Chapter 4

Timescale Heterogeneiety in Reservoir Computing

Abstract

Biological neural systems exhibit significant heterogeneity in the temporal dynamics of their nodes, with neurons operating on a wide range of intrinsic timescales. In contrast, most artificial recurrent networks, including those in the reservoir computing (RC) framework, use homogeneous decay parameters across units. In this work, we investigate whether introducing decay heterogeneity improves task performance in Echo State Networks (ESNs) and Distance-based Delay Networks (DDNs). We evaluate four heterogeneity configurations:homogeneous ~~network~~, homogeneous clusterS, heterogeneous ~~network~~, and heterogeneous clusterS, across two benchmark tasks: NARMA-30 and Mackey-Glass. Networks are optimized using Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and evaluated on task performance,. We also studied the dynamical stability (Lyapunov exponents), representational dimensionality, and linear memory capacity of these networks. Our results show that DDNs consistently outperform ESNs, with homogeneous DDNs achieving the highest task performance. Timescale heterogeneity enhances memory flexibility in DDNs without compromising stability, while ESNs remain relatively unaffected. These findings suggest that structured heterogeneity in intrinsic timescales significantly improves the computational capabilities of delay-based recurrent systems.

time constant?
decay time constant?

Introduction

Biological networks nodes with heterogeneous properties

Physical systems such as brains consist of ~~heterogeneous neurons~~; not just in terms of morphology and molecular composition, but also in their temporal properties Koch and Laurent (1999); Habashy et al. (2024). In some cases, temporal differences among neuron types span several orders of magnitude London et al. (2010); Brunel and Wang (2003); Wang et al. (2020). This diversity serves multiple functional roles, including supporting motor control Cavanagh et al. (2020), preserving remote memories Runyan et al. (2017b), and enabling spatial memory formation McNaughton et al. (2006). In particular, variation in neuronal time constants and decay dynamics is thought to be critical for temporal processing and memory formation in the brain Hasson et al. (2008); Chu et al. (2020).

In contrast, artificial neural networks, especially recurrent neural networks and their variants like the reservoir computing (RC) framework, are typically designed with homogeneous units that share identical intrinsic parameters. RC is a powerful approach for modeling temporal sequences Lukoševičius and Jaeger (2009) using high-dimensional dynamical systems such as Echo State Networks (ESNs) Jaeger (2001b). These networks rely on a fixed, randomly connected recurrent reservoir, whose states are linearly read out ~~combined~~ by a trainable readout layer. This setup simplifies training while maintaining a strong performance on memory-intensive and nonlinear tasks. A key hyperparameter in RC systems is the leak or decay rate, which controls how much past information is taken into consideration in each unit, shaping the network's temporal memory Schrauwen et al. (2007).

Recently, an extension of ESNs called Distance-based Delay Networks (DDNs) Iacob et al. (2022) has been introduced. DDNs incorporate non-uniform delays between reservoir units, motivated by the fact that axonal delays are ubiquitous in the brain and that information in physical systems like the brain is constrained by

of the reservoir neurons

Traditionally, or originally

heterogeneous

the spatial structure and finite signal transmission speed Izhikevich (2006). Prior work has shown that these delays lead to a redistribution of memory across timescales and can improve task performance compared to standard ESNs Iacob et al. (2022); Soriano et al. (2014); ?.

However, while the role of inter-unit delays has been explored, it remains unclear whether heterogeneity in intrinsic parameters, specifically, decay rates-can further enhance performance in both ESNs and DDNs. Most prior studies assume a single, homogeneous decay parameter across all units, effectively enforcing a single timescale on the reservoir. This may limit the system's ability to capture rich, multi-timescale dynamics, especially in tasks that demand both, short-and long-term memory. In contrast, biological neural systems inherently operate with a range of time constants London et al. (2010); Nam et al. (2017); Zhou et al. (2023), suggesting that incorporating decay heterogeneity could yield computational benefits.

To investigate this, we design and analyze two sets of reservoir networks, ESNs and DDNs, each implemented in four different configurations. These include a homogeneous setup where all units share a fixed decay value, a fixed-cluster configuration where the reservoir is divided into modules, each with its own fixed decay; a network-heterogeneous configuration where each unit's decay is sampled independently from a common distribution; and a cluster-heterogeneous configuration where different modules sample their decays from different distributions. We evaluate all configurations on two standard benchmark tasks: the NARMA-30 task, which tests nonlinear memory capacity, and the Mackey-Glass task, which involves chaotic time-series prediction. Performance is optimized using a Covariance Matrix Adaptation Evolution Strategy (CMA-ES), with task evaluation based on the normalized root mean square error (NRMSE) for NARMA and the prediction horizon for Mackey-Glass.

Beyond performance, we analyze the networks' internal dynamics to understand the mechanisms underlying any observed differences. We compute Lyapunov exponents to assess dynamical stability and chaotic behavior, measure representational dimensionality using SVD-based metrics, and evaluate linear memory capacity. Our results show that DDNs consistently outperform ESNs across all heterogeneity types, with fixed-cluster DDNs performing best. These findings suggest that structured temporal heterogeneity, particularly when organized in modular clusters, significantly enhances the computational capabilities of reservoir networks for tasks involving memory and temporal dynamics. ~~But the heterogniety requirement is task dependent.~~

Methods

explain first readout layer, ref to fig1

Network Design

Echo State networks

ref based on...

Echo state networks (ESN) are essentially recurrent networks with fixed weights and randomly connected units, given by the following equation:

$$x(n+1) = (1-\alpha)x(n) + \alpha f(W_{res}x(n) + b_{res} + W_{in}v(n)), \tag{4.1}$$

where x B.. α is .. ρ is.. explain variables!

Contrary to RNNS, only the linear readout layer is trained for an ESN in order to optimize it for a task. In the given formulation above, x(n) represents the state of the reservoir at time step n. The size of the reservoir is given by the number of units N. W_{res} represents the $N \cdot N$ weight matrix of the reservoir. Symbol W_{in} represents a $N_{in} \cdot N$ input weight matrix. b_{res} represents the bias of the reservoir nodes, ~~reservoir weights of size~~ ~~N. The symbol~~ $v(n)$ is the input to the reservoir at time step n, α is the leak parameter, which decides the importance of previous states for the current state, ~~this acting as memory~~. This leak parameter can be fixed or ~~distinct~~ variable for each unit. $f(\cdot)$ is the non-linear activation function, which is usually sigmoid or hyperbolic tangent.

explain

thereby acting as a memory decay parameter.

Distance based delay networks

both capital or not

The architecture of Distance based delay networks are similar to ESNs, except for the introduction of delays. The delays are implemented by assuming that each unit lies on a 2-D Euclidean ~~space~~ plane and the distance between two units determines the delay between them. A distance matrix D is computed, where each element D_{ij} represents the distance between units i and j scaled by the signal propagation velocity and simulation timestep. A new masked weight matrix $W^{res}_{D=d}$ is created accounting

ref based on ...

for delays, each element in this matrix is given by the following equation:

what method?

where

$$W_{i,j,D=d} = \delta_{i,D_{ij}} \cdot W_{i,j} \tag{4.2}$$

then

here, d is $\in [1, D_{max}]$ and δ is the Kronecker delta operator. D_{max} is the maximum value the delay can take. This method allows for all the weights other than at a delay of d to be zero. The final update equation for DDNs is given by:

$$i(n) = \sum_{d=0}^{D_{max}} \left(W^{res}_{D=d} x(n-d) + W^{in}_{D=d} v(n-d) \right) + b_{res} \tag{4.3}$$

$$x(n) = (1-\alpha)x(n-1) + \alpha f(i(n)) \tag{4.4}$$

that need/can

All the parameters in the above set of equations are identical except for the addition of delays. As described in Iacob et al. (2022), the delays between individual units are treated as hyperparameters to be optimized. The coordinates are sampled using a Gaussian Mixture model, with K clusters, instead of finding the optimal coordinates for each unit, the GMM model is optimized to find the fitting distribution for each cluster.

of the neurons are initially

Task

NARMA-30

consistency

The Nonlinear auto-regressive moving average (NARMA) is a popular benchmark task used to evaluate the performance of a RC system. It is a system of equation that is highly non-linear and has clear temporal dependencies, defined by the order parameter. The general system of equation is given by the following equation:

$$y(t+1) = c_1 \cdot y(t) + c_2 \cdot \sum_{i=0}^{p} y(t-i) + c_3 \cdot u(t-p)u(t) + c_4 \tag{4.5}$$

p?, give symbol

which task? or when?

Where $c_1 - c_4$ are task parameters, and p is the NARMA order. $y(t)$ is the state of the system at time t, $u(t)$ is the input to the system at time t, which is uniformly sampled between 0.0 and 0.5. Essentially, the task entails predicting the next state given the serialized previous states, we used the NARMA-30 task to train our networks, the parameters are $p = 29$, $c_1 = 0.2$, $c_2 = 0.004$, $c_3 = 1.5$, and $c_4 = 0.1$.

not correct sentence

Evaluation

the consistency

We evaluated the NARMA-30 task using Normalized root mean squared error (NRMSE) given as follow:

$$NRMSE = \frac{1}{\bar{y}}\sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{n}} \tag{4.6}$$

where $\bar{y}$ is the mean of the ground truth, y_i and $\hat{y}_i$ are ground truth and network readout respectively, n is the total number of elements in the time series.

and

Mackey-Glass

why now t and not n?

The discretized Mackey-Glass timeseries is given by the following equation:

$$x(t+1) = x(t) + \beta\frac{x(t-\tau)}{1 + x(t-\tau)^n} - \gamma x(t) \tag{4.7}$$

are / were chosen as follows

The parameters ~~the following~~ $\tau = 17$, $n = 10$, $\beta = 0.2$ and $\gamma = 0.1$. The aim of this task is to accurately generate the Mackey-Glass sequence as long as possible using the previous step and not receiving any external input or correction. This is done by training the readout layers of the networks for one step ahead prediction. The performance is measured by a metric called the prediction horizon, which measures how many time steps into the future can be predicted by the network accurately within an error margin. ~~The error margins are~~ $\pm 0.1\sigma_l^2$, where σ_l^2 is the label variance.

what task? & when?

which is given by

double

Evaluation

the

We evaluated the network performance using prediction horizon, which the number of time steps the network can predict the time series within the error margins.

so what are the error margins?

Training

CMA-ES Optimization

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) Hansen et al. (2006) is a widely used evolutionary algorithm for hyperparameter optimization, especially effective for non-convex and high-dimensional problems. CMA-ES consists of a multivariate Gaussian search distribution defined by a mean vector, stepsize, and covariance matrix. At each iteration, it samples a population of candidate solutions from this distribution, evaluates their fitness, and updates the mean and covariance matrix to sample future samples from more fitting regions of the search space. The mean vector represents the current best candidate based on the population's progress, while the covariance matrix spans the space of the search distribution. This iterative process enables CMA-ES to efficiently explore complex solution spaces and is robust due to its invariance properties and ability to handle parallel evaluations. In case of ESN and DDN optimization, the mean vectors are the hyperparameters of DDNs and ESNs. Initially, all hyperparameters are roughly made to fall in the same range, after every iteration, a new hyperparameter vector is generated based on the evolution of the multi-variate distribution. Using this new solution vector, networks are generated and are used to perform validation scores. Both ESN and DDN optimizations are run for 200 generations with a population size of 20 for Mackey Glass task and population size 25 for NARMA 30 task.

to what?

Give params & initial ranges!

Readout training

The randomly sampled network is not directly able to perform the task, ~~it is~~ essentially it acts as a temporal kernel that increases the dimensionality of the input and thus increases the linear separability, . The the activity from a network driven by a random input can be recorded and can be treated as a feature matrix X of size $N \times T$, where N is the number of neurons and T is the number of time steps of the simulation. The output labels $\hat{y}$ and and the corresponding input vector $v(n)$ are ordered vectors of size T. The objective function J given by the following equation below is solved using ridge regression trained on network activities:

$||_2$ -> L_2 norm?

$$J(W_{readout}) = ||XW_{readout} - \hat{y}||_2^2 + \lambda||W_{readout}||_2^2 \tag{4.8}$$

Optimizing the above equation gives the $W_{readout}$, where λ is the regularizing parameter and $W_{readout}$ is the readout.

Initialization $W_{readout}$?

Lyapunov exponent

the Dynamics of the RC system can be studied by estimating the Lyapunov exponent of the network trajectory, . W we calculated the Lyapunov exponent of ESNs and DDNs using a numerical method, explained by Boedecker et al. (2012). the Lyapunov exponent is given by the following equation:

$$\lambda = \lim_{t \to +\infty} ln\left(\frac{\gamma_t}{\gamma_0}\right) \tag{4.9}$$

where γ_t & γ_0 are n -

The steps involved in the numerical calculation as described by Boedecker et al. (2012) is as follows:

1. Two identical networks X_1 and X_2 were created taking the best sampled networks?, . A a reservoir unit from one of the network X_1 was given a small pertur-

Explain what γ is, how calculated!

bation and the network X_2 was kept as is. The initial $L2$ normalized distance between X_1 and X_2 is therefore γ_0.

2. The network is driven by the input that the ~~respective~~ network is trained on, that is either NARMA-30 or Mackey-Glass inputs. At each simulation step t a normalized $L2$ distance between networks $\gamma_t = ||X1(t) - X2(t)||^2$ is calculated.

3. The perturbed network X_1 is reset to the initial distance γ_0, $X_1 \rightarrow X_2(t) + (\gamma_0/\gamma_t)(X_2(t) - X_1(t))$ once it crosses the upper or lower thresholds, i.e, $\gamma_t \notin [1e^{-5}, 1e^{-14}]$, this step is essential to prevent numerical overflows.

4. The simulation is run for 200 steps, and is repeated for multiple perturbation intensities ranging from $1e^{-1}$ to $1e^{-9}$. Finally the Lyapunov exponent is given by ~~the~~ equation 4.9.

The above steps are repeated for all the networks for both the tasks. All the numerical simulations were performed using python and Tensorflow 1.x.

Dimensionality

The dimensionality of a given network while performing a task was calculated using Singular Vector decomposition of the activity matrix and counting the top singular values that explain~~s~~ 99% of the variance. The SVD is given as follows:

$$A = U\Sigma V^T \tag{4.10}$$

where U of and V^T are left and right singular matrices of dimensions $N \times N$ and $T \times T$ respectively, Σ is the diagonal matrix of dimensions $N \times T$. The dimensionality is given by:

$$dim = \frac{\sum_{i=1}^{n} \sigma_{i,i}}{\sum_{i=1}^{N} \sigma_{i,i}}; n < N \tag{4.11}$$

where $\sigma_{i,i}$ are the diagonal elements of the singular value matrix Σ, N is the number of reservoir units in ESNs and DDNs.

Linear Memory Capacity

Linear Memory Capacity measures short term memory distribution of a dynamical system Jaeger (2001a). In order to calculate the linear memory capacity, the ENSs and DDNs are driven with a uniform random input and the readout layer is trained to reproduce the input from l time lag ago. The linear memory capacity at a specific lag l shows how well the reproduction by a network correlates with the input l time steps ago. the Total memory capacity is given by summing all the memory capacity ies for each lag. The idea is formulated as follows:

$$MC_l = r(u(n-l), \hat{u}(n))^2 \tag{4.12}$$

$$MC_{total} = \sum_{l=1}^{\infty} MC_l \tag{4.13}$$

Where $r(\cdot)$ is the Pearson correlation, $u(n)$ is the input at timestep n and $\hat{u}(n)$ is the readout reproduction of the input at timestep $n-l$.

Maybe add a time line with CMAES optimization & evaluation of NMSE, prediction horizon, etc

Results

The aim of this study was to study the effect of intrinsic heterogeneity on task performance of ESNs and DDNs. ~~The network design and dynamics are explained in detail in Methods section 5.1. For the stated aim~~ Therefore, we designed 4 heterogeneity/homogeneity configurations ~~explained~~ as follows:

1. **Homogeneous Network**: All units share a single decay parameter.

2. **Homogeneous Cluster**: The reservoir is divided into clusters, each with its own fixed decay parameter.

3. **Heterogeneous Network**: Each unit samples its decay from a shared distribution.

4. **Heterogeneous Cluster**: Each cluster samples decay parameters from different distributions.

These configurations are tested on two benchmark tasks namely NARMA-30 and Mackey-Glass tasks (see Methods section 5.2):

- **NARMA-30**: A nonlinear auto-regressive moving average task designed to test long-range memory and nonlinear dynamics. Parameters for the NARMA-30 task ~~is~~ are given in Methods section 5.2.1.

- **Mackey-Glass**: A chaotic time-series prediction task that evaluates a network's ability to generate stable yet complex temporal outputs. Parameters for the Mackey-Glass task ~~is~~ are given in Methods section 5.2.2.

A key reason to optimize ee networks for these two tasks is to understand if inherent heterogeneity is utilized based on task performance. We optimized all networks using the CMA-ES learning algorithm (see Method section 5.3). We evaluated the hyperparameters of

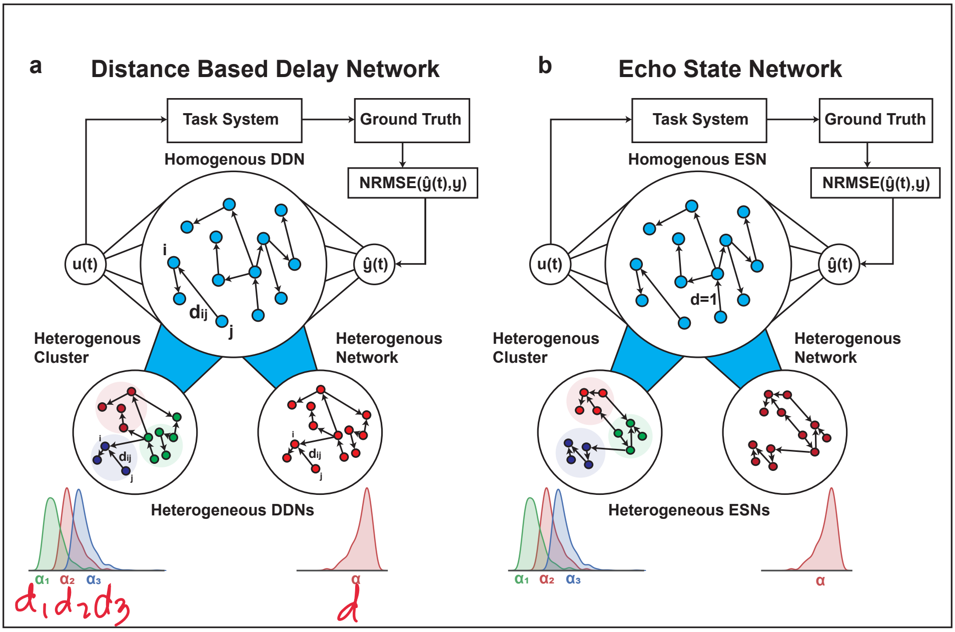


Figure 4.1. **Reservoir architecture and heterogeneity design in DDNs and ESNs: (a)** DDNs were made heterogeneous by introducing heterogeniety in delays (*α*), C cluster heterogeneity is designed by sampling delays from different distributions for each cluster, d S similarly, network heterogeneity is designed by sampling delays from a single distribution. **(b)** same heterogeneity designed for ESNs. It ~~can also be seen that the key difference between ESNs and DDNs are~~ is ~~the introduction of delays between units in case of DDNs.~~

the task performance for NARMA-30 and Mackey-Glass tasks using NRMSE and prediction horizon measures respectively (see Method section 5.1).

Task Performance — check tenses!

NARMA 30

In order to study the effect of heterogeneity in leak parameters in ESNs and DDNs, we evaluated the networks using their validation scores for each generation of the CMA-ES algorithm ~~shown in Fig. 4.2a~~ while these networks learn to perform NARMA-30 task (see Methods). ~~It can be seen~~ that as the networks are optimized over generations, the ~~validation scores~~ errors ~~lower~~ decrease for both ESNs and DDNs, but the NRMSE

In Fig 4.2a we show that

parameter

scores for DDNs, especially DDNs with a network wide fixed decay, a per cluster distributed decay and a per cluster fixed decay are much lower than ESNs. It can also be seen that all ESNs eventually reached ~~are optimized to~~ similar NRMSE scores. Interestingly, it ~~can be seen that~~ DDNs with per-cluster fixed decay out~~-~~perform, DDNs with higher levels of heterogeneity, and ~~while~~ the DDNs with network wide distributed decay parameters show~~ed~~ a similar performance to ~~as~~ the ESNs. We also evaluated the performance of the best optimized networks on a test set ~~shown in~~ Fig. 4.2b, it can be seen that DDNs with a network wide fixed decay ha~~s~~ a lower NRMSE score than ESNs. For per-cluster case, the performance of DDNs and ESNs are indistinguishable.

Conclusion

Mackey-Glass

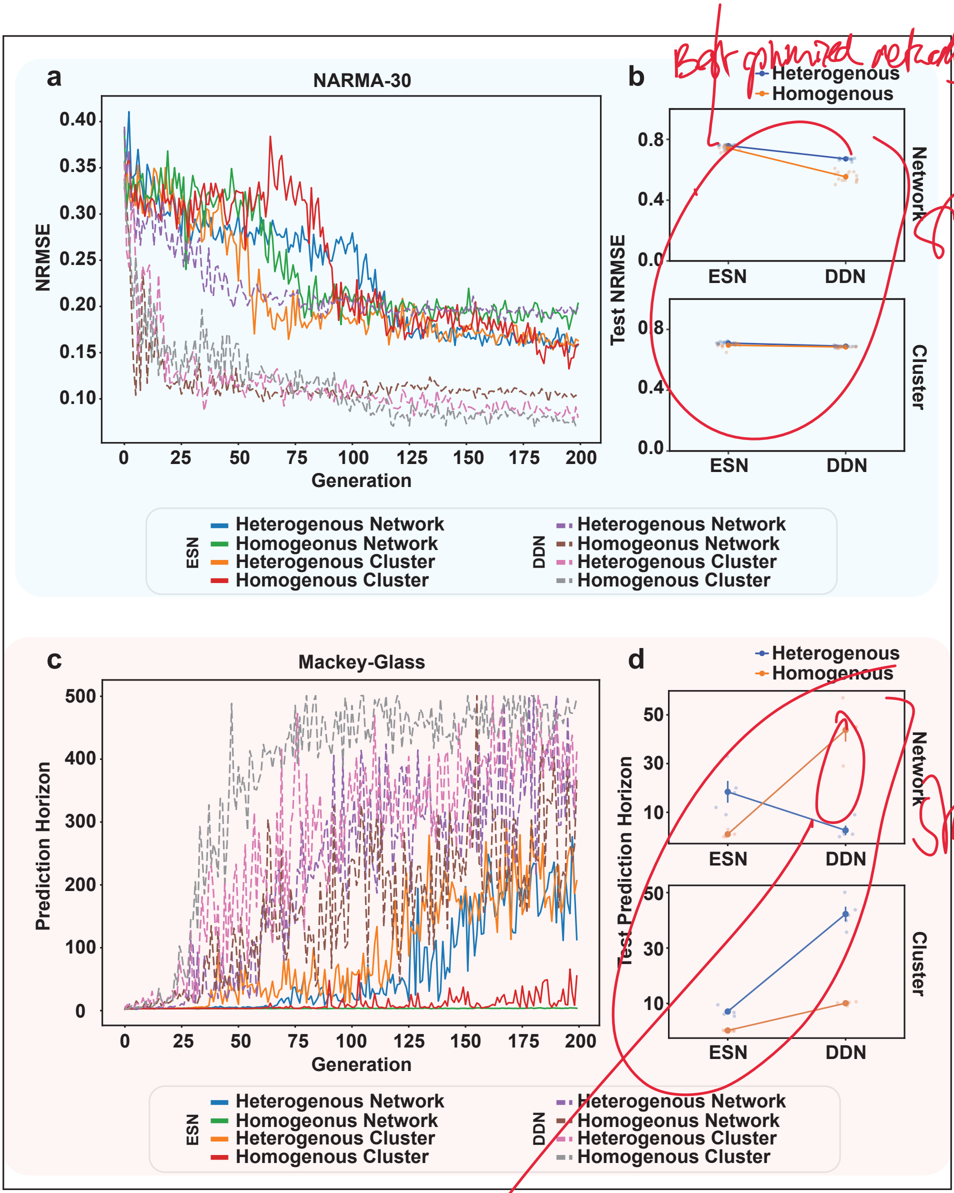
Goal of this section

Similar to the NARMA-30 task we optimized ESNs and DDNs with varying levels of heterogeneity for Mackey-Glass time series prediction task (see Methods section). We ~~we~~ evaluated the performance of the networks using prediction horizon metric (see Methods section). ~~it can be seen from~~ Fig. we show Fig. 4.2b that prediction horizons for validation runs over generation of CMA-ES optimization for DDNs are consistently higher than the ESNs. It can also be seen that DDNs with network wide distributed decay perform~~s~~ the best among all the DDNs, reaching a ~~the~~ prediction horizon of 500 steps. It is also noticeable that ESNs with per cluster distributed decay and network wide fixed decay perform the worse compared to all other variants. ESNs with network wide distributed decay params and per cluster fixed decay params perform objectively better, than ...? reaching a prediction horizon value of 300 steps. Similar to the NARMA-30 taks, we evaluated the performance of the best optimized networks on a test set ~~shown in~~ Fig. 4.2b, it can be seen that DDNs with a network wide fixed decay has ~~lower~~ value NRMSE score ~~than ESNs~~ the lowest than ESNs.

Phrase of

grammar?

Last generation!,



Best optimized networks

Stats?

Stats?

Figure 4.2. **Network performance for DDNs and ESNs with or without heterogeneity**: **(a)** ~~Line plot~~ shows the perforamce in terms of NRMSE over generations during CMA-ES optimization for DDNs and ESNs during NARMA 30 task.

Striking: ← for DDN the heterogeneity decreases performance!, the best performing networks

(b) Test NRMSE for different network configurations for both ESNs and DDNs. It can be seen that DDNs with/without heterogeneity perform better than ESNs. **(c)** Performance of ESNs and DDNs with or without heterogeneity optimized for Mackey-Glass prediction task over generations of CMA-ES optimization. **(d)** Test performance of ESNs and DDNs optimized for Mackey-Glass timeseries prediction task, separated based on heterogeneity types.

Stability and Dimensionality

Maximum Lyapunov exponents

In order to evaluate the advantage of intrinsic heterogeneity on ESNs and DDNs we calculated the largest lyapunov exponents of the sampled best networks while they perform the task (see Methods section). This was done in order to examine the stability of the network dynamics while the networks perform the task, we hypothesized that intrinsic decay heterogeneity might cause the networks to be more chaotic than their homogeneous counterparts. We evaluated the Maximum Lyapunov exponents for each network and heterogeneity types for different perturbation magnitudes ranging from $1e^{-1}$ to $1e^{-9}$. We perform this analysis for NARMA-30 and Mackey-Glass tasks separately, the results are summarized in **Fig. 4.3a-b**.

the of

For NARMA-30 task (**Fig. 4.3a**), it can be seen that for each perturbation magnitude, the largest lyapunov exponent are negative for all ESNs variants and are lower than DDNs, suggesting that ESNs have more contracting dynamic than DDNs despite the heterogeneity. The outlier to this observation is the ESN with network wide fixed delay, for which the lyapunov exponent is negative but invariant to the perturbation magnitude and comparable to DDNs. It is also important to observe that for all perturbation magnitudes, the highest perturbation yields the lowest exponent. In case of DDNs, the per cluster fixed decay configuration the maximum lyapunov exponent was invariant to the perturbation magnitude. The

? Expand?

clear x on fig..

(clear)

MLE for DNNs with network wide distributed decay and per cluster distributed decay were comparable, on the other hand, DDNs with network wide fixed decay shows a positive MLE for highest magnitude perturbation ($1e^{-1}$) suggesting chaotic dynamics.

the these those of

For Mackey-Glass timeseries prediction task (**Fig. 4.3b**), it can be seen that for each perturbation magnitude, the largest lyapunov exponent are is negative for all ESNs variants and are lower than DDNs, suggesting that ESNs have more contracting dynamic than DDNs for this task as well, despite the heterogeneity. ? It is also important to observe that for all perturbation magnitudes, the highest perturbation yields the lowest exponent just as in case of NARMA-30. The ESNs with network wide distributed decay were found to have the highest MLE among ESNs and the ESNs with per cluster fixed decay were found to have the lowest MLE. In case of DDNs, the per cluster fixed decay configuration shows a positive exponent for the magnitude of $1e^{-}9$ and $1e^{-}7$. The MLE for DNNs with all other variants had comparable negative MLE for all each perturbation magnitude suggesting a stable contracting dynamic.

Conclusion about relation performance & LE

Dimensionality

In order to further evaluate the usefulness of intrinsic heterogeneity in terms of recruiting reservoir units for performing the a task, we calculated the dimensionality of the network using the network states while the network is driven by the task input.

achieving during which was

We hypothesized that intrinsic decay heterogeneity might increase the dimensionality of the network, improving the task performance. We drove the network with NARMA-30 and Mackey-Glass input for 2000 steps, and used the network state after a 300 steps of warm-up period. For NARMA-30 task, we observed a low dimensionality for all networks, the highest for ESN with fixed decay per cluster (D=15) and lowest for DDNs with network wide fixed decay (D=8), dis-

the dimensionality was obtained a

the a

again: so show relation

tributed decay per cluster (D=8) and fixed decay per cluster (D=8). Similarly, for the Mackey-glass time series prediction task, we observe a low dimensionality for all networks except for ESNs with a distributed decay per cluster (D=98). The network with the lowest dimension was only found to be the DDN~~s~~ with a fixed decay per cluster. Overall, this shows that both ESNs and DDNs do not recruit a high number of units in order to perform both NARMA-30 and Mackey-glass tasks.

active? nodes, dimensions

Linear Memory capacity

split sentence

In order to understand if intrinsic decay heterogeneity has an effect on the memory of ESNs and DDNs, especially to see if networks with heterogeneous decay parameters change the way networks store temporal information, especially the number of time steps these networks can recall information from, we calculated the linear memory capacity (see methods section). We used ~~the~~ a maximum delay of 40 steps. The observed linear memory capacity for each network variant is summarized ~~on~~ in **Fig. 4.4**.

and ref!

For the NARMA-30 task (**Fig. 4.4 left**), it can be seen that every ESN variant has the capacity to recall from upto 20 time steps ago and the s capacity is distributed over all the steps between 0 and 20. We do not observe a stark improvement in the memory capacity as a result of heterogeneity. On the other hand, DDNs show a much more nuanced difference in terms of memory distribution for NARMA-30 task. DDNs with a fixed decay per cluster show a concentration of memory capacity at 3 different ~~time steps,~~ lags while DDNs with a distributed decay per cluster show a prominent concentration at two ~~delay steps.~~ lags It is ~~also clearly~~ comparable observed that DDNs do not utilize the ~~time window~~ lags closer to the input, creating a except for DDNs with network wide fixed decay. This suggests that ESNs have a limited mobility to recall the past steps in order to perfrom the task, and decay heterogeneity doesn't provide any improvement on this. While, un DDNs scatter the capacity more sparsely, ~~thus~~ thereby

??

?

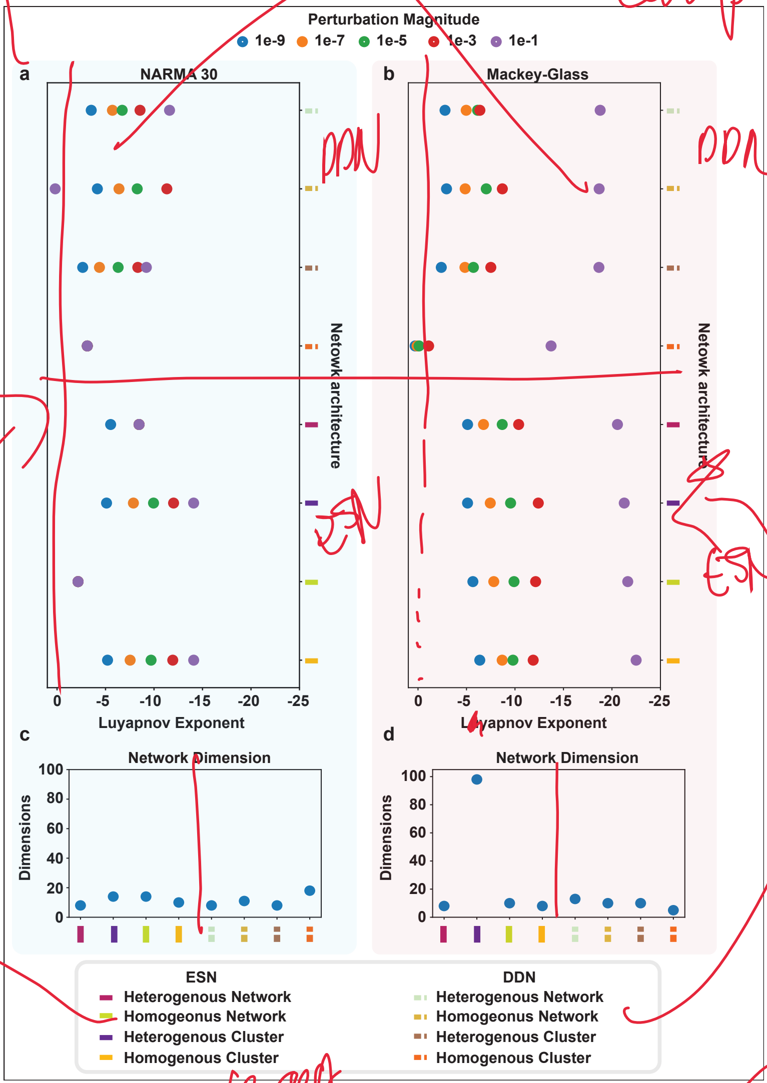
on the other hand can

where is the input?

what does this mean?

correlate with performance?

difficult to compare this way



target

task

Figure 4.3. **Dynamics based on Lyapunov exponent and dimensionality of optimized networks: (a-b)** ~~shows~~ must the Lyapunov exponents of the optimized networks while performing the NARMA 30 (blue background) and Mackey-Glass (red background). **(c-d)** ~~shows~~ ant the network dimensionality while performing the NARMA 30 (blue background) and Mackey-Glass timeseries prediction tasks (red background).

Only single value? No error bars or multiple values? Doesn't mean much then...

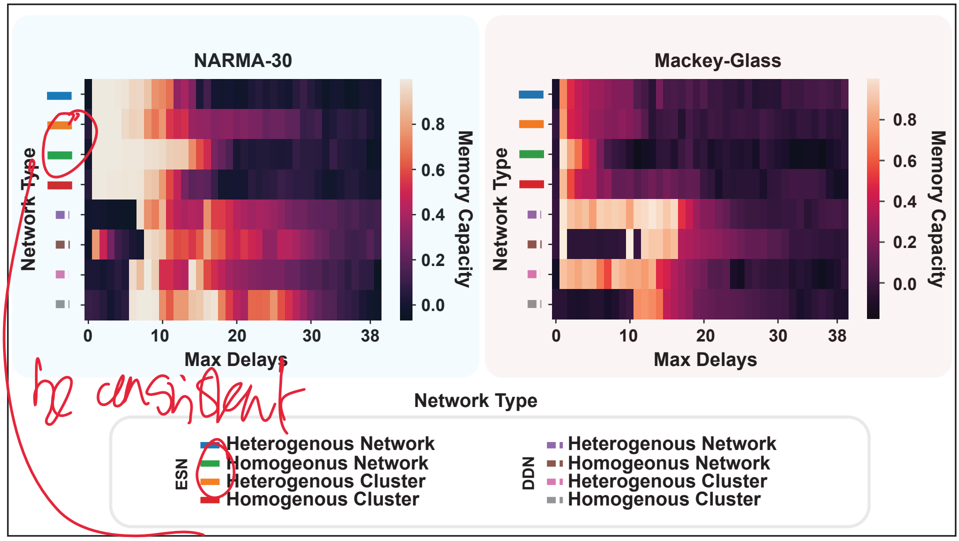


Figure 4.4. **Linear Memory Capacity of optimized networks**: **(Left)** Linear memory capacity of the networks optimized for NARMA 30 task for different delays. **(Right)** Linear memory capacity of the networks optimized for Mackey-Glass time series prediction task for different delays.

utilizing the past states more strategically.

For the Mackey-Glass time series prediction task (**Fig. 4.4 left**), it can be seen that every ESN variant ustilizes just a few past states from when the input is received, most of the capacity is recruited from just the previous step, we do not observe a stark improvement in the memory capacity as a result of heterogeneity for the case of Mackey-Glass as well. On the other hand, DDNs show a much more variability, DDNs with network wide distributed decay and DDNs with distributed decay per cluster, show a wide spread of memory capacity over multiple delay steps, while DDNs with fixed values show a concentration of capacity around 10 time steps ago.

In conclusion ~~This goes to show that~~ while for ESNs, the decay heterogeneity doesn't change drastically, for DDNs, the memory capacity is altered due to decay heterogeneity, therefore aiding in task performance. explain

with...

by...

than traditional ESNs without delay,

we... were not [illegible]!

Discussion

This study aimed at studying the effect of timescale (decay parameter) heterogeneity of individual units in reservoir networks on their computational performance, particularly in ESNs and DDNs. Set across two benchmark tasks, namely, NARMA-30 and Mackey-Glass, and across four heterogeneity configurations. We found that across all heterogeneity configurations, DDNs outperform ESNs in both tasks. This ~~confirms~~ is in agreement with previous work ~~showing~~ that showed that delay-based architectures are better suited for memory-intensive tasks (Iacob et al. (2022)), ~~this can be because~~ as the introduction of time delays enables more flexible temporal dynamics. ~~even more so than inherent decay, rendering inherent decay redundant~~. We observed ~~that~~ for ESNs trained on the NARMA-30 task, the homogeneous cluster configurations performs the best. ~~But not~~ However, [illegible] for the Mackey-Glass task, ~~where~~ as ESNs with Heterogeneous clusters shows the best performance. Surprisingly, homogeneous network DDNs outperform heterogeneous network ESNs, suggesting that architectural delay mechanisms alone offer substantial computational benefits. Thus ~~extending~~ extends the hypothesis of Tanaka et al. (2022) ~~that~~ who claimed that diverse timescales enhance network performance in both DDNs and ESNs.

The ultimate best-performing configuration across both tasks ~~was~~ were the homogeneous cluster DDNs. This suggests that structured, modular heterogeneity in decay parameters is more beneficial than randomly distributed heterogeneity. One likely reason is that this setup creates functionally specialized subpopulations with different integration windows, which are quite common in biological neural circuits. ~~It~~ Such clusters allows the reservoir to simultaneously retain short and long term dependencies spread across multiple units. These results support the hypothesis rooted in neuroscience that the diversity of timescales is functionally beneficial for temporal processing and memory (Lundqvist et al. (2016); Runyan et al. (2017a); Ca-

do you mean delay or decay? Not clear

Contradiction?

vanagh et al. (2022).

Contrary to our initial hypothesis, networks with decay heterogeneity did not show increased instability. In fact, all ESNs remained in a contracting regime, with strongly negative Lyapunov exponents across perturbation magnitudes. DDNs were more variable, with some configurations (e.g., homogeneous network) approaching neutral or even slightly positive exponents under high perturbations. This suggests that DDNs trade off stability for richer dynamics, and that heterogeneity does not necessarily lead to chaotic or unstable behavior. Interestingly, for all network types, the largest perturbation magnitude yielded the most negative Lyapunov exponent. This is likely due to saturation or nonlinear contraction in state space, and highlights the importance of evaluating stability in the local (infinitesimal) regime to interpret true Lyapunov behavior.

We hypothesized that decay heterogeneity would increase the effective dimensionality of the network enabling richer representations. This was not supported by our results. Most networks exhibited relatively low-dimensional dynamics, with little difference between configurations. This could mean that the tasks at hand do not require high-dimensional embeddings, or that the reservoirs learn to compress relevant dynamics efficiently regardless of heterogeneity. It may also suggest that stability and memory, not dimensionality, are the primary contributors to performance in this setting.

relate to Gao & Ganguli, Neural Task complexity

Linear memory capacity analysis revealed a more nuanced story. While ESNs showed limited changes across heterogeneity types, DDNs exhibited significant differences in how memory was distributed. Notably, DDNs with distributed decay showed a wider memory spread, while fixed-decay variants concentrated memory at specific lags. This suggests that decay heterogeneity allows DDNs to strategically allocate memory, a critical property for tasks like NARMA, where both recent and delayed inputs matter. This behavior was absent in ESNs, which tended

between what & what?

Than what?

to have more rigid, shallow memory profiles regardless of heterogeneity.

This study is limited to two benchmark tasks and a specific class of recurrent models. Future work should explore: Broader tasks, including those involving categorical sequence prediction or noisy real-world data, other forms of heterogeneity (e.g., in connectivity, input weights), or... how decay heterogeneity interacts with delay heterogeneity under more complex learning rules or online adaptation. It would also be valuable to explore the Information processing capacity of these systems, and how dimensionality evolves over time or under different input statistics.

make less sweeping but more clear

which? contradiction?

Our results suggest that introducing modular heterogeneity in decay dynamics can meaningfully improve performance and memory organization in reservoir networks, particularly when paired with architectural delays. From a neuroscience perspective, this supports the idea that diverse time constants in biological systems are not incidental, but serve computational roles such as memory stratification and temporal coding.

?? ?? to what?

From an engineering viewpoint, these findings argue for designing non-uniform, structured reservoirs in artificial systems, especially for tasks involving long-~~range~~ timescale? dependencies or multi-timescale dynamics. Rather than tuning a single decay or leak parameter, practitioners could benefit from multi-cluster architectures with targeted decay profiles.

or delay

References

Boedecker, J., Obst, O., Lizier, J. T., Mayer, N. M., and Asada, M. (2012). Information processing in echo state networks at the edge of chaos. *Theory in Biosciences*, 131:205–213.

Brunel, N. and Wang, X.-J. (2003). Firing frequency, spike timing and synaptic plasticity of the hippocampal neurons: modeling and experiments. *Journal of Neurophysiology*, 90(1):415–430.

Cavanagh, S., Towers, J., Wallis, J., Hunt, L., and Kennerley, S. (2022). Reconciling persistent and dynamic hypotheses of working memory coding in prefrontal cortex. *Nature Communications*, 13:1–18.

Cavanagh, S. E., Hunt, L. T., and Kennerley, S. W. (2020). A diversity of intrinsic timescales underlie neural computations. *Frontiers in neural circuits*, 14:615626.

Chu, J. C., Wang, Q., Baldassano, C., and Guo, N. D. (2020). Long-timescale processing in human auditory and visual cortex supports short-term memory. *Nature Neuroscience*, 23:1686–1695.

Habashy, K. G., Evans, B. D., Goodman, D. F., and Bowers, J. S. (2024). Adapting to time: Why nature may have evolved a diverse set of neurons. *PLOS Computational Biology*, 20(12):e1012673.

Hansen, N., Müller, S. D., and Koumoutsakos, P. (2006). The cma evolution strategy: A comparing review. In Lozano, J., Larrañaga, P., Inza, I., and Bengoetxea, E., editors, *Towards a New Evolutionary Computation. Advances in the Estimation of Distribution Algorithms*, pages 75–102. Springer.

Hasson, U., Yang, E., Vallines, I., Heeger, D., and Rubin, N. (2008). A hierarchy of temporal receptive windows in human cortex. *Journal of Neuroscience*, 28(10):2539–2550.

Iacob, S., Freiberger, M., and Dambre, J. (2022). Distance-based delays in echo state networks. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 211–222. Springer.

Izhikevich, E. M. (2006). Polychronization: Computation with spikes. *Neural Computation*, 18(2):245–282.

Jaeger, H. (2001a). Short term memory in echo state networks.

Jaeger, H. (2001b). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *GMD Technical Report 148*.

Koch, C. and Laurent, G. (1999). Complexity and the nervous system. *Science*, 284(5411):96–98.

London, M., Roth, A., Beeren, L., Häusser, M., and Latham, P. E. (2010). Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. *Nature*, 466:123–127.

Lukoševičius, M. and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149.

Lundqvist, M., Rose, J., Herman, P., Brincat, S., Buschman, T., and Miller, E. (2016). Gamma and beta bursts underlie working memory. *Neuron*, 90(1):152–164.

McNaughton, B. L., Battaglia, F. P., Jensen, O., Moser, E. I., and Moser, M.-B. (2006). Path integration and the neural basis of the 'cognitive map'. *Nature Reviews Neuroscience*, 7(8):663–678.

Nam, L., Kang, R., Kim, D., and Marder, E. (2017). Diversity matters: Neural variability promotes learning and generalization. *Neuron*, 96(4):795–807.

Runyan, C., Piasini, E., Panzeri, S., and Harvey, C. (2017a). Distinct timescales of population coding across cortex. *Nature*, 548:92–96.

Runyan, C. A., Piasini, E., Panzeri, S., and Harvey, C. D. (2017b). Distinct timescales of population coding across cortex. *Nature*, 548(7665):92–96.

Schrauwen, B., Verstraeten, D., and Van Campenhout, J. (2007). An overview of reservoir computing: theory, applications and implementations. *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*.

Soriano, M. C., Ortín, S., Keuninckx, L., Appeltant, L., Danckaert, J., Pesquera, L., and Van der Sande, G. (2014). Delay-based reservoir computing: noise effects in a combined analog and digital implementation. *IEEE transactions on neural networks and learning systems*, 26(2):388–393.

Tanaka, G., Matsumori, T., Yoshida, H., and Aihara, K. (2022). Reservoir computing with diverse timescales for prediction of multiscale dynamics. *Physical Review Research*, 4(3):L032014.

Wang, P. X., Farzadfard, F., and Pehlevan, C. (2020). Heterogeneity of time constants improves memory performance in recurrent neural networks. *arXiv preprint arXiv:2012.13074*.

Zhou, W., Fiete, I., and Lampl, I. (2023). Temporal diversity in neural populations enhances robustness and memory capacity in recurrent networks. *Science Advances*, 9(10):eaaz8693.