X = Homogeneous or Heterogeneous.

Short Proof: (*See Suppl. Sec. B for full proof*) The Lemma indicates that the variance of HRSNN is less than that of RSNN with homogeneous STDP. Hence, the probability of reaching a value higher than some threshold is also lower. Thus, the HRSNN population with heterogeneous STDP dynamics fires fewer spikes. This can also be proved using the fact that $\Delta\text{Var}[V_R] < \Delta\text{Var}[V_M] \Rightarrow$ average rate of incoming spike for RSNN with heterogeneous STDP is less than that of HRSNN with homogeneous STDP dynamics.

Theorem 2: *For a given number of neurons $N_{\mathcal{R}}$, the spike efficiency of the model $\mathcal{E} = \frac{\mathcal{C}(N_{\mathcal{R}})}{\tilde{S}}$ for HRSNN ($\mathcal{E}_R$) is greater than MRSNN ($\mathcal{E}_M$) i.e., $\mathcal{E}_R \geq \mathcal{E}_M$*

Short Proof: *(See Suppl. Sec. B for full proof)* First, using Lemma 3.2.1, we show that the number of spikes decreases when we use heterogeneity in the LTP/LTD Dynamics. Hence, we compare the efficiencies of HRSNN with that of MRSNN as follows:

$$\frac{\mathcal{E}_R}{\mathcal{E}_M} = \frac{M_R(N_{\mathcal{R}}) \times \tilde{S_M}}{\tilde{S_R} \times M_M(N_{\mathcal{R}})} = \frac{\sum_{\tau=1}^{N_{\mathcal{R}}} \frac{\text{Cov}^2(x(t-\tau), \mathbf{a}_\tau^R \mathbf{r}_R(t))}{\text{Var}(\mathbf{a}_\tau^R \mathbf{r}_R(t))} \times \int_{t_{ref}}^{\infty} t f_R^{ISI} dt}{\sum_{\tau=1}^{N_{\mathcal{R}}} \frac{\text{Cov}^2(x(t-\tau), \mathbf{a}_\tau^M \mathbf{r}_M(t))}{\text{Var}(\mathbf{a}_\tau^M \mathbf{r}_M(t))} \times \int_{t_{ref}}^{\infty} t f_M^{ISI} dt} \tag{9}$$

Since $S_R \leq S_M$ and also,the covariance increases when the neurons become correlated, and as neuronal correlation decreases, $\mathcal{H}_X$ increases (Theorem 1), we see that $\mathcal{E}_R/\mathcal{E}_M \geq 1 \Rightarrow \mathcal{E}_R \geq \mathcal{E}_M$ ■

3.3 Optimal Heterogeneity using Bayesian Optimization for Distributions

To get optimal heterogeneity in the neuron and STDP dynamics, we use a modified Bayesian Optimization (BO) technique. However, using BO for high-dimensional problems remains a significant challenge. In our case, optimizing HRSNN model parameters for 5000 neurons requires the optimization of two parameters per neuron and four parameters per STDP synapse, where standard BO fails to converge to an optimal solution. However, the parameters to be optimized are correlated and can be drawn from a probability distribution as shown by Perez-Nieves et al. (2021). Thus, we design a modified BO to estimate parameter distributions instead of individual parameters for the LIF neurons and the STDP synapses, for which we modify the BO's surrogate model and acquisition function. This makes our modified BO highly scalable over all the variables (dimensions) used. The loss for the surrogate model's update is calculated using the Wasserstein distance between the parameter distributions. We use the modified Matern function on the Wasserstein metric space as a kernel function for the BO problem. The detailed BO methods are discussed in Suppl. Sec. A. BO uses a Gaussian process to model the distribution of an objective function and an acquisition function to decide points to evaluate. For data points $x \in X$ and the corresponding output $y \in Y$, an SNN with network structure $\mathcal{V}$ and neuron parameters $\mathcal{W}$ acts as a function $f_{\mathcal{V},\mathcal{W}}(x)$ that maps input data x to y. The optimization problem can be defined as: $\min_{\mathcal{V},\mathcal{W}} \sum_{x \in X, y \in Y} \mathcal{L}\left(y, f_{\mathcal{V},\mathcal{W}}(x)\right)$ where $\mathcal{V}$ is the set of hyperparameters of the neurons in $\mathcal{R}$ and $\mathcal{W}$ is the multi-variate distribution constituting the distributions of: (i) the membrane time constants τ_{m-E}, τ_{m-I} of LIF neurons, (ii) the scaling function constants (A_+, A_-) and (iii) the decay time constants τ_+, τ_- for the STDP learning rule in $\mathcal{S}_{\mathcal{R}\mathcal{R}}$.

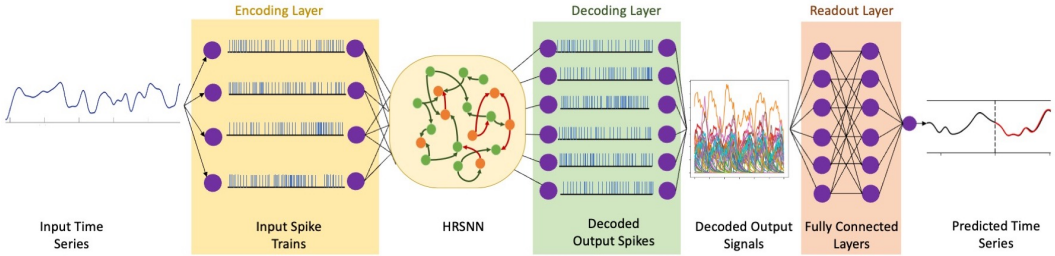

Figure 2: Block Diagram showing the methodology using HRSNN for prediction

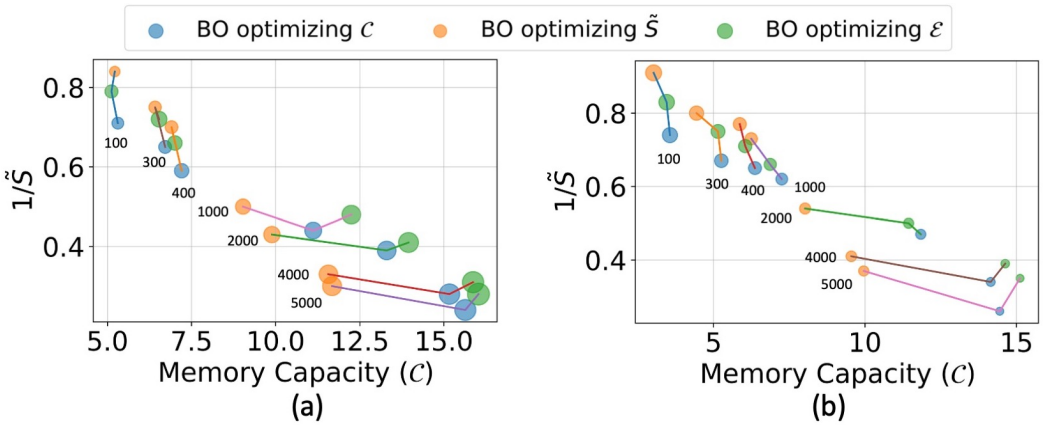

Figure 3: Figure showing the results of ablation studies of BO for the following three cases: (a) for prediction problem - the radius indicates the normalized NRMSE loss with a smaller radius indicating lower (better) NRMSE (b) for classification problem - the radius indicates the normalized accuracy with larger radius indicating higher (better) accuracy. The numbers represent the number of neurons used in each model, and the line joins the three corresponding models with the same model size. The detailed results are shown in Suppl. Sec. A

4 Experimental Results

4.1 Model and Architecture

We empirically verify our analytical results using HRSNN for classification and prediction tasks. Fig. 2 shows the overall architecture of the prediction model. The time-series data is encoded to a series of spike trains using a rate-encoding methodology. This high-dimensional spike train acts as the input to HRSNN. The output spike trains from HRSNN act as the input to a decoder and a readout layer that finally gives the prediction results. For the classification task, we use a similar method. However, we do not use the decoding layer for the signal but directly feed the output spike signals from HRSNN into the fully connected layer.

4.2 Datasets:

Classification: We use the Spoken Heidelberg Digits (SHD) spiking dataset to benchmark the HRSNN model with other standard spiking neural networks (Cramer et al., 2020).
Prediction: We use a multiscale Lorenz 96 system (Lorenz, 1996) which is a set of coupled nonlinear ODEs and an extension of Lorenz's original model for multiscale chaotic variability of weather and climate systems which we use as a testbed for the prediction capabilities of the HRSNN model (Thornes et al., 2017). Further details on both the datasets is provided in Suppl. Sec. A.

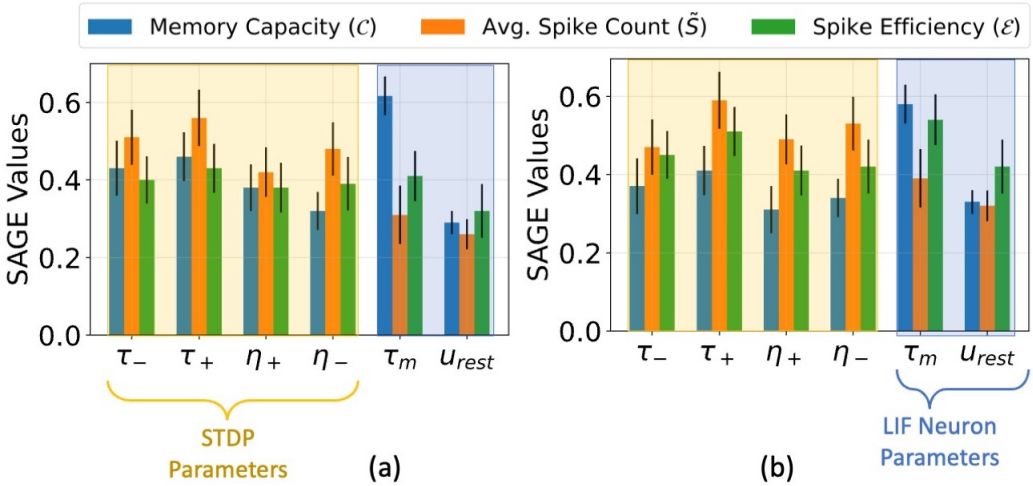

Figure 4: Bar chart showing the global importance of different heterogeneous parameters using HRSNN on the dataset. The experiments were repeated five times with different parameters from the same distribution (a) Classification (b) Prediction

4.3 Bayesian Optimization Ablation Studies

First, we perform an ablation study of BO for the following three cases: (i) Using Memory Capacity $\mathcal{C}$ as the objective function (ii) Using Average Spike Count $\tilde{S}$ as the objective function and (iii) Using $\mathcal{E}$ as the objective function. We optimize both LIF neuron parameter distribution and STDP dynamics distributions for each. We plot $\mathcal{C}$, $\tilde{S}$, the empirical spike efficiency $\hat{\mathcal{E}}$, and the observed RMSE of the model obtained from BO with different numbers of neurons. The results for classification and prediction problems are shown in Fig. 3(a) and (b), respectively. Ideally, we want to design networks with high $\mathcal{C}$ and low spike count, i.e., models in the upper right corner of the graph. The observed results show that BO using $\mathcal{E}$ as the objective gives the best accuracy with the fewest spikes. Thus, we can say that this model has learned a better-orthogonalized subspace representation, leading to better encoding of the input space with fewer spikes. Hence, for the remainder of this paper, we focus on this BO model, keeping the $\mathcal{E}$ as the objective function.

4.4 Heterogeneity Parameter Importance

We use SAGE (Shapley Additive Global importancE) (Covert et al., 2020), a game-theoretic approach to understand black-box models to calculate the significance of adding heterogeneity to each parameter for improving $\mathcal{C}$ and $\tilde{S}$. SAGE summarizes the importance of each feature based on the predictive power it contributes and considers complex feature interactions using the principles of Shapley value, with a higher SAGE value signifying a more important feature. We tested the HRSNN model using SAGE on the Lorenz96 and the SHD datasets. The results are shown in Fig. 4. We see that τ_m has the greatest SAGE values for $\mathcal{C}$, signifying that it has the greatest impact on improving $\mathcal{C}$ when heterogeneity is added. Conversely, we see that heterogeneous STDP parameters (viz., $\tau_\pm, \eta_\pm$) play a more critical role in determining the average neuronal spike activation. Hence, we confirm the notions proved in Sec. 3 that heterogeneity in neuronal dynamics improves the $\mathcal{C}$ while heterogeneity in STDP dynamics improves the spike count. Thus, we need to optimize the heterogeneity of both to achieve maximum $\mathcal{E}$.

4.5 Results

We perform an ablation study to evaluate the performance of the HRSNN model and compare it to standard BP-based spiking models. We study the performances of both the SHD dataset for classification and the Lorenz system for prediction. The results are shown in Table 2. We compare the Normalized Root Mean Squared Error (NRMSE) loss (prediction), Accuracy (classification), Average

Table 2: Table showing the comparison of the Accuracy and NRMSE losses for the SHD Classification and Lorenz System Prediction tasks, respectively. The results marked with * denotes we implemented the open-source code for the model and evaluated the given results

	Method	**SHD (Classification)**			**Chaotic Lorenz System (Prediction)**		
		Accuracy	**Avg Spike Count ($\tilde{S}$)**	**Efficiency $\hat{\mathcal{E}}$ = $\frac{1000\times\textbf{Accuracy}}{\tilde{S}}$**	**NRMSE**	**Avg Spike Count ($\tilde{S}$)**	**Efficiency $\hat{\mathcal{E}}$ = $\frac{1000}{\textbf{NRMSE}\times\tilde{S}}$**
Unsupervised RSNN	MRSNN (Homogeneous LIF, Homogeneous STDP)	73.58	3991.25	18.44	0.395	3215.48	0.787
	HRSNN (Heterogeneous LIF, Homogeneous STDP)	78.87	4588.19	17.19	0.203	3784.44	1.302
	HRSNN (Homogeneous LIF, Heterogeneous STDP)	74.03	3294.62	22.47	0.372	2884.59	0.932
	HRSNN (Heterogeneous LIF, Heterogeneous STDP)	80.49	3305.37	24.35	0.195	2972.27	1.725
RSNN with BP	MRSNN-BP (Homogeneous LIF, BP)	81.42	4815.91	16.9	0.182	4752.23	1.16
	HRSNN-BP (Heterogeneous LIF, BP)	83.54	5419.3	15.42	0.178	5109.58	1.09
	Adaptive SRNN (Yin et al., 2020)	84.46	5007.61*	17.21	0.174*	4826.36*	1.19*

Spike Count $\tilde{S}$ and the application level empirical spiking efficiency $\hat{\mathcal{E}}$ calculated as $\frac{1000}{\text{NRMSE}\times\tilde{S}}$ (prediction) and $\frac{1000\times\text{Accuracy}}{\tilde{S}}$ (classification). We perform the experiments using 5000 neurons in $\mathcal{R}$ on both classification and prediction datasets. We see that the HRSNN model with heterogeneous LIF and heterogeneous STDP outperforms other HRSNN and MRSNN models in terms of NRMSE scores while keeping the $\tilde{S}$ much lower than HRSNN with heterogeneous LIF and homogeneous STDP. From the experiments, we can conclude that the heterogeneous LIF neurons have the greatest contribution to improving the model's performance. In contrast, heterogeneity in STDP has the most significant impact on a spike-efficient representation of the data. HRSNN with heterogeneous LIF and STDP leverages the best of both worlds by achieving the best RMSE with low spike activations, as seen from Table 2.

5 Conclusion

This paper analytically and empirically proved that heterogeneity in neuronal (LIF) and synaptic (STDP) dynamics leads to an unsupervised RSNN with more memory capacity, reduced spiking count, and hence, better spiking efficiency. We show that HRSNN can achieve similar performance as an MRSNN but with sparse spiking leading to the improved energy efficiency of the network. In conclusion, this work establishes important mathematical properties of an RSNN for neuromorphic machine learning applications like time series classification and prediction. However, it is interesting to note that the mathematical results from this paper also conform to the recent neurobiological research that suggests that the brain has large variability between the types of neurons and learning methods. For example, intrinsic biophysical properties of neurons, like densities and properties of ionic channels, vary significantly between neurons where the variance in synaptic learning rules invokes reliable and efficient signal processing in several animals (Destexhe & Marder, 2004; Marder & Taylor, 2011; Douglass et al., 1993). Experiments in different brain regions and diverse neuronal types have revealed a wide range of STDP forms with varying neuronal dynamics that vary in plasticity direction, temporal dependence, and the involvement of signaling pathways (Sjostrom et al., 2008; Korte & Schmitz, 2016). Thus, heterogeneity is essential in encoding and decoding stimuli in biological systems. In conclusion, this work establishes connections between the mathematical properties of an RSNN for neuromorphic machine learning applications like time series classification and prediction with neurobiological observations.

REFERENCES

Larry F Abbott and Peter Dayan. The effect of correlated variability on the accuracy of a population code. *Neural computation*, 11(1):91–101, 1999.

Pau Vilimelis Aceituno, Gang Yan, and Yang-Yu Liu. Tailoring echo state networks for optimal learning. *Iscience*, 23(9):101440, 2020.

György Buzsáki and Kenji Mizuseki. The log-dynamic brain: how skewed distributions affect network operations. *Nature Reviews Neuroscience*, 15(4):264–278, 2014.

Biswadeep Chakraborty, Xueyuan She, and Saibal Mukhopadhyay. A fully spiking hybrid neural network for energy-efficient object detection. *arXiv preprint arXiv:2104.10719*, 2021.

Ashesh Chattopadhyay, Pedram Hassanzadeh, and Devika Subramanian. Data-driven predictions of a multiscale lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Processes in Geophysics*, 27(3):373–389, 2020.

Lee Cossell, Maria Florencia Iacaruso, Dylan R Muir, Rachael Houlton, Elie N Sader, Ho Ko, Sonja B Hofer, and Thomas D Mrsic-Flogel. Functional organization of excitatory synaptic strength in primary visual cortex. *Nature*, 518(7539):399–403, 2015.

Ian Covert, Scott M Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33:17212–17223, 2020.

Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Alain Destexhe and Eve Marder. Plasticity in single neuron and circuit computations. *Nature*, 431 (7010):789–795, 2004.

John K Douglass, Lon Wilkens, Eleni Pantazelou, and Frank Moss. Noise enhancement of information transfer in crayfish mechanoreceptors by stochastic resonance. *Nature*, 365(6444):337–340, 1993.

Igor Farkaš and Peter Gergel'. Maximizing memory capacity of echo state networks with orthogonalized reservoirs. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2437–2442. IEEE, 2017.

Igor Farkaš, Radomír Bosák, and Peter Gergel'. Computational analysis of memory capacity in echo state networks. *Neural Networks*, 83:109–120, 2016.

Edris Zaman Farsa, Soheila Nazari, and Morteza Gholami. Function approximation by hardware spiking neural network. *Journal of Computational Electronics*, 14(3):707–716, 2015.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2681–2690. PMLR, 2019.

Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Erol Gelenbe, Zhi-Hong Mao, and Yan-Da Li. Function approximation with spiked random networks. *IEEE Transactions on Neural Networks*, 10(1):3–9, 1999.

Wulfram Gerstner and Werner M Kistler. Mathematical formulations of hebbian learning. *Biological cybernetics*, 87(5):404–415, 2002.

Aditya Gilra and Wulfram Gerstner. Predicting non-linear dynamics by stable local learning in a recurrent spiking neural network. *Elife*, 6:e28295, 2017.

Mirko Goldmann, Felix Köster, Kathy Lüdge, and Serhiy Yanchuk. Deep time-delay reservoir computing: Dynamics and memory capacity. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(9):093124, 2020.

Nathan W Gouwens, Staci A Sorensen, Jim Berg, Changkyu Lee, Tim Jarsky, Jonathan Ting, Susan M Sunkin, David Feng, Costas A Anastassiou, Eliza Barkan, et al. Classification of electrophysiological and morphological neuron types in the mouse visual cortex. *Nature neuroscience*, 22(7): 1182–1195, 2019.

David Hansel, Germán Mato, and Claude Meunier. Synchrony in excitatory neural networks. *Neural computation*, 7(2):307–337, 1995.

Hatsuo Hayashi and Jun Igarashi. Ltd windows of the stdp learning rule and synaptic connections having a large transmission delay enable robust sequence learning amid background noise. *Cognitive neurodynamics*, 3(2):119–130, 2009.

Nicolangelo Iannella and Andrew D. Back. A spiking neural network architecture for nonlinear function approximation. *Neural Networks*, 14(6):933–939, 2001. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(01)00080-6. URL https://www.sciencedirect.com/science/article/pii/S0893608001000806.

Herbert Jaeger. 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):13, 2001.

Herbert Jaeger. Short term memory in echo state networks. gmd-report 152. In *GMD-German National Research Institute for Computer Science (2002), http://www. faculty. jacobs-university. de/hjaeger/pubs/STMEchoStatesTechRep. pdf*. Citeseer, 2002.

Herbert Jaeger et al. *Short term memory in echo state networks*, volume 5. GMD-Forschungszentrum Informationstechnik Bremen, Germany, 2001.

Mina A Khoei, Amirreza Yousefzadeh, Arash Pourtaherian, Orlando Moreira, and Jonathan Tapson. Sparnet: Sparse asynchronous neural network execution for energy efficient inference. In *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 256–260. IEEE, 2020.

Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: Spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11270–11277, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Martin Korte and Dietmar Schmitz. Cellular and system biology of memory: timing, molecules, and beyond. *Physiological reviews*, 96(2):647–693, 2016.

Łukasz Kuśmierz, Shun Ogawa, and Taro Toyoizumi. Edge of chaos and avalanches in neural networks with heavy-tailed synaptic weight distribution. *Physical Review Letters*, 125(2):028101, 2020.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, 1996.

Christian K Machens, Ranulfo Romo, and Carlos D Brody. Functional, but not anatomical, separation of "what" and "when" in prefrontal cortex. *Journal of Neuroscience*, 30(1):350–360, 2010.

Rosario N Mantegna and H Eugene Stanley. Scaling behaviour in the dynamics of an economic index. *Nature*, 376(6535):46–49, 1995.

Eve Marder and Adam L Taylor. Multiple models to capture the variability in biological neurons and networks. *Nature neuroscience*, 14(2):133–138, 2011.

Fernando Montani, Robin AA Ince, Riccardo Senatore, Ehsan Arabzadeh, Mathew E Diamond, and Stefano Panzeri. The impact of high-order interactions on the rate of synchronous discharge and information transmission in somatosensory cortex. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1901):3297–3310, 2009.

Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks. *IEEE Signal Processing Magazine*, 36:61–63, 2019.

Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

Erkki Oja. Neural networks, principal components, and subspaces. *International journal of neural systems*, 1(01):61–68, 1989.

Nicolas Perez-Nieves, Vincent CH Leung, Pier Luigi Dragotti, and Dan FM Goodman. Neural heterogeneity promotes robust learning. *Nature communications*, 12(1):1–9, 2021.

Balint Petro, Nikola Kasabov, and Rita M Kiss. Selection and optimization of temporal spike encoding methods for spiking neural networks. *IEEE transactions on neural networks and learning systems*, 31(2):358–370, 2019.

Filip Ponulak and Andrzej Kasinski. Introduction to spiking neural networks: Information processing, learning and applications. *Acta neurobiologiae experimentalis*, 71(4):409–433, 2011.

R Rossi Pool and Germán Mato. Spike-timing-dependent plasticity and reliability optimization: the role of neuron dynamics. *Neural computation*, 23(7):1768–1789, 2011.

Steven A Prescott, Yves De Koninck, and Terrence J Sejnowski. Biophysical basis for three distinct dynamical mechanisms of action potential initiation. *PLoS computational biology*, 4(10):e1000198, 2008.

Ryan Pyle and Robert Rosenbaum. Spatiotemporal dynamics and reliable computations in recurrent spiking neural networks. *Physical review letters*, 118(1):018103, 2017.

Nitin Rathi, Amogh Agrawal, Chankyu Lee, Adarsh Kumar Kosta, and Kaushik Roy. Exploring spike-based learning for neuromorphic computing: Prospects and perspectives. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 902–907. IEEE, 2021.

Xueyuan She, Saurabh Dash, Daehyun Kim, and Saibal Mukhopadhyay. A heterogeneous spiking neural network for unsupervised learning of spatiotemporal patterns. *Frontiers in Neuroscience*, 14:1406, 2021.

Xueyuan She, Saurabh Dash, and Saibal Mukhopadhyay. Sequence approximation using feedforward spiking neural network for spatiotemporal learning: Theory and optimization methods. In *Proceedings of 10th International Conference on Learning Representations, ICLR (forthcoming)*, 2022. URL `https://openreview.net/forum?id=bp-LJ4y_XC`.

Michael F Shlesinger, BJ West, and Joseph Klafter. Lévy dynamics of enhanced diffusion: Application to turbulence. *Physical Review Letters*, 58(11):1100, 1987.

P Jesper Sjostrom, Ede A Rancz, Arnd Roth, and Michael Hausser. Dendritic excitability and synaptic plasticity. *Physiological reviews*, 88(2):769–840, 2008.

Martino Sorbaro, Qian Liu, Massimo Bortone, and Sadique Sheik. Optimizing the energy consumption of spiking neural networks for neuromorphic applications. *Frontiers in neuroscience*, 14:662, 2020.

Nicholas Soures and Dhireesha Kudithipudi. Spiking reservoir networks: Brain-inspired recurrent algorithms that use random, fixed synaptic strengths. *IEEE Signal Processing Magazine*, 36(6): 78–87, 2019.

Gopalakrishnan Srinivasan and Kaushik Roy. Restocnet: Residual stochastic binary convolutional spiking neural network for memory-efficient neuromorphic computing. *Frontiers in neuroscience*, 13:189, 2019.

Benjamin Staude, Stefan Rotter, and Sonja Grün. Cubic: cumulant based inference of higher-order correlations in massively parallel spike trains. *Journal of computational neuroscience*, 29(1): 327–350, 2010.

Tobias Thornes, Peter Düben, and Tim Palmer. On the use of scale-dependent precision in earth system modelling. *Quarterly Journal of the Royal Meteorological Society*, 143(703):897–908, 2017.

Hiroyuki Torikai, Atsuo Funew, and Toshimichi Saito. Digital spiking neuron and its learning for approximation of various spike-trains. *Neural Networks*, 21(2):140–149, 2008. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2007.12.045. URL https://www.sciencedirect.com/science/article/pii/S0893608008000051. Advances in Neural Networks Research: IJCNN '07.

Roman Vershynin. Memory capacity of neural networks with threshold and rectified linear unit activations. *SIAM Journal on Mathematics of Data Science*, 2(4):1004–1033, 2020.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1311–1318, 2019.

Bojian Yin, Federico Corradi, and Sander M Bohté. Effective and efficient computation with multiple-timescale spiking recurrent neural networks. In *International Conference on Neuromorphic Systems 2020*, pp. 1–8, 2020.

Wenrui Zhang and Peng Li. Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Advances in Neural Information Processing Systems*, 33:12022–12033, 2020.

SUPPLEMENTARY SECTION A

SUMMARY OF KEY RESULTS

- ***Increasing the heterogeneity in LIF neuron dynamics increases the memory capacity***: We show the variance of neuronal states is a function of the covariance of the neuronal parameters. Leveraging the inverse relation between memory capacity and neuronal state, we demonstrate that the correlation among the neuronal states (neuronal state correlation) decreases with increasing heterogeneity, hence increasing the memory capacity.
- ***HRSNN with heterogeneous LIF and STDP dynamics achieves similar memory capacity as MRSNN with fewer neurons and sparser activation:*** We prove the variance of membrane potential overall all neurons at a given time point in an HRSNN is less than that of the MRSNN. Hence, the probability of the membrane voltage exceeding the threshold is lower for heterogeneous neuronal populations. Next, we project the neurons on an ellipsoid based on their state and calculate the distance between the points. We prove that as the number of neurons in the recurrent layer $N_{\mathcal{R}}$ decreases, the distance decreases further, increasing the effect of heterogeneity on the neuron states. Using these results, we prove HRSNN demonstrates better spiking efficiency than MRSNN, i.e., it achieves similar performance with fewer spike activations.

EXPERIMENTAL SETUP

HRSNN AND MRSNN

Table 3: Table showing the description of the models described in the paper

	LIF Neurons	**STDP parameters**
HRSNN	Heterogeneous	Heterogeneous
MRSNN	Homogeneous	Homogeneous

The models used in this paper were the Heterogeneous Recurrent SNN (HRSNN) and the Homogeneous Recurrent SNN (MRSNN). Both models use STDP as the learning method. For MRSNN, we use STDP with uniform parameters for all the synapses. However, for HRSNN, we use a distribution for each parameter to get a rich class of diverse LTP/LTD dynamics. But, at the core, all the training is done using STDP.

SPIKE CODING

Encoding: For the RSNN to process our time series, the signal must be represented as spikes. We use a temporal encoding technique for representing signals in this paper. The spikes are only generated whenever the signal changes in value. The implementation of the temporal encoding used in this research is based on the Step-Forward (SF) algorithm Petro et al. (2019). The percentage of neurons to input the spikes to (α) is also chosen to provide good recurrent layer dynamics.

Decoding: To represent the recurrent state, we use an exponentially decreasing rate decoding strategy by taking the sum of all the spikes s over the last τ timesteps into account as follows:

$$x_i^X(t) = \sum_{n=0}^{\tau} \gamma^n s_i(t-n) \quad \forall i \in E$$

where X denotes the model representation. The parameters τ and γ are balanced to optimize the memory size of the stored data (e.g., $\tau \leq 50$) and its containment of information, which includes adjusting τ to the pace at which the temporal data is presented and processed. The state of the recurrent layer will be only based on the output of excitatory neurons. Thus, it is crucial for the discount γ not to be too small, as it possibly flattens older values in the window to 0, making part of the sliding window unusable. Recent spikes hardly affect the recurrent layer state when setting γ too high in combination with a large window size. This causes the decoder to react too late to recent information provided by the recurrent layer and complicates the learning process of the readout layer.

READOUT

After the initialization of the recurrent layer, the readout is the only component of the LSM with trainable parameters. It consists of a single fully connected layer f_θ to do regression or classification. The readout does not have to be any deeper, as the output of the recurrent layer is already a high-dimensional representation of the processed input. x and y present the continuous signals of the time series T and model representation X.

$$x^T(t+k) = y^T(t) \approx \hat{y}^X(t) = f_\theta\left(x^X(t)\right)$$

The mean squared error (MSE) is used as the loss function to train the readout, and the network was trained using the stochastic optimizer Adam Kingma & Ba (2014).

$$\mathcal{L}\left(y^T, \hat{y}^X\right) = \frac{1}{n}\sum_{i=0}^{n}\left(y_i^T - \hat{y}_i^X\right)^2$$

DATASETS

Lorenz96: Our objective is more clearly demonstrated using the canonical chaotic system that we will use as a test bed for the prediction capabilities of the HRSNN model. We use a multiscale Lorenz 96 system which is a set of coupled nonlinear ODEs, and an extension of Lorenz's original model Thornes et al. (2017), Chattopadhyay et al. (2020).

$$\begin{aligned}
\frac{\mathrm{d}X_k}{\mathrm{d}t} &= X_{k-1}\left(X_{k+1} - X_{k-2}\right) + F - \frac{hc}{b}\Sigma_j Y_{j,k} \\
\frac{\mathrm{d}Y_{j,k}}{\mathrm{d}t} &= -cbY_{j+1,k}\left(Y_{j+2,k} - Y_{j-1,k}\right) - cY_{j,k} + \frac{hc}{b}X_k - \frac{he}{d}\Sigma_i Z_{i,j,k} \\
\frac{\mathrm{d}Z_{i,j,k}}{\mathrm{d}t} &= edZ_{i-1,j,k}\left(Z_{i+1,j,k} - Z_{i-2,j,k}\right) - geZ_{i,j,k} + \frac{he}{d}Y_{j,k}
\end{aligned}$$

This set of coupled nonlinear ordinary differential equations (ODEs) is a three-tier extension of Lorenz's original model and has been proposed by Thornes et al. as a fitting prototype for multiscale chaotic variability of the weather and climate system and a useful test bed for novel methods. In these equations, $F = 20$ is a large-scale forcing that makes the system highly chaotic, and $b = c = e = d = g = 10$ and $h = 1$ are tuned to produce appropriate spatiotemporal variability of the three variables. For this paper, we focus on predicting Y axes, which have relatively moderate amplitudes compared to X, Z and demonstrate high-frequency variability and intermittency, which makes the prediction problem difficult. The details about the Lorenz-96 system are given in Supplementary Section A.

SHD dataset: We use the Spoken Heidelberg Digits spiking dataset to benchmark the HRSNN model with other standard spiking neural networks Cramer et al. (2020). It was created based on the Heidelberg Digits (HD) audio dataset which comprises 20 classes of spoken digits from zero to nine in English and German, spoken by 12 individuals. For training and evaluation, the dataset (10420 samples) is split into a training set (8156 samples) and test set (2264 samples). To apply our RSNNs, we converted all audio samples into 250- by-700 binary matrices. For this, all samples fit within a 1 the second window, shorter samples were padded with zeros, and longer samples were cut by removing the tail. Spikes were then binned in time bins, both of sizes 10ms and 4ms; for the RSNNs, the presence or non-presence of any spikes in the time bin is noted as a single binary event.

HYPERPARAMETERS

The hyperparameters used in this paper are summarized in Table 4

MAXIMUM ENTROPY DISTRIBUTION

This subsection proves that the maximum entropy distribution with a fixed covariance matrix is Gaussian.

Table 4: Table showing the hyperparameters used in the experiments and their values

Parameter	Value	Description
|E|/|N|	80%	Excitatory/inhibitory ratio
λ	2	Leak Exponent
τ	50	Sliding window size
$\gamma^{\tau-1}$	0.02	Sliding Window Leak

Mackey Glass Parameters			Lorenz-63 Parameters		
Parameter	**Value**	**Description**	**Parameter**	**Value**	**Description**
τ	17	Time delay	ρ	28.0	ρ-parameter
a	0.2	a parameter	σ	10.0	σ-parameter
b	0.1	b parameter	β	8/3	β-parameter
n	10	n parameter	x_0	[1.0,1.0,1.0]	Initial Condition
x_0	1.2	Initial Condition	h	0.03	Time delta between two discrete timesteps
h	1.0	Time delta between two discrete timesteps			

Lemma: *Let $q(\mathbf{r})$ be any density satisfying $\int q(\mathbf{r}) x_i x_j d\mathbf{r} = \Sigma_{ij}$. Let $p = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Then $h(q) \leq h(p)$*
Proof.

$$\begin{aligned} 0 \leq \mathbb{KL}(q\|p) &= \int q(\mathbf{r}) \log \frac{q(\mathbf{r})}{p(\mathbf{r})} d\mathbf{r} \\ &= -h(q) - \int q(\mathbf{r}) \log p(\mathbf{r}) d\mathbf{r} \\ &= \quad -h(q) - \int p(\mathbf{r}) \log p(\mathbf{r}) d\mathbf{r} \\ &= -h(q) + h(p) \end{aligned}$$

since q and p yield the same moments for the quadratic form encoded by $\log p(\mathbf{r})$.

Optimal Hyperparameter Selection using Bayesian Optimization

While BO is used in various settings, successful applications are often limited to low-dimensional problems Frazier (2018). However, for optimizing a heterogeneous RSNN model, with a huge number of hyperaprameters to be optimized, using BO remains a significant challenge. However, our parameters to be optimized are highly correlated and can be drawn from a probability distribution as shown by Perez-Nieves et al. (2021). Thus, in this paper, we use a modified BO to estimate parameter distributions for the LIF neurons and the STDP dynamics. To learn the probability distribution of the data, we modify the surrogate model and the acquisition function of the BO to treat the parameter distributions instead of individual variables. This makes our modified BO highly scalable over all the variables (dimensions) used. The loss for the surrogate model's update is calculated using the Wasserstein distance between the parameter distributions.

BO uses a Gaussian process to model the distribution of an objective function and an acquisition function to decide points to evaluate. For data points in a target dataset $x \in X$ and the corresponding label $y \in Y$, an SNN with network structure $\mathcal{V}$ and neuron parameters $\mathcal{W}$ acts as a function $f_{\mathcal{V},\mathcal{W}}(x)$ that maps input data x to predicted label $\tilde{y}$. The optimization problem in this work is defined as

$$\min_{\mathcal{V},\mathcal{W}} \sum_{x \in X, y \in Y} \mathcal{L}\left(y, f_{\mathcal{V},\mathcal{W}}(x)\right) \tag{10}$$

where $\mathcal{V}$ is the set of hyperparameters of the neurons in $\mathcal{R}$ (Details of hyperparameters given in the Supplementary) and $\mathcal{W}$ is the multi-variate distribution constituting the distributions of: (i) the membrane time constants τ_{m-E}, τ_{m-I} of the LIF neurons, (ii) the scaling function constants (A_+, A_-) and (iii) the decay time constants τ_+, τ_- for the STDP learning rule in $\mathcal{S}_{\mathcal{RR}}$.

Again, BO needs a prior distribution of the objective function $f(\vec{x})$ on the given data $\mathcal{D}_{1:k} = \{\vec{x}_{1:k}, f(\vec{x}_{1:k})\}$. In GP-based BO, it is assumed that the prior distribution of $f(\vec{x}_{1:k})$ follows the multivariate Gaussian distribution which follows a Gaussian Process with mean $\vec{\mu}_{\mathcal{D}_{1:k}}$ and covariance $\vec{\Sigma}_{\mathcal{D}_{1:k}}$. We estimate $\vec{\Sigma}_{\mathcal{D}_{1:k}}$ using the modified Matern kernel function which is given in Eq. **??**. In

Table 5: The list of parameter settings for the Bayesian Optimization-based hyperparameter search

Parameter	Initial Value	Range
η	10	(0,50)
γ	5	(0,10)
ζ	2.5	(0,10)
η^*	1	(0,3)
g	2	(0,10)
ω	0.5	(0,1)
k	50	(0,100)
λ (SHD)	1	(0,2)
λ (Lorenz)	1.5	(0,4)
P_{IR}	0.05	(0,0.1)
τ_{n-E}, τ_{n-I} (SHD)	$50ms$	$(0ms, 100ms)$
τ_{n-E}, τ_{n-I} (Lorenz)	$100ms$	$(0ms, 300ms)$
$A_{en-R}, A_{EE}, A_{EI}, A_{IE}, A_{II}$	30	(0,60)

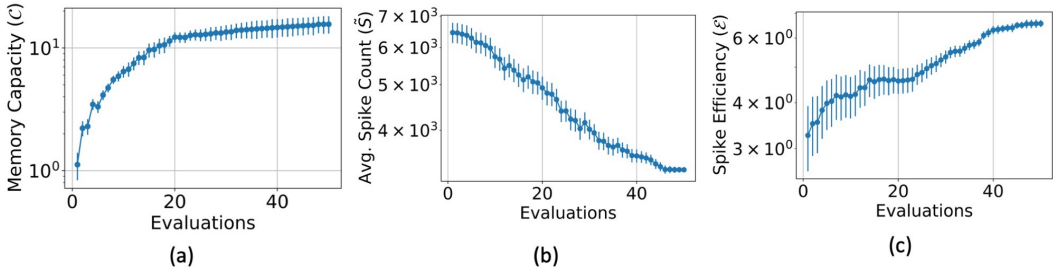

Figure 5: Figure Showing the convergence behaviors of the three types of BO described in the paper (a) BO optimizing the memory capacity $\mathcal{C}$ (b) BO optimizing the average spike count $\tilde{S}$ and (c) BO optimizing the spike efficiency $\mathcal{E}$

this paper, we use $d(x, x')$ as the Wasserstein distance between the multivariate distributions of the different parameters. It is to be noted here that for higher-dimensional metric spaces, we use the Sinkhorn distance as a regularized version of the Wasserstein distance to approximate the Wasserstein distance Feydy et al. (2019).

$\mathcal{D}_{1:k}$ are the points that have been evaluated by the objective function and the GP will estimate the mean $\vec{\mu}_{\mathcal{D}_{k:n}}$ and variance $\vec{\sigma}_{\mathcal{D}_{k:n}}$ for the rest unevaluated data $\mathcal{D}_{k:n}$. The acquisition function used in this work is the expected improvement (EI) of the prediction fitness, as:

$$EI\left(\vec{x}_{k:n}\right) = \left(\vec{\mu}_{\mathcal{D}_{k:n}} - f\left(x_{\text{best}}\right)\right)\Phi(\vec{Z}) + \vec{\sigma}_{\mathcal{D}_{k:n}}\phi(\vec{Z}) \tag{11}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ denote the probability distribution function and the cumulative distribution function of the prior distributions, respectively. $f\left(x_{\text{best}}\right) = \max f\left(\vec{x}_{1:k}\right)$ is the maximum value that has been evaluated by the original function f in all evaluated data $\mathcal{D}_{1:k}$ and $\vec{Z} = \frac{\vec{\mu}_{\mathcal{D}_{k:n}} - f\left(x_{\text{best}}\right)}{\vec{\sigma}_{\mathcal{D}_{k:n}}}$. BO will choose the data $x_j = \operatorname{argmax}\left\{EI\left(\vec{x}_{k:n}\right); x_j \subseteq \vec{x}_{k:n}\right\}$ as the next point to be evaluated using the original objective function.

Optimized Hyperparameters

The list of the hyperparameters optimized using the Bayesian Optimization technique is shown in Table 5. We also show the range of the hyperparameters used and the initial values.

Convergence Analysis

We compare the convergence analysis of the three Bayesian Optimization techniques and the results are shown in Fig. 5. Each of the experiments was repeated five times and the mean and variance

Table 6: Table showing the performance of the Bayesian Optimization on the SHD Classification dataset for the three different cases where BO 1 optimizes $\mathcal{C}$, BO 2 optimizes $\tilde{S}$ and BO 3 optimizes $\mathcal{E}$

N_R	BO 1 Memory Capacity	BO 1 Average Spike Count	BO 1 Accuracy	BO 2 Memory Capacity	BO 2 Average Spike Count	BO 2 Accuracy	BO 3 Memory Capacity	BO 3 Average Spike Count	BO 3 Accuracy
100	5.31 ± 0.36	1399.14 ± 113.66	67.41 ± 4.86	5.22 ± 0.58	1189.37 ± 86.95	66.23 ± 5.74	5.12 ± 0.31	1268.86 ± 91.15	68.67 ± 4.97
200	5.45 ± 0.39	1479.86 ± 133.72	67.85 ± 4.16	5.38 ± 0.53	1233.19 ± 122.56	67.21 ± 5.49	5.28 ± 0.34	1328.18 ± 131.45	69.54 ± 4.27
300	6.72 ± 0.4	1541.14 ± 148.83	68.41 ± 4.87	6.42 ± 0.57	1325.25 ± 128.47	67.95 ± 5.11	6.54 ± 0.36	1391.68 ± 140.37	70.1 ± 4.69
400	7.21 ± 0.44	1682.68 ± 239.95	70.11 ± 4.33	6.91 ± 0.51	1425.69 ± 129.57	68.23 ± 5.27	7.01 ± 0.35	1511.79 ± 200.96	71.54 ± 4.06
500	8.59 ± 0.48	1768.24 ± 287.94	71.05 ± 3.98	7.69 ± 0.46	1555.29 ± 139.17	69.31 ± 5.03	8.63 ± 0.38	1621.8 ± 250.46	73.05 ± 3.87
1000	11.22 ± 0.46	2251.17 ± 319.75	72.93 ± 3.38	9.03 ± 0.44	2015.24 ± 147.44	70.89 ± 4.91	12.25 ± 0.39	2102.59 ± 279.86	75.32 ± 3.44
2000	13.3 ± 0.51	2566.21 ± 348.68	75.36 ± 3.29	9.89 ± 0.46	2314.59 ± 151.18	72.33 ± 4.88	13.95 ± 0.37	2410.08 ± 301.57	77.25 ± 3.17
3000	14.47 ± 0.53	2825.47 ± 355.87	77.14 ± 3.38	10.48 ± 0.41	2623.41 ± 177.94	74.63 ± 4.93	14.88 ± 0.42	2708.52 ± 315.34	78.21 ± 3.24
4000	15.17 ± 0.52	3551.07 ± 366.19	78.05 ± 3.25	11.57 ± 0.45	3045.28 ± 225.53	75.15 ± 4.85	15.87 ± 0.38	3218.42 ± 328.19	79.36 ± 3.13
5000	15.64 ± 0.57	4186.49 ± 383.09	78.92 ± 3.31	11.68 ± 0.48	3294.62 ± 241.14	75.87 ± 4.81	16.03 ± 0.41	3573.51 ± 331.18	80.49 ± 3.15

Table 7: Table showing the performance of the Bayesian Optimization on the Lorenz System Prediction dataset for the three different cases where BO 1 optimizes $\mathcal{C}$, BO 2 optimizes $\tilde{S}$ and BO 3 optimizes $\mathcal{E}$

N_R	BO 1 Memory Capacity	BO 1 Average Spike Count	BO 1 RMSE	BO 2 Memory Capacity	BO 2 Average Spike Count	BO 2 RMSE	BO 3 Memory Capacity	BO 3 Average Spike Count	BO 3 RMSE
100	3.56 ± 0.36	1354.35 ± 108.96	0.617 ± 0.019	3.02 ± 0.52	1101.25 ± 75.93	0.684 ± 0.026	3.45 ± 0.3	1207.52 ± 67.26	0.654 ± 0.0207
200	4.12 ± 0.39	1443.59 ± 127.23	0.587 ± 0.02	3.89 ± 0.48	1207.35 ± 118.57	0.639 ± 0.027	4.01 ± 0.33	1302.47 ± 96.05	0.613 ± 0.0218
300	5.26 ± 0.37	1499.62 ± 141.73	0.503 ± 0.027	4.44 ± 0.55	1257.26 ± 1257.26	0.558 ± 0.034	5.15 ± 0.34	1335.81 ± 168.02	0.531 ± 0.0287
400	6.37 ± 0.41	1528.73 ± 228.87	0.459 ± 0.033	5.87 ± 0.49	1304.35 ± 126.18	0.467 ± 0.04	6.05 ± 0.35	1415.32 ± 213.49	0.482 ± 0.0347
500	7.25 ± 0.45	1601.27 ± 277.97	0.389 ± 0.036	6.25 ± 0.45	1365.35 ± 137.04	0.421 ± 0.043	6.87 ± 0.36	1507.29 ± 219.58	0.411 ± 0.0377
1000	10.12 ± 0.43	1868.14 ± 301.17	0.316 ± 0.04	7.41 ± 0.51	1563.25 ± 146.76	0.396 ± 0.047	9.03 ± 0.37	1699.27 ± 275.79	0.332 ± 0.0417
2000	11.84 ± 0.48	2105.95 ± 331.54	0.293 ± 0.045	8.02 ± 0.5	1854.35 ± 150.28	0.351 ± 0.052	11.44 ± 0.39	2014.12 ± 280.03	0.301 ± 0.0467
3000	13.71 ± 0.51	2408.35 ± 348.26	0.258 ± 0.042	8.94 ± 0.53	2195.82 ± 179.75	0.335 ± 0.058	13.87 ± 0.4	2236.59 ± 281.05	0.241 ± 0.0482
4000	14.15 ± 0.52	2951.56 ± 352.66	0.242 ± 0.063	9.55 ± 0.55	2445.31 ± 217.73	0.326 ± 0.07	14.63 ± 0.41	2546.25 ± 289.81	0.227 ± 0.0649
5000	14.45 ± 0.54	3784.44 ± 353.51	0.203 ± 0.064	9.96 ± 0.58	2684.59 ± 234.63	0.302 ± 0.071	15.12 ± 0.42	2898.27 ± 307.14	0.195 ± 0.0655

of the observations are shown in the Figure. It is to be noted here that since we define the BO as a minimization principle, we minimize $\frac{1}{\mathcal{C}}$, $\tilde{S}$ and $\frac{1}{\mathcal{E}}$.

Comparing Bayesian Optimization Objective Functions

We show the results of the Bayesian Optimization results for the three cases we are considering in this paper for both the classification and prediction problems. The results for the classification problem are shown in Table 6. We tabulate the memory capacity, the average spike count and the observed accuracy for the three BO cases. Similarly, the results for the prediction problem are shown in Table 7. In that case, we tabulate the memory capacity, the average spike count and the observed NRMSE for the three BO cases. We rerun each of the experiments 5 times and report the mean and standard deviation of the results obtained.

Supplementary Section B

Memory Capacity

Let $x(t) \in U$ (where $-\infty < t < +\infty$ and $U \subset \mathbb{R}$ is a compact interval) be a single-channel stationary input signal. Assume that we have a RSNN, specified by its internal weight matrix $\mathbf{W}$, its input weight vector $\mathbf{w}^{\text{in}}$ and the the unit output functions $\mathbf{f}, \mathbf{f}^{\text{out}}$. The network receives $x(t)$ at its input unit. For a given delay τ and an output unit y_τ with connection weight vector $\mathbf{w}_\tau^{\text{out}}$ we consider the determination coefficient

$$
\begin{aligned}
& d\left[\mathbf{w}_\tau^{\text{out}}\right](x(t-\tau), y_\tau(t)) = \\
& = d\left(x(t-\tau), \mathbf{w}_\tau^{\text{out}} \begin{pmatrix} x(t) \\ \mathbf{r}(t) \end{pmatrix}\right) \\
& = \frac{\operatorname{Cov}^2\left(x(t-\tau), y_\tau(t)\right)}{\sigma^2(x(t))\sigma^2\left(y_\tau(t)\right)}
\end{aligned}
$$

where Cov denotes covariance and σ^2 variance. 1. The τ-delay Memory capacity of the network is defined by

$$\mathcal{C}_\tau = \max_{\mathbf{w}_\tau^{\text{out}}} d\left[\mathbf{w}_\tau^{\text{out}}\right](x(t-\tau), y_\tau(t))$$

2. The Memory capacity of the network is

$$\mathcal{C} = \sum_{\tau=1}^{\infty} \mathcal{C}_\tau.$$

The determination coefficient of two signals is the squared correlation coefficient. It ranges between 0 and 1 and represents the fraction of variance explainable in one signal by the other. Thus, the Memory capacity measures how much variance of the delayed input signal can be recovered from optimally trained output units, summed over all delays. Note that the output units do not interfere; arbitrarily, many output units y_τ can be attached to the same network.

The performance of the heterogeneous network model derives from its ability to retain the memory of previous inputs. To quantify the relationship between the recurrent layer dynamics and the memory capacity, we note that the extraction of information from the recurrent layer is made through a linear combination of the neurons' states. Hence, more linearly independent neurons would offer more variable states and, thus, more extended memory.

For reservoir computing (RC), Jaeger Jaeger (2002) shows that $\mathcal{C}$ is bounded by the reservoir network size of the linear RC with the identity activation function and the independent and identically distributed (i.i.d.) model input. Memory capacity ($\mathcal{C}$) is used to quantify the memory of RSNN. Such memory capacity measures the ability of RC to reconstruct precisely the past information of the model input. Also, the network's structural properties can greatly impact the $\mathcal{C}$ of the linear RC. Now, the question arises what is the need to maximize the memory capacity of the network? The $\mathcal{C}$ normally serves as a global index to quantify the memory property of the network. To comprehensively examine the memory property deeply, the local measurement of its memory property is indispensable. Thus, maximizing the $\mathcal{C}$ acts as an estimator for better prediction results of the trained network.

Since the first-order approximation of the model is linear, the heterogeneity between state variables depends on all the eigenvalues of the adjacency matrix, with a larger mean eigenvalue meaning higher heterogeneity. Hence we can use the eigenvalues $\{\lambda_i\}$ of the weight matrix W to quantify how fast the input decays in the recurrent layer approximately. In other words, the eigenvalues of W should be related to the memory capacity of the heterogeneous neural network model. Indeed, we find that the average eigenvalue modulus: $\langle|\lambda|\rangle = 1/N_R \sum_{i=1}^{N_R} |\lambda_i|$ strongly correlates with $\mathcal{H}$ and therefore with $\mathcal{C}$ as well. Note that, as opposed to $\mathcal{C}$ and $\mathcal{H}$, $\langle|\lambda|\rangle$ is much easier to compute and is solely determined by the recurrent layer network.

The memory capacity reflects the precision with which previous inputs can be recovered. The nonlinearity of the recurrent layer and other far-in-the-past inputs induce noise that complicates recovery. Thus, similar to the analysis done by Aceituno et al. Aceituno et al. (2020) for Echo state networks, the variance of the linear part of the recurrent layer is placed to maximize the recoverable information. Thus, the inputs are projected into orthogonal directions of the recurrent layer state space to not add noise to each other. The spread of variance across the different dimensions should be evenly distributed within those orthogonal directions, quantified by the neurons' covariance.

We start by noticing that the linear nature of the projection vector $\mathbf{w}_{\text{out}}$ implies that we are treating the system as

$$\mathbf{r}(t) = \sum_{\tau=0}^{\infty} \mathbf{a}_\tau x(t-\tau) + \varepsilon(t) \tag{12}$$

where the vectors $\mathbf{a}_\tau \in \mathbb{R}^{N_R}$ correspond to the linearly extractable effect of $x(t-\tau)$ onto $\mathbf{r}(t)$ and $\varepsilon(t)$ is the nonlinear contribution of all the inputs onto the state of $\mathbf{r}(t)$.

Previous works have shown that linear recurrent layers have more extended memory, but nonlinearity is needed to perform interesting computations. Here we show that for a fixed ratio of the nonlinearity, greater heterogeneity leads to a lesser neuronal correlation, leading to a higher memory capacity.

To maintain this trade-off between linear and non-linear behavior, we will assume that linear and non-linear strengths distribution is fixed. This can be achieved if we impose that the probabilities of the neuron states do not change, meaning that the mean, variance, and other moments of the neuron

outputs are unchanged; hence, the strength of the non-linear effects is unchanged. A first constraint can also be obtained from the maintained strength of the linear side of Eq.12

$$\mathrm{Var}\left(\sum_{\tau=1}^{\infty} \mathbf{a}_\tau x(t-\tau)\right) = c \tag{13}$$

where c is a constant.

Lemma 3.1.1: *The state of the neuron can be written as follows:*

$$r_i(t) = \sum_{k=0}^{N_R}\sum_{n=1}^{N_R} \lambda_n^k \left\langle v_n^{-1}, \mathbf{w}^{\mathrm{in}}\right\rangle (v_n)_i \, x(t-k) \tag{14}$$

where $\mathbf{v}_n, \mathbf{v}_n^{-1} \in \mathbf{V}$ are, respectively, the left and right eigenvectors of $\mathbf{W}$, and $\lambda_n^k \in \lambda$ belongs to the diagonal matrix containing the eigenvalues of $\mathbf{W}$; $\mathbf{a}_i = [a_{i,0}, a_{i,1}, \ldots]$ represents the coefficients that the previous inputs $\mathbf{x}_t = [x(t), x(t-1), \ldots]$ have on $r_i(t)$.

Proof: We build on the work of Aceituno et al. Aceituno et al. (2020) where they showed that higher heterogeneity among the neuronal states implies higher memory capacity. Here we aim to show that as the number of neurons N_R in the recurrent layer decreases, heterogeneity increases the spectral radius. More formally, the spectral radius $|\lambda_n|$ is directly proportional to $\mathcal{H}$ as N_R decreases. We express the state of a neuron $r_i(t)$ as

$$r_i(t) = \sum_{k=0}^{\infty} \left(W^k \mathbf{w}^{\mathrm{in}}\right)_i x(t-k) = \sum_{k=0}^{\infty} a_{i,k} x(t-k) = \langle \mathbf{a}_i, \mathbf{x}_t \rangle \tag{15}$$

where the vector $\mathbf{a}_i = [a_{i,0}, a_{i,1}, \ldots]$ represents the coefficients that the previous inputs $\mathbf{x}_t = [x(t), x(t-1), \ldots]$ have on $r_i(t)$. We can then plug this into the covariance between two neurons,

$$\begin{aligned}
\mathrm{Cov}\left(r_i, r_j\right) &= \lim_{T\to\infty} \frac{1}{T} \sum_{q=t}^{t+T} \langle \mathbf{a}_i, \mathbf{x}_q \rangle \langle \mathbf{a}_j, \mathbf{x}_q \rangle \\
&= \langle \mathbf{a}_i, \mathbf{a}_j \rangle \lim_{T\to\infty} \frac{1}{T} \sum_{q_i=0}^{T} \sum_{q_j=0}^{T} \left\langle \mathbf{x}_{q_i}, \mathbf{x}_{q_j} \right\rangle \\
&= \langle \mathbf{a}_i, \mathbf{a}_j \rangle \lim_{T\to\infty} \frac{1}{T} \sum_{q=0}^{T} \langle \mathbf{x}_q, \mathbf{x}_q \rangle \\
&= \langle \mathbf{a}_i, \mathbf{a}_j \rangle \times \mathbb{E}\left[x^2(t)\right] \\
&= \langle \mathbf{a}_i, \mathbf{a}_j \rangle
\end{aligned} \tag{16}$$

Now we write $\mathbf{a}_i$ as a function of the eigenvalues of $\mathbf{W}$. Using the eigenvalue decomposition of the weight matrix $\mathbf{W}$, we rewrite the state of the neuron as follows:

$$r_i(t) = \sum_{k=0}^{N_R}\sum_{n=1}^{N_R} \lambda_n^k \left\langle v_n^{-1}, \mathbf{w}^{\mathrm{in}}\right\rangle (v_n)_i \, x(t-k) \tag{17}$$

where $\mathbf{v}_n, \mathbf{v}_n^{-1} \in \mathbf{V}$ are, respectively, the left and right eigenvectors of $\mathbf{W}$, and $\lambda_n^k \in \lambda$ belongs to the diagonal matrix containing the eigenvalues of $\mathbf{W}$; $\mathbf{a}_i = [a_{i,0}, a_{i,1}, \ldots]$ represents the coefficients that the previous inputs $\mathbf{x}_t = [x(t), x(t-1), \ldots]$ have on $r_i(t)$. ■

Theorem 1: *For HRSNN, heterogeneity in the neuronal parameters $\mathcal{H}$ varies inversely to the correlation among the neuronal states measured as $\sum_{n=1}^{N_\mathcal{R}} \sum_{m=1}^{N_\mathcal{R}} \mathrm{Cov}^2\left(x_n(t), x_m(t)\right)$ which in turn varies inversely with the memory capacity. Hence, the model's memory capacity increases as the heterogeneity in the parameters increases.*

Proof: As shown by Aceituno et al. Aceituno et al. (2020), the memory capacity increases when the variance along the projections of the input into the recurrent layer state has higher heterogeneity.

This can be expressed in terms of the state space of the recurrent layer. Now, we aim to project the inputs into orthogonal directions of the network state space. Thus, we model the system as

$$\mathbf{r}(t) = \sum_{\tau=1}^{\infty} \mathbf{a}_\tau x(t-\tau) + \varepsilon(t) \tag{18}$$

where the vectors $\mathbf{a}_\tau \in \mathbb{R}^N$ are correspond to the linearly extractable effect of $x(t-\tau)$ onto $\mathbf{r}(t)$ and $\varepsilon(t)$ is the nonlinear contribution of all the inputs onto the state of $\mathbf{r}(t)$.

Since our goal is to have a variance as homogeneous as possible along with the directions of a_τ, we need a variance that is as homogeneous along with orthogonal directions, where the vectors $\mathbf{a}_\tau \in \mathbb{R}^N$ correspond to the linearly extractable effect of the input variable $x(t)$ onto the states of the neurons ($\mathbf{r}(t)$).

Now,

$$\left(\sum_{n=1}^{N_\mathcal{R}} \lambda_n(\mathbf{\Sigma})\right)^2 = (\mathrm{tr}[\mathbf{\Sigma}])^2 = (\sum_{n=1}^{N_\mathcal{R}} \mathrm{Var}\left(r_n(t)\right))^2 \tag{19}$$

which is constant by the assumption that the probability distributions of the neuron activities are fixed. Hence we can focus on the value of $\sum_{n=1}^{N_\mathcal{R}} \lambda_n^2(\mathbf{\Sigma})$ which is true since

$$\mathbf{\Sigma}^k \mathbf{e}_n(\mathbf{\Sigma}) = \lambda_n(\mathbf{\Sigma})\mathbf{\Sigma}^{k-1}\mathbf{e}_n(\mathbf{\Sigma}) = \lambda_n^k(\mathbf{\Sigma})\mathbf{e}_n(\mathbf{\Sigma}) \Rightarrow \sum_{n=1}^{N_\mathcal{R}} \lambda_n^2(\mathbf{\Sigma}) = \mathrm{tr}\left[\mathbf{\Sigma}^2\right] \tag{20}$$

where $\mathbf{e}_n(\mathbf{\Sigma})$ and $\lambda_n(\mathbf{\Sigma})$ are, resp. the nth eigenvector and eigenvalue of $\mathbf{\Sigma}$. Hence, we can compute this by decomposing the square of the covariance matrix as follows:

$$\sum_{n=1}^{N_\mathcal{R}} \lambda_n^2(\mathbf{\Sigma}) = \sum_{n=1}^{N_\mathcal{R}} \sum_{m=1}^{N_\mathcal{R}} \mathbf{\Sigma}_{nm}\mathbf{\Sigma}_{mn} = \sum_{n=1}^{N_\mathcal{R}} \sum_{m=1}^{N_\mathcal{R}} \mathrm{Cov}^2\left(x_n(t), x_m(t)\right) \tag{21}$$

where $\mathbf{\Sigma}_{ij}$ are the factor matrices obtained using Cholesky decomposition of $\mathbf{\Sigma}$. Thus, $\sum_{n=1}^{N_\mathcal{R}} \lambda_n^2(\mathbf{\Sigma})$ increases as the neurons become more correlated, hence heterogeneity decreases. Therefore, using heterogeneity, the mean with respect to the square root of the raw variance

$$\hat{\mathcal{H}} = \frac{\sum_{n=1}^{N_\mathcal{R}} \lambda_n^2(\mathbf{\Sigma})}{\left(\sum_{n=1}^{N_\mathcal{R}} \lambda_n(\mathbf{\Sigma})\right)^2} \tag{22}$$

Thus, heterogeneity is inversely proportional to $\sum_{n=1}^{N_\mathcal{R}} \mathrm{Cov}^2\left(x_n(t), x_m(t)\right)$. We see that increasing the correlations between neuronal states decreases the heterogeneity of the eigenvalues, which would reduce the memory capacity of the model. We show that the heterogeneity is bounded by the determinant of the covariance between neuronal parameters. Now, using 21, we see that the covariance increases when the neurons become correlated, and as neuronal state correlation decreases, $\mathcal{H}$ increases, i.e., the heterogeneity increases. Aceituno et al. Aceituno et al. (2020) proved that the neuronal state correlation is inversely related to the memory capacity of the network. Hence, we claim that as $\mathcal{H}$ increases, the memory capacity $\mathcal{C}$ also increases. Thus, we prove that increasing the heterogeneity in the neuronal parameters increases the memory capacity of the model. ■

Spiking Efficiency

The goal of using heterogeneity in the STDP dynamics is to get better orthogonalization among the recurrent network states to lower higher-order correlations in spike trains. Studies have shown that the correlation of higher order progressively decreases the information available through neural population (Montani et al., 2009; Abbott & Dayan, 1999) . Since we are trying to engineer a spike-efficient model, we leverage the heterogeneity in the STDP dynamics to reduce the higher-order correlations. The hypothesis is that using heterogeneity in STDP helps us orthogonalize the recurrent layer that can help us achieve an *efficient representation* of the input spike patterns with fewer spikes. This may be interpreted as the recurrent layer acting as an orthogonal bases function where inputs are projected onto these bases. Thus, having orthogonal bases can efficiently map inputs without much loss. While heterogeneous LIF neurons help us increase the number of principal components, thereby

enabling us to store a greater subclass of features, heterogeneous STDP helps us efficiently encode this orthogonalization of the recurrent layer, resulting in a lesser voltage variance across the neuron population. This leads to fewer spikes (since the mean is constant) compared to a homogeneous RSNN. Thus, in effect, heterogeneous STDP parameters can learn the output more precisely, which is projected back into the recurrent network. One of the primary reasons why heterogeneous STDP helps project the input to orthogonal activations of the recurrent network can be attributed to the distribution of LTD dynamics, as this increases the competition and helps in distributing the projection of the inputs to multiple principal components. (See Suppl. Sec. C for more details.) We discuss that the heterogeneous LTP/LTD dynamics in STDP cause the disappearance of late post-synaptic spikes, leading to fewer spikes for the transmission of information.

Lemma 3.2.1: *Heterogeneity in LTP/LTD dynamics of STDP in HRSNN promotes sparsity in the neural firing as it leads to a lower variance for the distribution of membrane voltages in the neuronal population than MRSNN keeping the mean constant which can be represented as follows:*

$$\Delta\mathrm{Var}[V_{\text{Heterogeneous}}] < \Delta\mathrm{Var}[V_{\text{Homogeneous}}] \tag{23}$$

where V_X denotes the random variable denoting the distribution of membrane voltages for the LIF neurons of type X, X = Homogeneous or Heterogeneous.

Proof: Let us consider a single postsynaptic spike at time t_{pos}. For $t < t_{\text{post}}$,

$$v(t) = \sum_{t_k<t} w_k e^{-\frac{t-t_k}{\tau_m}}, \tag{24}$$

and initially $v(t) < v_{th}$. To study the effect of the spike time when the weight w_k changes, we look into the expected value of the time difference in the post-synaptic spikes.

$$\mathbb{E}\left[\Delta t_{\text{post}}\right] = \mathbb{E}\left[t_{\text{post}} - t_{\text{post}}\right] = \left(\mathbb{E}\left[t_{\text{post}}\right] - t_{\text{post}}\right)\Pr[s] \tag{25}$$

where r accounts for the number of times the spike train has been repeated, and $\Pr[s]$ is the probability that a post-synaptic spike still exists.

We know that increasing the excitatory input weights can only lead to an earlier post-synaptic spike because $v(t)$ can only increase, and thus it might reach v_{th} earlier. We assume that when the expected input increases, the expected post-synaptic firing time also increases, i.e., $\mathbb{E}\left[\Delta t_{\text{post}}\right]$ is a function that decreases monotonically with

$$\mathbb{E}\left[\Delta v(t)\right] = \mathbb{E}\left[\int_{-\infty}^{t} \Delta i(t) e^{-\frac{t-x}{\tau_m}} dx\right] = \int_{-\infty}^{t} \mathbb{E}\left[\Delta i(t)\right] e^{-\frac{t-x}{\tau_m}} dx \tag{26}$$

for all $t < t_{\text{post}}$, meaning that if the expected value of $\Delta v(t)$ averaged over all realizations of the input spike train producing a spike at t_{post} is positive, then $\mathbb{E}\left[\Delta t_{\text{post}}\right]$ will be negative.

We now study the expected input to the neuron at time t ($\mathbb{E}\left[i(t)\right]$), which is decomposed into its excitatory and inhibitory components as $\mathbb{E}\left[i_e(t)\right], \mathbb{E}\left[i_i(t)\right]$.

$$\mathbb{E}\left[\Delta i(t)\right] = \Delta\mathbb{E}\left[i_e(t)\right] - \Delta\mathbb{E}\left[i_i(t)\right] \quad \text{for } t < t_{\text{post}} \tag{27}$$

This implies,

$$\mathbb{E}\left[i_e(t)\right] = \rho_e \int_0^{\infty} \mu_{w_e}(w,t) dw \qquad \mathbb{E}\left[i_i(t)\right] = \rho_i \int_0^{\infty} \mu_{w_i}(w,t) dw \tag{28}$$

where ρ_e, ρ_i are the rates of incoming spikes and $\mu_{w_e}(w,t), \mu_{w_i}(w,t)$ the probabilities of the weights associated to time t.

Thus, Eq. 25 reduces to

$$\int_0^{\infty} [\rho_e \Delta w_e(r) \mu_{w_e}(w,t) - \rho_i \Delta w_i(r) \mu_{w_i}(w,t)] dw, \tag{29}$$

where $\Delta w(r)$ are given by STDP over r repetitions of the input spike train.

Now,

$$\Delta\,\mathrm{Var}[v(t)] = \Delta \int_{-\infty}^{t} \mathrm{Var}[i(t)] dt = \Delta \int_{-\infty}^{t} \left(\mathbb{E}\left[i^2(t)\right] - \mathbb{E}[i(t)]^2\right) dt \tag{30}$$

Let us consider the case of a neuron population with homogeneous STDP (M) and another population with heterogeneous STDP (R). Thus, the difference in the variances between the two populations is given as follows:

$$\Delta\text{Var}[V_M] - \Delta\text{Var}[V_R] = \Delta \int_{-\infty}^{t} \mathbb{E}\left[i_M^2(t)\right] dt - \Delta \int_{-\infty}^{t} \left(\mathbb{E}\left[i_R^2(t)\right] - \mathbb{E}[i_R(t)]^2\right) dt \tag{31}$$

Since $t < t_{\text{post}}$, STDP potentiates both inhibitory and excitatory synapses, so $\Delta\mathbb{E}\left[i_i^2(t)\right] > 0$, $\Delta\mathbb{E}\left[i_e^2(t)\right] > 0$.

The term $\mathbb{E}[i_M(t)]^2 = 0$ by the symmetry of the weights, and it is maintained at zero by the symmetry of the STDP. But for heterogeneous neuron populations, as described above, there exists an asymmetry of the weights. Based on balanced spiking neural networks with heterogeneous connection strengths, previous works have revealed that such heterogeneous networks possess heavy-tailed, Lévy fluctuations Shlesinger et al. (1987); Mantegna & Stanley (1995); Cossell et al. (2015). The heterogeneous heavy-tailed distributions of synaptic weights have been fitted to lognormal distributions Buzsáki & Mizuseki (2014); Kuśmierz et al. (2020). Since the heterogeneous network follows a heavy-tailed distribution, the integral of the probability distribution associated with time t, μ_{w_M} is lesser than the non-heterogenous STDP, which does not have a heavy tail distribution. This implies

$$\begin{aligned} \mathbb{E}[i_R(t)]^2 > 0 \Rightarrow \Delta \int_{-\infty}^{t} \left(\mathbb{E}\left[i_R^2(t)\right] - \mathbb{E}[i_R(t)]^2\right) dt < \Delta \int_{-\infty}^{t} \mathbb{E}\left[i_M^2(t)\right] dt \\ \Rightarrow \Delta\text{Var}[V_R] < \Delta\text{Var}[V_M] \end{aligned} \tag{32}$$

■

For the next proof, we define the heterogeneity in neuronal states over time that arises due to the heterogeneity in the neuronal and learning parameters. It is defined as the inverse of the mean inner product of the square of the covariance matrix of the pairwise neuronal states of the network and is given as: $\mathcal{V} = 1/\langle \text{Cov}^2(\mathbf{r}_i(t), \mathbf{r}_j(t))\rangle$

Theorem 2: *For a given number of neurons $N_{\mathcal{R}}$, the spike efficiency of the model $\mathcal{E} = \frac{\mathcal{C}(N_{\mathcal{R}})}{S}$ for HRSNN ($\mathcal{E}_R$) is greater than MRSNN ($\mathcal{E}_M$) i.e., $\mathcal{E}_R \geq \mathcal{E}_M$*

Proof: To study the effect of the spike time when the weight w_k changes, we look into the expected value of the time difference in the post-synaptic spikes, which is given as:

$$\mathbb{E}\left[\Delta t_{\text{post}}\right] = \mathbb{E}\left[t_{\text{post}} - t_{\text{post}}\right] = \left(\mathbb{E}\left[t_{\text{post}}\right] - t_{\text{post}}\right) \Pr[s] \tag{33}$$

where $\Pr[\exists s]$ is the probability of occurrence of the post-synaptic spike. Thus, the expected input to the neuron at time $t(\mathbb{E}[i(t)])$, which comprises of its excitatory and inhibitory components $\mathbb{E}[i_e(t)], \mathbb{E}[i_i(t)]$ can be expressed as:

$$\mathbb{E}[\Delta i(t)] = \Delta\mathbb{E}[i_e(t)] - \Delta\mathbb{E}[i_i(t)] \quad \text{for } t < t_{\text{post}} \tag{34}$$

$$\text{where} \quad \mathbb{E}[i_e(t)] = \rho_e \int_0^{\infty} \mu_{w_e}(w,t) dw \quad ; \quad \mathbb{E}[i_i(t)] = \rho_i \int_0^{\infty} \mu_{w_i}(w,t) dw \tag{35}$$

where ρ_e, ρ_i are the rates of incoming spikes and $\mu_{w_e}(w,t), \mu_{w_i}(w,t)$ the probabilities of the weights associated to time t. Now, considering the case for RSNNs with homogeneous STDP (M) and with heterogeneous STDP (R), the difference in the variances of the two populations is given as:

$$\Delta\text{Var}[V_M] - \Delta\text{Var}[V_R] = \Delta \int_{-\infty}^{t} [\mathbb{E}\left[i_M^2(t)\right] - \left(\mathbb{E}\left[i_R^2(t)\right] - \mathbb{E}[i_R(t)]^2\right)] dt \tag{36}$$

Since $t < t_{\text{post}}$, STDP potentiates both inhibitory and excitatory synapses, so $\Delta\mathbb{E}\left[i_i^2(t)\right] > 0, \Delta\mathbb{E}\left[i_e^2(t)\right] > 0$. The term $\mathbb{E}[i_M(t)]^2 = 0$ by the symmetry of the weights, and it is maintained at zero by the symmetry of the STDP. But for heterogeneous neuron populations, as described above, there exists an asymmetry of the weights. Based on balanced spiking neural networks with heterogeneous connection strengths, previous works have revealed that such heterogeneous networks possess heavy-tailed, Lévy fluctuations Shlesinger et al. (1987); Mantegna & Stanley (1995); Cossell et al. (2015). This implies $\mathbb{E}[i_R(t)]^2 > 0 \Rightarrow \Delta\text{Var}[V_R] < \Delta\text{Var}[v(t)_M]$ We calculate the number of

post-synaptic spikes triggered when the stimulus is present. Now, representing the spike frequency of the HRSNN and the MRSNN as f_R, f_M resp.,

$$\int_0^t f_R^{ISI}(t)dt \leq \int_0^t f_M^{ISI}(t) \Rightarrow S_R = N_{\mathcal{R}} \frac{T}{\hat{t}_R^{ISI}} \leq N_{\mathcal{R}} \frac{T}{\hat{t}_M^{ISI}} = S_M \tag{37}$$

Thus, spikes decrease when we use heterogeneity in the LTP/LTD Dynamics. Hence, we compare the efficiencies of the HRSNN with that of MRSNN as follows:

$$\frac{\mathcal{E}_R}{\mathcal{E}_M} = \frac{M_R(N_{\mathcal{R}}) \times S_M}{S_R \times M_M(N_{\mathcal{R}})} = \frac{\sum_{\tau=1}^{N_{\mathcal{R}}} \frac{\mathrm{Cov}^2\left(x(t-\tau), \mathbf{a}_\tau^R \mathbf{r}_R(t)\right)}{\mathrm{Var}(\mathbf{a}_\tau^R \mathbf{r}_R(t))} \times \int\limits_{t_{ref}}^{\infty} t f_R^{ISI} dt}{\sum_{\tau=1}^{N_{\mathcal{R}}} \frac{\mathrm{Cov}^2(x(t-\tau), \mathbf{a}_\tau^M \mathbf{r}_M(t))}{\mathrm{Var}(\mathbf{a}_\tau^M \mathbf{r}_M(t))} \times \int\limits_{t_{ref}}^{\infty} t f_M^{ISI} dt} \tag{38}$$

Since $S_R \leq S_M$ and also,the covariance increases when the neurons become correlated, and as neuronal correlation decreases, $\mathcal{H}$ increases (Theorem 1), we see that $\frac{\mathcal{E}_R}{\mathcal{E}_M} \geq 1 \Rightarrow \mathcal{E}_R \geq \mathcal{E}_M$ ■

SUPPLEMENTARY SECTION C

HIGHER ORDER CORRELATION

In this paper, we took inspiration from results in reservoir computing, which show that we can maximize memory capacity using orthogonalization among reservoir states in the case of reservoir computers Farkaš & Gergel' (2017), Farkaš et al. (2016). The goal of using heterogeneous STDP dynamics is to get better orthogonalized recurrent network states to achieve more efficient information transfer with lower higher-order correlations in spike trains. Recent studies Montani et al. (2009), Abbott & Dayan (1999) have shown that the correlation of higher order progressively decreases the information available through the neural population. The decrease in information becomes larger as the interaction order grows. Since we are trying to engineer a spike-efficient model, we leverage the heterogeneity in neuronal parameters to reduce the higher-order correlations. The hypothesis is that an orthogonal recurrent layer can help us achieve an efficient representation of the input spike patterns with fewer spikes. This may be interpreted as the recurrent layer acting as an orthogonal bases function where the inputs are projected onto these bases. Thus, having orthogonal bases can efficiently map the inputs without much loss. The heterogeneous STDP helps us efficiently achieve this orthogonalization of the recurrent layer, resulting in a lesser voltage variance across the neuron population. This leads to fewer spikes (since the mean is constant) compared to a homogeneous RSNN. Thus, in effect, heterogeneous STDP parameters can learn the output more precisely, which is projected back into the recurrent network. Hence, using heterogeneous STDP parameters leads to a better orthogonalization among the neuronal states and hence, a higher $\mathcal{C}$.

In this paper, we show that using a distribution of LTP/LTD dynamics in the STDP parameters helps us in mappings the input onto the orthogonal activations of the recurrent network to capture the principal components of the input signal. The LTD dynamics play an important role in determining the orthogonality of neuronal activations. LTD windows of the STDP rules enable robust sequence learning amid background noise in cooperation with a large signal transmission delay between neurons and a theta rhythm Hayashi & Igarashi (2009). The LTD window in the range of positive spike-timing plays an important role in preventing noise influences with sequence learning. Oja Oja (1982), Oja (1989) showed that the LIF neuron's time constant is very fast compared to the time constant of learning in which the weights w_{ji} change. The learning is assumed to take place according to the STDP type conjunction of the inputs ξ_i and the integrated effect of the inputs, ν_j, with an additional forgetting term attributed to the LTD dynamics: $\frac{dw_{ji}}{dt} = \alpha \nu_j \xi_i - f\left(\nu_j, \xi_i, w_{ji}\right)$ In the case of homogeneous STDP, $f(.)$ is a constant; hence, the model can only efficiently learn the first principal component of the input. However, quite interesting functions emerge when considering STDP to have a distribution. This also helps us determine the next principal components other than the first one. Hence the diversity in the different LTD dynamics increases the competition and helps that not all inputs are mapped to the first principle component. Thus, the diversity in the LTD dynamics helps in projecting the input to orthogonal activations of the recurrent network.

Now, for homogeneous RSNNs, several higher-order correlations, which according to our hypothesis, arise because of the poor orthogonalization among the network states. This results in the redundancies

of spikes for encoding the same information. In this paper, we use heterogeneous STDP dynamics to learn an efficient orthogonal representation of the state space, which result in the network learning the same patterns but using fewer spikes. (theorem: 3) We also show that heterogeneity in the neuronal parameters decreases the neuronal correlation (theorem 1 And fig 2a). Thus, since heterogeneity results in better orthogonalization among the neuronal states, it results in fewer higher-order correlations. Moreover, recent studies have shown that the correlation of higher order progressively decreases the information available through the neural population, and the decrease in information becomes larger as the interaction order grows. Since we are trying to engineer an efficient model, we aim to reduce the higher-order correlations using heterogeneity in neuronal parameters (as shown in Theorem 1). In addition to this, to verify this, we used CuBIC Staude et al. (2010), a cumulant-based inference of higher-order correlations in massively parallel spike trains. The details of the experimental methodology are given in Supplementary Section C. The outcome of CuBIC is a lower bound $\hat{\xi}$ on the order of correlation in the spiking activity of large groups of simultaneously recorded neurons. CuBIC can provide statistical evidence for large correlated groups without the discouraging requirements on a sample size that direct tests for higher-order correlations have to meet. This is achieved by exploiting constraining relations among correlations of different orders. However, it must be noted that CuBIC is not designed to estimate the order of correlation directly; the inferred lower bound might not always correspond to the maximal order of correlation present in a given data set.

Table 8: Table Showing the estimated highest order of correlation for HRSNN vs. MRSNN using CuBIC

	$\hat{\xi}$	p-value	Estimated Highest-order of correlation ($\hat{\xi}+1$)
HRSNN	2	0.423	3
MRSNN	5	0.358	6