TUNE: A NON-STATIONARY KERNEL PERSPECTIVE ON LINEAR ECHO STATE NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Echo State Networks are high-dimensional randomized recurrent neural networks where only the output layer is trained. It is common to apply a non-linear transfer function to the recurrent layer while training a linear perceptron at the output layer. Recently, it has been demonstrated that Echo State Networks, which rely on a linear recurrent layer followed by a non-linear output layer, are also performant. This design choice is attractive, as the resulting networks are fully parallelizable and can perform inference faster and more efficiently. In this study, we demonstrate that the response function of a finite-width linear Echo State Network is, in expectation over its input random projection matrix, a matrix-valued Gaussian Process that, depending on the topology of the recurrent layer, the induced kernel can be stationary, locally stationary, or non-stationary. Following this perspective, we show that tunability between the different statistical properties of Gaussian Processes provides insight into the dynamics of the recurrent layer and can lead to richer representations. Furthermore, it motivates a novel architecture: Tuneable and Unconstrained Non-stationary Echo State Network (TUNE). In this architecture, random features are drawn from a parameterizable spectral measure and combined with tuneable temporal weighting, replacing the recurrent dynamics. We show that TUNE admits a lower computational complexity of the hidden state initialization than linear Echo State Networks with dense recurrent connectivity, providing competitive performance on both synthetic and real-world datasets for classification and (long-term) forecasting.

1 INTRODUCTION

The modern machine learning landscape revolves largely around overparametrized deep networks. Transformers (Vaswani et al., 2017) are an example of the neural network that substantially improved the state-of-the-art across many application domains, where they largely overshadowed recurrent neural networks. Unlike the hidden state dynamics of recurrent neural networks with a non-linear transfer function, the attention mechanism is fully parallelizable, but its quadratic scaling leads to a computational bottleneck when modeling sequential data with long-term dependencies. Linearized recurrent neural networks, also known as linear state space models, have been shown to rival transformer's performance while only having linear scaling with sequence length. Architectures such as Legendre Memory Unit (LMU) (Voelker et al., 2019), Structured State Space Sequence Model (S4) (Gu et al., 2021), and Selective State Space Model (MAMBA) (Gu & Dao, 2024) demonstrate that it is possible to model the long-term dependencies without the attention mechanism.

An efficient and performant alternative to fully-trainable recurrent neural networks is the Echo State Network (ESNs) (Jaeger, 2001), a subclass of reservoir computing, which on specific tasks such as forecasting chaotic dynamical systems lies near the efficiency-performance Pareto front (Gilpin, 2023). The efficiency of training ESNs stems from their one-step training procedure (usually based on regularized least squares), but recurrence in ESNs with non-linear transfer function is slow since they cannot be fully parallelized. ESNs with linear transfer function, however, are optimal for long-term memorization (Verstraeten et al., 2010; Dambre et al., 2012) and can be parallelized because they are, in essence, randomized linear state space models, even if this correspondence is not frequently stated in the literature. ESNs with linear transfer function and non-linear readout layer (referred to as linear ESNs) have also been proven to be universal approximators for fading mem-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ory filters (Grigoryeva & Ortega, 2018), which suggests that linear ESNs might provide a basis for architectures alternative to the fully-trainable ones.

An important design choice that must be considered to optimize ESN performance is the topology of its recurrent layer (called reservoir) (Cui et al., 2012; Dale et al., 2021), which determines the hidden state dynamics. Common choices include fully random-, orthogonal-, permutation-, or locally-connected random matrices. The optimal reservoir topology is typically selected in an ad hoc manner and cannot, in general, be determined a priori (Gallicchio & Micheli, 2019). Consequently, in practice, it is often treated as a categorical hyperparameter.

To move beyond this discrete architectural choice, we provide a kernel perspective on general reservoir topology. Under random initialization of an input projection matrix, the ensemble of linear ESNs with reservoir size k induces a matrix-valued Gaussian Process (GP) over the random initialization. Unlike the infinite-width neural network GP limit (Lee et al., 2018), whose Gaussianity is a central-limit statement over infinitely many neurons, this correspondence holds at any fixed reservoir size. Because the randomness is generated by only finitely many random variables, the resulting GP is degenerate. We therefore use its induced covariance kernel as a reference and extend the analysis to a corresponding non-degenerate process.

The main contribution of this study is centered around the covariance kernel; we show that it splits into three terms: stationary (lag-dependent), locally stationary (a mid-point-dependent), and non-stationary (a commutator remainder controlled by the non-normality of the transfer matrix[1]). The split following from the kernel perspective inspires the proposed architecture, Tuneable and Unconstrained Non-stationary ESN (TUNE), which is illustrated in Figure 1. TUNE allows for explicit adjustment of the different statistical components of ESN response functions. The novel response function samples are drawn at once rather than computed iteratively via the transfer matrix, which reduces the asymptotic complexity in the reservoir size at a fixed number of random features compared to linear ESNs. Furthermore, the kernel perspective allows to classify common reservoir topologies as shown in Table 1. We evaluate TUNE in terms of test mean squared error (MSE), classification accuracy, optimization time, and energy consumption on a wide range of synthetic and real-world datasets including classification and forecasting tasks. We demonstrate that TUNE outperforms other ESN baselines in terms of test MSE and classification accuracy, while reducing the initialization time compared with iterated response function generation. TUNE is evaluated on a range of synthetic and real-world benchmarks. The Synthetic datasets include two novel datasets (Figure 3) as well as datasets based on five chaotic systems from dysts benchmark (Gilpin, 2021), and a dataset for measuring memory capacity (Appendix F.3). Real-world datasets include serialized versions of the MNIST dataset (Table 2), ETT dataset collection (Table 3), and datasets from the UEA (Bagnall et al., 2018) and UCR (Dau et al., 2019) time series classification archives (Appendix F.1).

Related work: The reservoir dynamics of ESNs has a large influence on its memorization and learning performance. A large body of work revolves around selecting the optimal topology for general performance Ma et al. (2014); Gallicchio & Micheli (2019); Dale et al. (2021); Xue et al. (2023). Indirectly, we take a similar approach by focusing on the kernel structure induced by the response function and derive a novel architecture that does not rely on recurrent dynamics. This study uses Bochner's theorem (Bochner, 1933), which states that positive definite kernels can always be represented in the frequency domain; a notable consequence for machine learning is the development of Random Fourier Features (Rahimi & Recht, 2007), where sampling from the spectral measure of a stationary kernel yields a finite feature map whose inner product approximates

[1]Commonly referred to as the connectivity or transition matrix in the reservoir computing literature.

Reservoir topology	GP interpretation
Random (Jaeger, 2001)	Non-stationary
Permutation (Rodan & Tino, 2011)	Stationary with deterministic and equidistant frequencies
Normal (Couillet et al., 2016)	Feature-wise locally stationary
Orthogonal (Tino, 2020)	Stationary

Table 1: Reservoir topologies from the proposed non-stationary Gaussian Process perspective on linear ESNs (assuming a spectral radius of the transfer matrix is one). All matrices are randomly initialized.

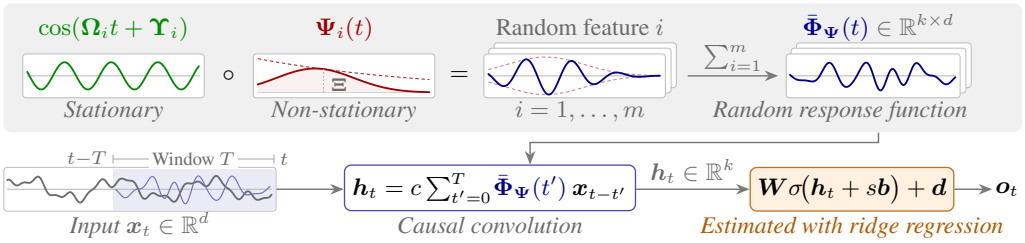


Figure 1: An overview of the components of the proposed TUNE architecture (see Section 3 for details).

that kernel. Furthermore, there are approaches for building feature spaces with frequency tuning, e.g., the Fourier recurrent unit (Zhang et al., 2018) relies on a Fourier basis expansion of linear state space models for more reliable long-term dependency learning. Several studies have used GPs and related kernel methods within reservoir computing (Hermans & Schrauwen, 2012). Notably, (Piatti et al., 2026) use an Euler-style recursion of Random Fourier Features to replace the non-linear hidden state of the ESN, approximating a path kernel with an architecture implemented in a frequency space, acting directly upon the input. We use similar ideas but rely on different techniques, focusing on the redesign of the memory mechanism by using Random Fourier Features that take time as an explicit input. An approach related to our study was pursued in (Pinna et al., 2026), that propose ParalESN: carefully constructed (shallow or deep) ESNs with a highly efficient temporal unrolling mechanism, which show competitive performance against deep network architectures on a variety of synthetic real-world datasets, whilst keeping the cost of training and inference to a minimum. Appendix C describes how ParalESN fits within the proposed architecture (Section 3.2) as a uniform-spectral-measure special case. We use shallow ParalESN as a baseline in all experiments, the critical difference diagram in Figure 2 summarizes the results and reports the average ranking of linear ESN, ParalESN, and TUNE (see Section 4 for experimental evaluation).

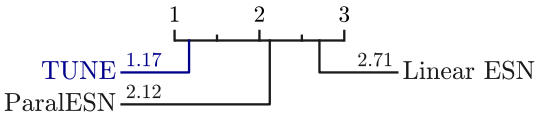


Figure 2: Critical difference diagram, all experiments (including Appendix) via Wilcoxon test (Demšar, 2006).

Contributions: Theoretically, we derive the non-stationary GP kernel associated with linear state space models driven by random input projection matrices, and characterize how different reservoir topologies affect the covariance kernel; we further show that the optimal spectral density of a stationary GP for maximizing the signal-to-noise ratio depends on the signal- and noise spectrum, providing a theoretically grounded argument for frequency tuning (Section 3 and Appendix A). We also provide experimental evidence for frequency tuning in Appendix F.4. Motivated by this theoretical connection, we introduce TUNE, a linear architecture that provides a controllable interpolation between stationary, locally stationary, and non-stationary behavior. The proposed architecture avoids repeated square matrix multiplications, yielding lower asymptotic computational complexity in the reservoir size and a reduced energy footprint compared to the linear ESN initialization.

2 Discrete- and Continuous-time Echo State Networks

This study focuses on extending ESNs with a linear recurrent layer and a non-linear readout layer; for completeness, we also present the standard formulation (Appendix B), in which the hidden state is non-linear and the readout layer is linear. The updates of the recurrent $\boldsymbol{h}_t$ and output $\boldsymbol{o}_t$ layers follow (Jaeger et al., 2007):

$$\textbf{non-linear ESN:} \quad \boldsymbol{h}_t = (1-\alpha)\boldsymbol{h}_{t-1} + \alpha\sigma(\tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}_{t-1} + s\boldsymbol{B}\boldsymbol{x}_t + s\boldsymbol{b}), \quad \boldsymbol{o}_t = \boldsymbol{W}\boldsymbol{h}_t + \boldsymbol{d}, \tag{1}$$

$$\textbf{linear ESN:} \quad \boldsymbol{h}_t = \tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}_{t-1} + \boldsymbol{B}\boldsymbol{x}_t, \quad \boldsymbol{o}_t = \boldsymbol{W}\sigma(c\boldsymbol{h}_t + s\boldsymbol{b}) + \boldsymbol{d}, \tag{2}$$

where $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ is input data with $\boldsymbol{x}_t \in \mathbb{R}^d$; $\boldsymbol{B} \in \mathbb{R}^{k \times d}$ is the input projection matrix with $\boldsymbol{B}_{i,j} \sim \mathcal{N}(0, 1/d)$; $\tilde{\boldsymbol{A}} \in \mathbb{R}^{k \times k}$ is a transfer matrix with a spectral radius $\rho(\tilde{\boldsymbol{A}}) = 1$; $\boldsymbol{W}$ are

the weights of the readout layer; $\boldsymbol{b} \in \mathbb{R}^k$ is the bias with $\boldsymbol{b}_i \sim \mathcal{N}(0,1)$; σ is a non-linear activation function ($\tanh$ in this study); α is the leakage hyperparameter; $\tilde{\lambda} \in [0,1]$ determines the decay rate; s is a hyperparameter that we refer to as the hidden state norm; and $c = s/\sqrt{\mathrm{Var}_T(\tilde{\lambda}^t \tilde{\boldsymbol{A}}^t \boldsymbol{B} dT)}$, with $\mathrm{Var}_T(\tilde{\lambda}^t \tilde{\boldsymbol{A}}^t \boldsymbol{B})$ the scalar variance of the response function multiplied with the decay rate measured over T, guaranteeing that $\boldsymbol{b}$ and $\boldsymbol{h}_t$ have identical variance, asserting unit variance $\boldsymbol{h}_t$.

ESNs can be seen as a subclass of simple recurrent networks (Elman, 1990), where most of the network parameters ($\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{b}$) are left untrained, whilst the readout layer ($\boldsymbol{W}$ and $\boldsymbol{d}$) is typically trained via linear- or ridge regression (Hoerl & Kennard, 1970). The non-linear ESN is the standard architecture in the reservoir computing literature, whilst the linear ESN is commonly used for analytical analysis (e.g., for studying memory capacity) (White et al., 2004; Tino, 2020), although variants resembling linear ESNs and related architectures have recently shown promising results (Bollt, 2021; Gauthier et al., 2021). The advantage of the linear ESN is its high level of parallelism as its hidden state update is a special case of the general discrete linear state equation (Brogan, 1991), which also has a continuous-time analogue; both forms have analytical solutions (zero initial states, $\boldsymbol{h}_{-1} = \boldsymbol{0}$ and $\boldsymbol{h}_0 = \boldsymbol{0}$ are assumed):

Discrete-time (this study), $t \in \mathbb{Z}$**:**	**Continuous-time,** $t \in \mathbb{R}$**:**
$\boldsymbol{h}_t = \tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}_{t-1} + \boldsymbol{B}\boldsymbol{x}_t,$	$\dfrac{d\boldsymbol{h}(t)}{dt} = (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{h}(t) + \boldsymbol{B}\boldsymbol{x}(t),$
$\boldsymbol{h}_t = \displaystyle\sum_{t'=0}^{t} \tilde{\lambda}^{t'} \tilde{\boldsymbol{A}}^{t'} \boldsymbol{B}\boldsymbol{x}_{t-t'},$	$\boldsymbol{h}(t) = \displaystyle\int_0^t e^{-\lambda t'} e^{\boldsymbol{A}t'} \boldsymbol{B}\boldsymbol{x}(t-t')\, dt',$

(3)

where $\tilde{\boldsymbol{A}} := e^{\boldsymbol{A}}$ and $\tilde{\lambda} := e^{-\lambda}$, so that $\tilde{\lambda}\tilde{\boldsymbol{A}} = e^{\boldsymbol{A}-\lambda\boldsymbol{I}}$ and the discrete response function corresponds to the continuous one when sampled for integer values of t. With the inclusion of a linear readout layer, these equations define (continuous-time) linear state space models. The solved hidden state over a window of length T (imposed for cleaner parallelization since the decay rate creates a cut-off implicitly) of a linear ESN in discrete- and continuous-time is as follows:

$$\boldsymbol{h}_t = \sum_{t'=0}^{T} \tilde{\lambda}^{t'} \boldsymbol{\Phi}_{t'} \boldsymbol{x}_{t-t'}, \qquad \boldsymbol{h}(t) = \int_0^T e^{-\lambda t'} \boldsymbol{\Phi}(t') \boldsymbol{x}(t-t')\, dt', \tag{4}$$

where $\boldsymbol{\Phi}_t = \tilde{\boldsymbol{A}}^t \boldsymbol{B}$, and $\boldsymbol{\Phi}(t) = e^{\boldsymbol{A}t}\boldsymbol{B}$ are the discrete- and continuous constructions; $\boldsymbol{o}_t$ and $\boldsymbol{o}(t)$ are identical to equation 2. The parameters $\boldsymbol{W}, \boldsymbol{d}$ are trained via linear or ridge regression to minimize $\mathbb{E}[(\boldsymbol{y} - \boldsymbol{o})^2]$ over targets $\boldsymbol{y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}$ with $\boldsymbol{y}_t \in \mathbb{R}^n$ and outputs $\boldsymbol{o} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_N\}$. The discrete formulation assumes that all data are sampled equidistantly, which is not a requirement for the continuous one.

3 A KERNEL PERSPECTIVE ON LINEAR ECHO STATE NETWORKS

The mathematical machinery used in this study is mostly identical for discrete- and continuous-time formulations; to be more general, we analyze the continuous-time linear state equation, and use the discrete-time variant when introducing and evaluating the proposed architecture.

Definition 3.1 (Matrix-valued GP)**.** *A matrix-valued stochastic process $\boldsymbol{\Phi}(t) \in \mathbb{R}^{k\times d}$, $t \in \mathcal{T}$, is a GP if for every finite set of points $\{t_1, \ldots, t_n\} \subset \mathcal{T}$ the stacked vector $\big(\mathrm{vec}\,\boldsymbol{\Phi}(t_1)^\top, \ldots, \mathrm{vec}\,\boldsymbol{\Phi}(t_n)^\top\big)^\top$ is jointly Gaussian. It is then fully specified by its mean function $\boldsymbol{m}(t) = \mathbb{E}[\boldsymbol{\Phi}(t)]$ together with its covariance, which throughout this study we report through the $k \times k$* row-covariance kernel

$$\boldsymbol{K}(t,t') = \mathbb{E}\big[(\boldsymbol{\Phi}(t) - \boldsymbol{m}(t))(\boldsymbol{\Phi}(t') - \boldsymbol{m}(t'))^\top\big].$$

A sample *is a single realization drawn from this process.*[2]

[2]Whenever $\boldsymbol{K}(t,t')$ has non-zero off-diagonal components the values $\boldsymbol{\Phi}(t)$ are correlated across both time and channels; if $\boldsymbol{K}$ is diagonal the channels are correlated through time only, and if those diagonal components are delta distributions in Δt the process is i.i.d., which corresponds to white noise.

Definition 3.2 (Stationarity classes)**.** *Let $\boldsymbol{\Phi}$ be a GP with mean function $\boldsymbol{m}(t)$ and row-covariance kernel $\boldsymbol{K}(t,t')$ as in Definition 3.1, and define $\Delta t = t - t'$. The process is classified as follows:*

1. ***Stationary:*** *There exists a function $\boldsymbol{K}_0$ such that $\boldsymbol{K}(t,t') = \boldsymbol{K}_0(\Delta t)$ for all t, t'.*
2. ***Locally stationary:*** *There exists a function $\boldsymbol{R}$ such that $\boldsymbol{K}(t,t') = \boldsymbol{R}(t+t', \Delta t)$, where $\boldsymbol{R}(u,\cdot)$ is a valid covariance function in Δt for each fixed u, and $\boldsymbol{R}(\cdot,\tau)$ varies on a time scale T_{loc} that is long relative to the autocorrelation time of $\boldsymbol{R}(u,\cdot)$, so that on any window short compared to T_{loc} the process is well approximated by a stationary one (Silverman, 1957).*
3. ***Non-stationary:*** *Neither of the above holds.*

It is well known that $\boldsymbol{h}(t)$ induces a GP in the infinite-width limit (Neal, 1996; Yang, 2019); such linear filters are always stationary by construction, as the only t dependence in $\boldsymbol{h}(t)$ is through $\boldsymbol{x}(t)$. Here we make a different and more elementary observation, not on $\boldsymbol{h}(t)$ but on $\boldsymbol{\Phi}(t)$, one that holds at any fixed width k and stems purely from the random initialization of $\boldsymbol{B}$.

Theorem 3.1 (Kernel structure – reduced statement)**.** *Let $\boldsymbol{A} \in \mathbb{R}^{k\times k}$ be fixed and let $\boldsymbol{B} \in \mathbb{R}^{k\times d}$ have i.i.d. components $\boldsymbol{B}_{i,j} \sim \mathcal{N}(0, 1/d)$. Then, at any finite reservoir size k, the response function $\boldsymbol{\Phi}(t) = e^{\boldsymbol{A}t}\boldsymbol{B}$ is, over the random initialization of $\boldsymbol{B}$, a zero-mean matrix-valued GP in the sense of Definition 3.1, Gaussian by linearity in $\boldsymbol{B}$, with row-covariance kernel*

$$\boldsymbol{K}(t,t') = e^{\boldsymbol{A}t}\, e^{\boldsymbol{A}^\top t'}.$$

Writing $\boldsymbol{H} = \frac{1}{2}(\boldsymbol{A}+\boldsymbol{A}^\top)$ for the symmetric part of $\boldsymbol{A}$ and $\boldsymbol{S} = \frac{1}{2}(\boldsymbol{A}-\boldsymbol{A}^\top)$ for its skew-symmetric part, this kernel decomposes as

$$\boldsymbol{K}(t,t') = e^{\boldsymbol{H}(t+t')+\boldsymbol{S}(t-t')+\boldsymbol{\Delta}_{BCH}(\boldsymbol{H},\boldsymbol{S},t,t')},$$

where $\boldsymbol{\Delta}_{BCH}$ collects the nested-commutator (Baker–Campbell–Hausdorff) terms.

Theorem 3.1 shows that the statistical behaviour of $\boldsymbol{\Phi}(t)$ is governed by the interplay between the symmetric part $\boldsymbol{H}$ and the skew-symmetric part $\boldsymbol{S}$ of $\boldsymbol{A}$: $\boldsymbol{H}$ enters only through the mid-point $t+t'$, $\boldsymbol{S}$ only through the lag Δt, and $\boldsymbol{\Delta}_{BCH}$ couples the two. Because $\boldsymbol{A}$ is fixed and only $\boldsymbol{B}$ is random, a width-k response function is a single realization of this process; the kernel $\boldsymbol{K}$, rather than any individual trajectory, is therefore the object we study. The following corollaries isolate the two topologies for which the coupling term vanishes: normal and skew-symmetric $\boldsymbol{A}$.

Corollary 3.2. *If $\boldsymbol{A}$ is a normal matrix (i.e. $\boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{A}^\top\boldsymbol{A}$), then $[\boldsymbol{H}, \boldsymbol{S}] = \boldsymbol{0}$ and, for all t, t',*

$$\boldsymbol{K}(t,t') = e^{\boldsymbol{H}(t+t')+\boldsymbol{S}(t-t')} = e^{\boldsymbol{H}(t+t')}e^{\boldsymbol{S}(t-t')},$$

which is a locally stationary kernel in the sense of Definition 3.2.

Corollary 3.3. *If $\boldsymbol{A}$ is skew-symmetric (i.e. $\boldsymbol{A}^\top = -\boldsymbol{A}$), then $\boldsymbol{H} = \boldsymbol{0}$ and $\boldsymbol{\Phi}$ is a stationary GP with orthogonal kernel $\boldsymbol{K}(t,t') = e^{\boldsymbol{A}(t-t')}$.*

The corollaries inform which class a given topology falls into (cf. Table 1), but not how to choose a kernel; the spectral measure of $\boldsymbol{K}$ is implicitly determined by the eigenvalues of $\boldsymbol{A}$, and remains opaque at initialization. Sections 3.1 and 3.2 therefore take the opposite approach and construct response functions whose kernel is specified explicitly, as a finite sum of m Monte-Carlo features, which we position as a substitute for the ESN response function.

3.1 Stationary Gaussian Processes as response function

In the response function $\boldsymbol{\Phi}(t) = e^{\boldsymbol{A}t}\boldsymbol{B}$, a skew-symmetric $\boldsymbol{A}$ treats every time point in the past equivalently, a normal $\boldsymbol{A}$ induces mode-wise exponential decay, and a general $\boldsymbol{A}$ is harder to interpret due to the mixing interactions. We therefore take a simplified approach and analyze the stationarity classes themselves. As suggested by the Bochner's theorem, any stationary GP has a spectral representation with a unique spectral measure $d\mu(\nu)$ (Gihman & Skorokhod, 2004; Rasmussen & Williams, 2005), which can be approximated via Random Fourier Features drawn from that spectral measure with an approximation rate of $\sqrt{\log(m)/m}$, where m is the number of Monte-Carlo

features (Rahimi & Recht, 2007); this is easily extended to matrix-valued GPs as:

$$\bar{\boldsymbol{\Phi}}(t) = \sqrt{\frac{2}{md}} \sum_{i=1}^{m} \boldsymbol{\Phi}_i(t), \qquad \boldsymbol{\Phi}_i(t) := \cos(\boldsymbol{\Omega}_i t + \boldsymbol{\Upsilon}_i), \tag{5}$$

where the cosine function is applied component-wise, $\boldsymbol{\Omega}_i, \boldsymbol{\Upsilon}_i \in \mathbb{R}^{k \times d}$, the components of $\boldsymbol{\Omega}_i$ are drawn from the spectral measure and $(\boldsymbol{\Upsilon}_i)_{j,l} \sim U(0, 2\pi)$, all i.i.d. over i, j and l; the bar indicates a finite ($m < \infty$) approximation; we also have $(2/m)\langle [\bar{\boldsymbol{\Phi}}_{i,j,l}(t)]_{i=1}^{m}, [\bar{\boldsymbol{\Phi}}_{i,j,l}(t')]_{i=1}^{m} \rangle \to \mathbb{E}[\bar{\boldsymbol{\Phi}}_{i,j,l}(t)\bar{\boldsymbol{\Phi}}_{i,j,l}(t')]$ as $m \to \infty$. By Lemma A.4 in Appendix A.5, $\bar{\boldsymbol{\Phi}}$ is zero-mean with $\mathbb{E}[\bar{\boldsymbol{\Phi}}(t)\bar{\boldsymbol{\Phi}}(t)^\top] = \boldsymbol{I}$. Components and features are independent and zero-mean, so only the diagonal terms remain and, $\mathbb{E}[\bar{\boldsymbol{\Phi}}_i(t)\bar{\boldsymbol{\Phi}}_i(t')^\top] = \mathbb{E}_\Omega[\cos(\Omega\Delta t)]\boldsymbol{I} =: k(\Delta t)\boldsymbol{I}$, where $\Delta t := t - t'$, Ω is distributed as a single component of $\boldsymbol{\Omega}_i$, and $k(\Delta t) = \operatorname{Re}\varphi(\Delta t)$ with φ the characteristic function of Ω. As $m \to \infty$, each component of $\bar{\boldsymbol{\Phi}}(t)$ converges in distribution to a zero-mean stationary GP with covariance $k(\Delta t)$ by the central limit theorem. This is not truly equivalent to general skew-symmetric $\boldsymbol{A}$ since $(e^{\boldsymbol{A}\Delta t})_{i,i'} \neq \delta(\Delta i)k(\Delta t)$; the channel-wise independent GP[3] does resemble a permutation matrix where $(\boldsymbol{P}^{\Delta t})_{i,i'} = \delta(\Delta i \bmod k, \Delta t \bmod k)$ if $k > T$. Independent sampling implies that $\lim_{T\to\infty} \int_0^T (\bar{\boldsymbol{\Phi}}(t))_{i,j} (\bar{\boldsymbol{\Phi}}(t))_{i',j'} dt = \delta(\Delta i)\delta(\Delta j)$, and approximately so for large time window-bandwidth time ratios T/ω, a desirable property directly sought out by (Voelker et al., 2019; Smith et al., 2023; Gu & Dao, 2024) by constructing orthogonal-in-time channels.

At the same time, it is not clear what the probability distribution of these random variables should be, or if tuning them is beneficial for performance; as justification, we show that the signal-to-noise ratio of the hidden state component $\bar{\boldsymbol{\Phi}}(t)_{i,j} := \bar{\phi}(t) \underset{m\to\infty}{\longrightarrow} \phi(t)$ for some infinite sequence and window size is maximized for a specific mean and standard deviation when the i.i.d. variables are Gaussian, the driving signal is fixed, and the hidden state is distorted by coloured noise (Theorem 3.4).

Theorem 3.4 (Bandwidth-optimal signal-to-noise ratio (SNR) – reduced statement)**.** *Let $h_i = \phi * (x + n)$, where ϕ is i.i.d. stationary Gaussian with frequencies $\sim \mathcal{N}(\mu, \omega^2)$, and the signal x and independent noise n have spectra $S_x := |\hat{x}|^2$ and S_n (non-negative, even, continuous, integrable, non-zero, finite second moment). With RMS detuning $\rho^2 := \int_{\mathbb{R}} (\nu - \mu)^2 S \, d\nu \big/ \int_{\mathbb{R}} S \, d\nu$, if $S_x(\mu) = 0$ and $\rho_x^2 < \rho_n^2$, the SNR has an interior maximizer $\omega^\star$; if $S_x(\mu) \neq 0$, none is guaranteed.*[4]

This optimum admits a *matched bandpass* interpretation, analogous to matched filters of classical detection theory (Turin, 1960). It suggests that tuned GPs can outperform linear ESNs, whose underlying spectral densities are not known upon initialization. Appendix F.4 provides an experimental verification that the SNR-optimal frequency overlaps with the optimal frequency for test accuracy. The SNR formulation assumes the noise is stationary and independent of the signal, which is a strong assumption for real-world applications; constructing a similar proof for non-stationary or signal-dependent noise is out of scope, and we therefore conjecture the following:

Conjecture 3.5. *If the noise corrupting the signal is non-stationary, then the response function maximizing the SNR of the hidden state is, in general, non-stationary.*

Note that in the scope of this paper SNR is a heuristic measure of performance, so one cannot assume that maximizing it is guaranteed to lead to better performance. In the next section we construct a variance-modulated non-stationary GP that allows for fine-tuning between stationary, local stationary, and non-stationary terms, which is subsequently used to introduce TUNE.

3.2 Local stationarity and non-stationarity as variance modulation

Local stationarity induces mode-specific exponential decay, which can also be introduced in a GP:

$$\bar{\boldsymbol{\Phi}}_{\boldsymbol{\Lambda}}(t) = \sqrt{\frac{2}{md}} \sum_{i=1}^{m} e^{-\boldsymbol{\Lambda}_i t} \circ \cos(\boldsymbol{\Omega}_i t + \boldsymbol{\Upsilon}_i), \tag{6}$$

inducing a kernel: $\mathbb{E}_{\Omega,\Lambda}\left[\cos(\Omega\Delta t)\, e^{-\Lambda(t+t')}\right] \boldsymbol{I} \sim k_0(t + t')k_s(\Delta t)\boldsymbol{I}$.

[3] Appendix A.8 provides a channel-correlated spatio-temporal GP construction, purely as a demonstration (not used in experiments).

[4] The full statement and proof are provided in Appendix A.4.

Adding such a term is known as variance modulation (Priestley, 1965), with kernel $K_\psi(t,t') = \psi(t)\psi(t')K(t-t')$, which induces time dependence in the variance of a GP, but not in its spectral measure. Inspired by the kernel expression for Gaussian i.i.d. variables and the first-order BCH correction inducing quadratic interactions, we opt for a heuristic approach to non-stationary variance modulation that induces sparsity in time:

$$\bar{\mathbf{\Phi}}_{\mathbf{\Psi}}(t) = \sqrt{\frac{2}{md}} \sum_{i=1}^{m} \mathbf{\Psi}_i(t) \circ \cos(\mathbf{\Omega}_i t + \mathbf{\Upsilon}_i), \tag{7}$$

where we multiply every feature with $\mathbf{\Psi}_i(t) = e^{-\mathbf{\Lambda}_i t - \frac{1}{2}\mathbf{\Theta}_i^2 (t-\mathbf{\Xi})^2}$ component-wise; the rank-2 tensor $\mathbf{\Xi}$ induces a component-specific biased time point, introducing non-stationarity per channel; the total term can be viewed as a random-translated Gabor filter (Gabor, 1946), with kernel:

$$\begin{aligned} \mathbb{E}[\bar{\mathbf{\Phi}}_{\mathbf{\Psi},i}(t)\bar{\mathbf{\Phi}}_{\mathbf{\Psi},i}(t')^\top] &= \mathbb{E}_{\Omega,\Lambda,\Theta,\Xi}\left[\cos(\Omega\Delta t)\, e^{-\Lambda(t+t') - \frac{1}{2}\Theta^2\left((t-\Xi)^2 + (t'-\Xi)^2\right)}\right] \boldsymbol{I} \\ &\sim k_0(t+t')k_1(t^2+t'^2)k_s(\Delta t)\boldsymbol{I}. \end{aligned} \tag{8}$$

3.3 TUNE: TUNEABLE AND UNCONSTRAINED NON-STATIONARY ECHO STATE NETWORK

Motivated by the preceding section, we let $\mathbf{\Phi}_t \to \bar{\mathbf{\Phi}}_{\mathbf{\Psi}}(t)$ (imposing discrete t) in the definition of a linear ESN (equation 4). The derivation did not assume any specific sampling distribution for the i.i.d. matrices, as they dictate the kernel induced in the infinite-sampling limit (Appendix A.6). Here we use Gaussian distributions, a common initialization in kernel methods literature, which induce a Radial Basis Function (RBF) kernel with an extra cosine term stemming from the mean of the Gaussian distributions when the process is stationary (Appendix A.5). The proposed *TUNE* architecture, shown schematically in Figure 1, is:

$$\bar{\mathbf{\Phi}}_{\mathbf{\Psi}}(t) = \sqrt{\frac{2}{md}} \sum_{i=1}^{m} \mathbf{\Psi}_i(t) \circ \cos(\mathbf{\Omega}_i t + \mathbf{\Upsilon}_i) \tag{9}$$

$$\boldsymbol{h}_t = \sum_{t'=0}^{T} \bar{\mathbf{\Phi}}_{\mathbf{\Psi}}(t')\boldsymbol{x}_{t-t'}, \quad \boldsymbol{o}_t = \boldsymbol{W}\sigma(c\boldsymbol{h}_t + s\boldsymbol{b}) + \boldsymbol{d}. \tag{10}$$

We define $\mathbf{\Psi}_i(t)$ as $\tilde{\mathbf{\Lambda}}_i^t e^{-\frac{1}{2}\mathbf{\Theta}_i^2(t-\mathbf{\Xi})^2}$ with $\mathbf{\Xi}_{i,j} \sim U(0,T)$, and c as in Section 2 with $\mathrm{Var}_T(\bar{\mathbf{\Phi}}_{\mathbf{\Psi}}(t))$ the variance of the response function measured over T. The terms $\mathbf{\Omega}_i, \mathbf{\Upsilon}_i, \mathbf{\Theta}_i \in \mathbb{R}^{k\times d}$ and $\tilde{\mathbf{\Lambda}}_i \in [0,1]^{k\times d}$ have components drawn randomly from $(\mathbf{\Omega}_i)_{j,l} \sim \mathcal{N}(\mu_\omega, \omega^2)$, $(\mathbf{\Upsilon}_i)_{j,l} \sim U(0, 2\pi)$, $(\mathbf{\Lambda}_i)_{j,l} \sim U(\lambda_{min}, \lambda_{max})$, and $(\mathbf{\Theta}_i)_{j,l} \sim \mathcal{N}(\mu_\theta, \theta^2)$. The Echo State Property (ESP) (Yildiz et al., 2012), quintessential in reservoir computing, is trivially satisfied by GPs and we prove it in Appendix A.7.

4 EXPERIMENTAL EVALUATION OF TUNE

Motivated by architectural similarities, the main baselines in the reported experiments are the linear ESN and shallow ParalESN (and the non-linear ESN in several experiments). The linear baselines differ only in the update of the hidden state. The evaluation is conducted under a resource-constrained setting where each model has a 10s training budget (Appendix E).

On the two synthetic bump tasks (Appendix E.6.1 and E.6.2), Figure 3 demonstrates that without the frequency tuning the linear ESN cannot efficiently extract localized information from noise, whereas ParalESN and TUNE learn both tasks. TUNE improves with the reservoir size on both tasks over the whole measured range, while ParalESN does so only on `KBumps`; its MSE on `BumpDistance` is not monotone in k, and is worst at $k = 2000$. At $k = 2000$ removing the envelope decreases TUNE's accuracy by 4% and increases MSE by 12%, while removing the stationary component does not, so non-stationary variance modulation is beneficial for larger networks.

Table 2 reports two serialized versions of the MNIST dataset, `sMNIST` and `psMNIST` (Appendix E.6.3), for models with $\sim$1K ($k = 100$), $\sim$8K ($k = T_{\text{seq}} = 784$), $\sim$50K ($k = 0.8N_{\text{train}}/n$) and $\sim$160K ($k = 16000$) trainable parameters. For $\sim$160K ($k = 16000$), Table 2 shows results from (Pinna et al., 2026) for additional baselines including long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997), transformer, S4, LRU (Orvieto et al., 2023), and

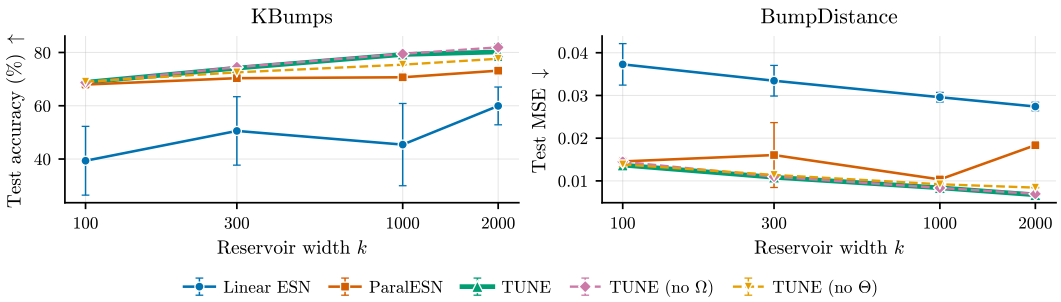


Figure 3: Linear ESN and ParalESN against TUNE (with ablation) on the synthetic bump tasks against k.

MAMBA. TUNE is the most accurate ESN-based architecture on sMNIST at every parameter count, where its advantage is most pronounced for smaller reservoirs. On psMNIST the ESN-based models perform similarly; in line with Theorem 3.4, tuning helps when signal and noise occupy different bands but the permutation spreads the signal over.

TUNE is the fastest model at ∼1K and near it at ∼8K and ∼50K, whereas the linear ESN scales poorly in k (Appendix D): at ∼50K it needs about $6\times$ the time and $15\times$ the energy of TUNE, and at ∼160K it runs out of time, as does the non-linear ESN from ∼50K. At ∼160K TUNE costs within 10% of ParalESN while being more accurate and stable, and with a closed-form readout it matches S4 and LRU on psMNIST and is top-2 on sMNIST trailing S4 by only 0.6%.

Table 3 reports the results for forecasting on the ETT dataset collection where TUNE is best in 9 of 12 settings and ranks second in the rest, with gains growing with the horizon; the non-linear ESN wins on ETTm2 for $H = 1$ and ETTh2 for $H \leq 96$ but becomes unstable at $H = 192$.

Appendix F presents additional experiments including the ablation of the oscillation ($\mathbf{\Omega}$), envelope ($\mathbf{\Theta}$), decay ($\mathbf{\Lambda}$) and Monte-Carlo features (m) on sMNIST and ETTh1. The stationary and non-stationary terms are largely interchangeable; removing either alone costs at most 1.2% ($k = 200$; without $\mathbf{\Omega}$ the model with $k = 2000$ is unstable) and 2% MSE.

Model	Param.	Test accuracy (%)		Training time per iter (s)		Energy per iter (Wh)	
		sMNIST	psMNIST	sMNIST	psMNIST	sMNIST	psMNIST
ESN	∼1K	73.1 ± 1.0	$\underline{82.8} \pm 0.1$	6.51	6.69	0.072	0.070
Linear ESN	∼1K	60.9 ± 7.1	82.6 ± 0.2	$\underline{0.09}$	$\underline{0.08}$	$\underline{0.002}$	$\underline{0.002}$
ParalESN	∼1K	$\underline{75.8} \pm 0.3$	82.7 ± 0.4	0.17	0.17	0.003	0.002
TUNE	∼1K	$\mathbf{86.9} \pm 0.3$	$\mathbf{82.9} \pm 0.3$	**0.05**	**0.05**	**0.002**	**0.002**
ESN	∼8K	89.2 ± 0.2	92.6 ± 0.4	6.64	6.91	0.430	0.427
Linear ESN	∼8K	91.9 ± 0.7	$\mathbf{93.0} \pm 0.1$	**0.06**	**0.05**	0.009	0.008
ParalESN	∼8K	$\underline{92.7} \pm 0.3$	90.3 ± 0.2	0.15	0.16	**0.004**	**0.004**
TUNE	∼8K	$\mathbf{95.2} \pm 0.2$	$\underline{92.9} \pm 0.1$	$\underline{0.08}$	$\underline{0.08}$	$\underline{0.006}$	$\underline{0.006}$
ESN	∼50K	–	–	o.o.t.	o.o.t.	o.o.t.	o.o.t.
Linear ESN	∼50K	$\underline{96.9} \pm 0.1$	$\mathbf{97.1} \pm 0.2$	4.21	3.96	0.740	0.674
ParalESN	∼50K	94.8 ± 0.2	96.2 ± 0.1	**0.52**	**0.53**	**0.039**	**0.040**
TUNE	∼50K	$\mathbf{97.9} \pm 0.0$	$\underline{96.9} \pm 0.1$	$\underline{0.63}$	$\underline{0.69}$	$\underline{0.044}$	$\underline{0.047}$
ESN	∼160K	–	–	o.o.t.	o.o.t.	o.o.t.	o.o.t.
Linear ESN	∼160K	–	–	o.o.t.	o.o.t.	o.o.t.	o.o.t.
ParalESN	∼160K	76.6 ± 17.2	97.4 ± 0.1	**7.64**	**7.74**	**0.500**	**0.510**
TUNE	∼160K	$\underline{98.6} \pm 0.0$	$\mathbf{98.0} \pm 0.1$	$\underline{8.33}$	$\underline{7.93}$	$\underline{0.546}$	$\underline{0.514}$
LSTM	∼160K	97.5 ± 1.4	92.8 ± 0.5	n/a	n/a	n/a	n/a
Transformer	∼160K	98.4 ± 0.1	97.4 ± 0.2	n/a	n/a	n/a	n/a
S4	∼160K	$\mathbf{99.2} \pm 0.0$	$\underline{97.9} \pm 0.0$	n/a	n/a	n/a	n/a
LRU	∼160K	98.5 ± 0.2	97.8 ± 0.1	n/a	n/a	n/a	n/a
MAMBA	∼160K	98.4 ± 0.1	92.6 ± 0.1	n/a	n/a	n/a	n/a

Table 2: Performance (test accuracy, training time and energy per training iteration) on sMNIST and psMNIST. Coloured rows indicate ESN-based models and *italics* are lifted results. **Bold** indicates best result and underline second-best (ranked before rounding).

Dataset	Horizon, H	ESN, MSE	Linear ESN, MSE	ParalESN, MSE	TUNE, MSE
ETTh1	1	0.204 ± 0.002	$\underline{0.204} \pm 0.002$	0.207 ± 0.004	$\mathbf{0.198} \pm 0.002$
	96	0.660 ± 0.007	0.652 ± 0.015	$\underline{0.649} \pm 0.028$	$\mathbf{0.609} \pm 0.001$
	192	0.901 ± 0.075	$\underline{0.792} \pm 0.035$	0.793 ± 0.005	$\mathbf{0.766} \pm 0.003$
ETTh2	1	$\mathbf{0.093} \pm 0.004$	0.107 ± 0.010	0.099 ± 0.004	$\underline{0.095} \pm 0.008$
	96	$\mathbf{0.481} \pm 0.011$	0.960 ± 0.044	0.672 ± 0.034	$\underline{0.614} \pm 0.044$
	192	1.856 ± 2.561	1.558 ± 0.199	$\underline{1.489} \pm 0.110$	$\mathbf{1.305} \pm 0.103$
ETTm1	1	$\underline{0.083} \pm 0.000$	0.085 ± 0.000	0.084 ± 0.001	$\mathbf{0.082} \pm 0.000$
	96	0.513 ± 0.027	0.679 ± 0.014	$\underline{0.496} \pm 0.014$	$\mathbf{0.461} \pm 0.002$
	192	$\underline{0.546} \pm 0.010$	0.727 ± 0.014	0.548 ± 0.006	$\mathbf{0.515} \pm 0.001$
ETTm2	1	$\mathbf{0.032} \pm 0.001$	0.033 ± 0.002	0.034 ± 0.001	$\underline{0.033} \pm 0.001$
	96	$\underline{0.492} \pm 0.018$	0.870 ± 0.022	1.104 ± 0.506	$\mathbf{0.358} \pm 0.046$
	192	$\underline{0.483} \pm 0.029$	0.977 ± 0.025	0.970 ± 0.017	$\mathbf{0.478} \pm 0.091$

Table 3: Long-term forecasting on ETT datasets; model uses $k = 100$ and $T = 0.01T_{\text{seq}}$. **Bold** indicates best result and underline second-best (ranked before rounding).

5 Discussion

In this study, we leveraged the parallels between linear Echo State Networks and matrix-valued Gaussian Processes, which suggests that the kernel induced by the response function can be considered in terms of its stationary, locally stationary, or non-stationary exponential terms. This perspective enables the classification of existing reservoir topologies (cf. Table 1) and suggests that replacing recurrent dynamics with explicit tuneable non-iterative stochastic samples improves performance on most considered benchmarks regardless of model size. These tuneable components are designed in such a way that they mimic the stationarity classes of various reservoir topologies, providing a kernel perspective on spectral tuning. Theorem 3.4 accounts for part of this result: tuning a spectral measure of reservoir topology can place the response function where the power of the signal dominates that of the noise. The theorem-suggested fixed-point equation locates the accuracy-optimal bandwidth on StarLightCurves, FaultDetectionA and Blink, but not on sMNIST and psMNIST (Appendix F.4), and requires information about the noise spectrum. It is therefore a starting point for future investigations into zero-shot hyperparameter optimization. The theorem is also stated for i.i.d. Gaussian matrices and assumes the noise to be stationary and independent of the signal. This assumption is violated in the non-stationary regime that motivates the envelope term and hence it is left as a conjecture in this case. The ablations in Appendix F.3 show how the terms interact: the stationary and non-stationary components are largely interchangeable on sMNIST and ETTh1, the envelope term matters most when information is localized in time, as in KBumps and BumpDistance.

The proposed TUNE architecture should be viewed as a design framework rather than a rigid construction; the random matrix initialization and variance modulation term can be changed to induce any desired kernel behaviour. ParalESN remains ahead on the chaotic Rossler, Lorenz96 and DoublePendulum systems, possibly because the TUNE kernel is diagonal across channels, which the channel-correlated construction of Appendix A.8 could address. The non-linear Echo State Network remains ahead on ETTh2 for $H \leq 96$ and ETTm2 for $H = 1$, suggesting that non-linear recurrence still leads to useful representations at shorter horizons. Notably, tuneability enlarges the search space, with up to eight hyperparameters against three for the linear Echo State Network under the same trial budget, and all comparisons were made under a 10s budget against a shallow ParalESN, with results for gradient-trained models taken from the literature. Finally, the explored non-stationary term is chosen ad-hoc: the envelope modulates the amplitude, and hence the variance, of each channel while leaving its spectral measure time-invariant; we also explored a time-varying spectral measure via multi-layer GPs; we did not find any performance improvement over the stationary case.

Lastly, the kernel perspective suggests several plausible extensions. The continuous-time formulation, in which time is an explicit input in the response function can be extended to process irregularly sampled sequential data. Linearized around a fixed point, a non-linear Echo State Network has a response function of the same form, with the Jacobian in place of the transfer matrix, so the perspective may also inform non-linear and physical reservoirs, whose impulse responses are fixed but measurable. We leave addressing these extensions, and a proof of the conjecture, to future work.

REFERENCES

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive, 2018, 2018. URL http://arxiv.org/abs/1811.00075. arXiv:1811.00075.

Sergio Blanes and Fernando Casas. On the convergence and optimization of the Baker–Campbell–Hausdorff formula. *Linear Algebra and its Applications*, 378:135–158, 2004. ISSN 00243795. doi: 10.1016/j.laa.2003.09.010. URL https://linkinghub.elsevier.com/retrieve/pii/S002437950300778X.

Salomon Bochner. Monotone funktionen, stieltjessche integrale und harmonische analyse. *Mathematische Annalen*, 108(1):378–410, 1933. ISSN 0025-5831, 1432-1807. doi: 10.1007/BF01452844. URL http://link.springer.com/10.1007/BF01452844.

Erik Bollt. On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(1):013108, 2021. ISSN 1054-1500, 1089-7682. doi: 10.1063/5.0024890. URL https://pubs.aip.org/cha/article/31/1/013108/341924/On-explaining-the-surprising-success-of-reservoir.

Richard P. Brent. An algorithm with guaranteed convergence for finding a minimum of a function of one variable. In *Algorithms for Minimization without Derivatives*, Prentice-Hall Series in Automatic Computation. Prentice-Hall, Englewood Cliffs, NJ, 1973. ISBN 0-13-022335-2.

William L. Brogan. *Modern control theory (3rd ed.)*. Prentice-Hall, Inc., USA, 1991. ISBN 0-13-589763-7.

Romain Couillet, Gilles Wainrib, Hafiz Tiomoko Ali, and Harry Sevi. A Random Matrix Approach to Echo-State Neural Networks. In *International Conference on Machine Learning (ICLR)*, volume 48, pp. 517–525, New York, New York, USA, 2016. URL https://proceedings.mlr.press/v48/couillet16.html.

Benoit Courty, Victor Schmidt, Sasha Luccioni, Goyal-Kamal, MarionCoutarel, Boris Feld, Jérémy Lecourt, LiamConnell, Amine Saboni, Inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, Marc Alencon, Michal Stechly, Christian Bauer, Lucas Otávio N. de Araújo, JPW, and MinervaBooks. mlco2/codecarbon: v2.4.1, 2024. URL https://doi.org/10.5281/zenodo.11171501.

Hongyan Cui, Xiang Liu, and Lixiang Li. The architecture of dynamic reservoir in the echo state network. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(3):033127, 2012. ISSN 1054-1500, 1089-7682. doi: 10.1063/1.4746765. URL https://pubs.aip.org/cha/article/22/3/033127/912036/The-architecture-of-dynamic-reservoir-in-the-echo.

Matthew Dale, Simon O'Keefe, Angelika Sebald, Susan Stepney, and Martin A. Trefzer. Reservoir computing quality: Connectivity and topology. *Natural Computing*, 20(2):205–216, 2021. ISSN 1567-7818, 1572-9796. doi: 10.1007/s11047-020-09823-1. URL https://link.springer.com/10.1007/s11047-020-09823-1.

Joni Dambre, David Verstraeten, Benjamin Schrauwen, and Serge Massar. Information Processing Capacity of Dynamical Systems. *Scientific Reports*, 2(1):514, 2012. ISSN 2045-2322. doi: 10.1038/srep00514. URL https://www.nature.com/articles/srep00514.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019. ISSN 2329-9266, 2329-9274. doi: 10.1109/JAS.2019.1911747. URL https://ieeexplore.ieee.org/document/8894743/.

Janez Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006. URL http://jmlr.org/papers/v7/demsar06a.html.

Jeffrey L. Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 1990. ISSN 0364-0213, 1551-6709. doi: 10.1207/s15516709cog1402_1. URL https://onlinelibrary.wiley.com/doi/10.1207/s15516709cog1402_1.

D. Gabor. Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering*, 93(26):429–441, 1946. ISSN 2054-0604. doi: 10.1049/ji-3-2.1946.0074. URL https://digital-library.theiet.org/content/journals/10.1049/ji-3-2.1946.0074.

Claudio Gallicchio and Alessio Micheli. Reservoir topology in deep echo state networks. In *International Conference on Artificial Neural Networks (ICANN)*, volume 11731, pp. 62–75. Cham, 2019. ISBN 978-3-030-30492-8 978-3-030-30493-5. doi: 10.1007/978-3-030-30493-5_6. URL https://link.springer.com/10.1007/978-3-030-30493-5_6.

Daniel J. Gauthier, Erik Bollt, Aaron Griffith, and Wendson A. S. Barbosa. Next generation reservoir computing. *Nature Communications*, 12(1):5564, 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-25801-2. URL https://www.nature.com/articles/s41467-021-25801-2.

Iosif Il'ich Gihman and Anatoliĭ Vladimirovich Skorokhod. *The Theory of Stochastic Processes I*. Classics in Mathematics. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-20284-4 978-3-642-61943-4. doi: 10.1007/978-3-642-61943-4. URL http://link.springer.com/10.1007/978-3-642-61943-4.

William Gilpin. Chaos as an interpretable benchmark for forecasting and data-driven modelling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://openreview.net/forum?id=enYjtbjYJrf.

William Gilpin. Model scale versus domain knowledge in statistical forecasting of chaotic systems. *Physical Review Research*, 5(4):043252, 2023. ISSN 2643-1564. doi: 10.1103/PhysRevResearch.5.043252. URL https://link.aps.org/doi/10.1103/PhysRevResearch.5.043252.

Lyudmila Grigoryeva and Juan-Pablo Ortega. Echo state networks are universal. *Neural Networks*, 108:495–508, 2018. ISSN 08936080. doi: 10.1016/j.neunet.2018.08.025. URL https://linkinghub.elsevier.com/retrieve/pii/S089360801830251X.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently Modeling Long Sequences with Structured State Spaces, 2021. URL https://arxiv.org/abs/2111.00396. Version Number: 3.

Michiel Hermans and Benjamin Schrauwen. Recurrent kernel machines: Computing with infinite echo state networks. *Neural Computation*, 24(1):104–133, 2012. ISSN 0899-7667, 1530-888X. doi: 10.1162/NECO_a_00200. URL https://direct.mit.edu/neco/article/24/1/104-133/7730.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997. ISSN 0899-7667, 1530-888X. doi: 10.1162/neco.1997.9.8.1735. URL https://direct.mit.edu/neco/article/9/8/1735-1780/6109.

Arthur E. Hoerl and Robert W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67, 1970. ISSN 0040-1706, 1537-2723. doi: 10.1080/00401706.1970.10488634. URL https://www.tandfonline.com/doi/full/10.1080/00401706.1970.10488634.

Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. Technical report, German national research center for information technology gmd technical report, Bonn, Germany, 2001.

Herbert Jaeger, Mantas Lukoševičius, Dan Popovici, and Udo Siewert. Optimization and applications of echo state networks with leaky- integrator neurons. *Neural Networks*, 20(3):335–352, 2007. ISSN 08936080. doi: 10.1016/j.neunet.2007.04.016. URL https://linkinghub.elsevier.com/retrieve/pii/S089360800700041X.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes, 2018. URL http://arxiv.org/abs/1711.00165. arXiv:1711.00165 [stat.ML].

Qianli Ma, Weibiao Chen, Jia Wei, and Zhiwen Yu. Direct model of memory properties and the linear reservoir topologies in echo state networks. *Applied Soft Computing*, 22:622–628, 2014. ISSN 15684946. doi: 10.1016/j.asoc.2014.04.038. URL https://linkinghub.elsevier.com/retrieve/pii/S1568494614002142.

Per Christian Moan and Jitse Niesen. Convergence of the Magnus Series. *Foundations of Computational Mathematics*, 8(3):291–301, 2008. doi: 10.1007/s10208-007-9010-0.

Cleve Moler and Charles Van Loan. Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later. *SIAM Review*, 45(1):3–49, 2003. doi: 10.1137/S00361445024180.

Radford M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer New York, New York, NY, 1996. ISBN 978-0-387-94724-2 978-1-4612-0745-0. doi: 10.1007/978-1-4612-0745-0. URL http://link.springer.com/10.1007/978-1-4612-0745-0.

Antonio Orvieto, Samuel L. Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting Recurrent Neural Networks for Long Sequences, 2023. URL http://arxiv.org/abs/2303.06349. arXiv:2303.06349.

Francesco Piatti, Thomas Cass, and William F. Turner. Random Controlled Differential Equations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kHqt0ZSbKT.

Matteo Pinna, Giacomo Lagomarsini, Andrea Ceni, and Claudio Gallicchio. ParalESN: Enabling parallel information processing in reservoir computing. In *International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=NsJUKcONFR.

M. B. Priestley. Evolutionary Spectra and Non-Stationary Processes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 27(2):204–229, 1965. ISSN 1369-7412, 1467-9868. doi: 10.1111/j.2517-6161.1965.tb01488.x. URL https://academic.oup.com/jrsssb/article/27/2/204/7026825.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 20, 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005. ISBN 978-0-262-25683-4. doi: 10.7551/mitpress/3206.001.0001. URL https://direct.mit.edu/books/book/2320/Gaussian-Processes-for-Machine-Learning.

A Rodan and P Tino. Minimum Complexity Echo State Network. *IEEE Transactions on Neural Networks*, 22(1):131–144, 2011. ISSN 1045-9227, 1941-0093. doi: 10.1109/TNN.2010.2089641. URL http://ieeexplore.ieee.org/document/5629375/.

R. Silverman. Locally stationary random processes. *IEEE Transactions on Information Theory*, 3(3):182–187, 1957. ISSN 0018-9448. doi: 10.1109/TIT.1957.1057413. URL http://ieeexplore.ieee.org/document/1057413/.

Jimmy T. H. Smith, Andrew Warrington, and Scott Linderman. Simplified State Space Layers for Sequence Modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Ai8Hw3AXqks.

Peter Tino. Dynamical systems as temporal feature spaces. *Journal of Machine Learning Research*, 21(1), 2020. ISSN 1532-4435.

G. Turin. An introduction to matched filters. *IEEE Transactions on Information Theory*, 6(3):311–329, 1960. ISSN 0018-9448. doi: 10.1109/TIT.1960.1057571. URL http://ieeexplore.ieee.org/document/1057571/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

David Verstraeten, Joni Dambre, Xavier Dutoit, and Benjamin Schrauwen. Memory versus non-linearity in reservoirs. In *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Barcelona, Spain, 2010. ISBN 978-1-4244-6916-1. doi: 10.1109/IJCNN.2010.5596492. URL http://ieeexplore.ieee.org/document/5596492/.

Aaron Voelker, Ivana Kajić, and Chris Eliasmith. Legendre memory units: Continuous-time representation in recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/952285b9b7e7a1be5aa7849f32ffff05-Paper.pdf.

Olivia L. White, Daniel D. Lee, and Haim Sompolinsky. Short-Term Memory in Orthogonal Neural Networks. *Physical Review Letters*, 92(14):148102, 2004. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.92.148102. URL https://link.aps.org/doi/10.1103/PhysRevLett.92.148102.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting, 2022. URL http://arxiv.org/abs/2106.13008. arXiv:2106.13008.

Yu Xue, Qi Zhang, and Adam Slowik. Automatic topology optimization of echo state network based on particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 117:105574, 2023. ISSN 09521976. doi: 10.1016/j.engappai.2022.105574. URL https://linkinghub.elsevier.com/retrieve/pii/S0952197622005644.

Greg Yang. Wide Feedforward or Recurrent Neural Networks of Any Architecture are Gaussian Processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5e69fda38cda2060819766569fd93aa5-Paper.pdf.

Izzet B. Yildiz, Herbert Jaeger, and Stefan J. Kiebel. Re-visiting the echo state property. *Neural Networks*, 35:1–9, 2012. ISSN 08936080. doi: 10.1016/j.neunet.2012.07.005. URL https://linkinghub.elsevier.com/retrieve/pii/S0893608012001852.

Jiong Zhang, Yibo Lin, Zhao Song, and Inderjit Dhillon. Learning long term dependencies via Fourier recurrent units. In *International Conference on Machine Learning (ICML)*, volume 80, pp. 5815–5823, 2018. URL https://proceedings.mlr.press/v80/zhang18h.html.

AI USE STATEMENT

In this study, we used generative AI tools (specifically, Claude Opus 5.0 between extra high- and max effort in all instances) to verify our mathematical constructions and methodology and occasionally provide initial ideations for proofs; we also deployed AI tools, that were supervised, to streamline our implemented protocol for experimental evaluation (not the experiments themselves); it was also used to check and suggest corrections to our experiment- and model implementations

(code exclusively), based on up-to-date drafts of the paper, to provide an additional level of assurance in correctness and accuracy of the declared results. We also let AI tools suggest grammatical improvements to the main text. AI tools were not used in a way allowing autonomous operation in any environment. All AI-assisted work has been reviewed manually by checking suggested alterations. We take responsibility for the final content of this study, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

All data, figures, and code will be made publicly available upon publication.

A Theory and derivations

A.1 Proof of Theorem 3.1

Theorem A.1 (Kernel structure – full statement)**.** *Let $\boldsymbol{A} \in \mathbb{R}^{k\times k}$ be fixed and let $\boldsymbol{B} \in \mathbb{R}^{k\times d}$ have i.i.d. components $\boldsymbol{B}_{i,j} \sim \mathcal{N}(0, 1/d)$. Then, at any finite reservoir size k, the response function $\boldsymbol{\Phi}(t) = e^{\boldsymbol{A}t}\boldsymbol{B}$ is, over the random initialization of $\boldsymbol{B}$, a zero-mean matrix-valued GP in the sense of Definition 3.1, Gaussian by linearity in $\boldsymbol{B}$, with row-covariance kernel*

$$\boldsymbol{K}(t, t') = e^{\boldsymbol{A}t}\, e^{\boldsymbol{A}^\top t'}.$$

Writing $\boldsymbol{H} = \frac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^\top)$ for the symmetric part of $\boldsymbol{A}$ and $\boldsymbol{S} = \frac{1}{2}(\boldsymbol{A} - \boldsymbol{A}^\top)$ for its skew-symmetric part, this kernel decomposes as

$$\boldsymbol{K}(t, t') = e^{\boldsymbol{H}(t+t') + \boldsymbol{S}(t-t') + \boldsymbol{\Delta}_{BCH}(\boldsymbol{H},\boldsymbol{S},t,t')},$$

where $\boldsymbol{\Delta}_{BCH}$ collects the nested-commutator (Baker–Campbell–Hausdorff) terms, a series that converges whenever $\|\boldsymbol{A}\|_2(|t| + |t'|) < \pi$ (Moan & Niesen, 2008). Without any restriction on t, t',

$$\left\|\boldsymbol{K}(t, t') - e^{\boldsymbol{H}(t+t') + \boldsymbol{S}(t-t')}\right\|_2 \le |tt'|\, \|[\boldsymbol{H}, \boldsymbol{S}]\|_2\, e^{\|\boldsymbol{A}\|_2(|t|+|t'|)},$$

so the decomposition holds exactly, with $\boldsymbol{\Delta}_{BCH} = \boldsymbol{0}$, when $[\boldsymbol{H}, \boldsymbol{S}] = \boldsymbol{0}$. The process is in general non-stationary in the sense of Definition 3.2.

Proof. The response function $\boldsymbol{\Phi}(t) = e^{\boldsymbol{A}t}\boldsymbol{B}$ is linear in the Gaussian matrix $\boldsymbol{B}$, so its finite-dimensional marginals are jointly Gaussian with zero mean (Gihman & Skorokhod, 2004). Since $\mathbb{E}[\boldsymbol{B}\boldsymbol{B}^\top] = \boldsymbol{I}_k$, $\boldsymbol{K}(t, t') = e^{\boldsymbol{A}t}\,\mathbb{E}[\boldsymbol{B}\boldsymbol{B}^\top]\, e^{\boldsymbol{A}^\top t'} = e^{\boldsymbol{A}t}e^{\boldsymbol{A}^\top t'}$, and because the columns of $\boldsymbol{B}$ are i.i.d., $\mathrm{Cov}\big(\mathrm{vec}\,\boldsymbol{\Phi}(t), \mathrm{vec}\,\boldsymbol{\Phi}(t')\big) = \frac{1}{d}\,\boldsymbol{I}_d \otimes \boldsymbol{K}(t, t')$, so $\boldsymbol{K}$ determines the process.

Write $\boldsymbol{K}(t, t') = e^{\boldsymbol{X}}e^{\boldsymbol{Y}}$ with $\boldsymbol{X} = \boldsymbol{A}t = (\boldsymbol{H} + \boldsymbol{S})t$ and $\boldsymbol{Y} = \boldsymbol{A}^\top t' = (\boldsymbol{H} - \boldsymbol{S})t'$. Then $\boldsymbol{X} + \boldsymbol{Y} = \boldsymbol{H}(t + t') + \boldsymbol{S}(\Delta t)$ and $[\boldsymbol{X}, \boldsymbol{Y}] = tt'[\boldsymbol{A}, \boldsymbol{A}^\top] = -2tt'[\boldsymbol{H}, \boldsymbol{S}]$. By the Baker–Campbell–Hausdorff (BCH) formula, $e^{\boldsymbol{X}}e^{\boldsymbol{Y}} = \exp\big(\boldsymbol{X} + \boldsymbol{Y} + \frac{1}{2}[\boldsymbol{X}, \boldsymbol{Y}] + \dots\big)$, where every higher-order term is a nested commutator with innermost bracket $[\boldsymbol{X}, \boldsymbol{Y}]$, and the series converges for $\|\boldsymbol{X}\|_2 + \|\boldsymbol{Y}\|_2 < \pi$ (Blanes & Casas, 2004; Moan & Niesen, 2008); this gives the decomposition with $\boldsymbol{\Delta}_{BCH} = -tt'[\boldsymbol{H}, \boldsymbol{S}] + \dots$. The global bound is the splitting estimate $\|e^{\boldsymbol{X}+\boldsymbol{Y}} - e^{\boldsymbol{X}}e^{\boldsymbol{Y}}\| \le \frac{1}{2}\|[\boldsymbol{X}, \boldsymbol{Y}]\|\, e^{\|\boldsymbol{X}\|+\|\boldsymbol{Y}\|}$ (Moler & Van Loan, 2003, equation (7.2)). □

A.2 Proof of Corollary 3.2

Proof. Since $[\boldsymbol{A}, \boldsymbol{A}^\top] = -2[\boldsymbol{H}, \boldsymbol{S}]$, normality gives $[\boldsymbol{H}, \boldsymbol{S}] = \boldsymbol{0}$, so the bound in Theorem 3.1 vanishes and $\boldsymbol{K}(t, t') = e^{\boldsymbol{H}(t+t') + \boldsymbol{S}(\Delta t)} = e^{\boldsymbol{H}(t+t')}e^{\boldsymbol{S}(\Delta t)}$ for all t, t', as $\boldsymbol{H}$ and $\boldsymbol{S}$ commute. □

A.3 Proof of Corollary 3.3

Proof. A skew-symmetric $\boldsymbol{A}$ is normal with $\boldsymbol{H} = \boldsymbol{0}$ and $\boldsymbol{S} = \boldsymbol{A}$, so Corollary 3.2 gives $\boldsymbol{K}(t, t') = e^{\boldsymbol{A}(\Delta t)}$, which is orthogonal and depends only on Δt. □

A.4 Proof of Theorem A.3

This subsection gives the full statement of Theorem 3.4, as Theorem A.3, together with its proof; the proof rests on the following lemma.

Lemma A.2 (Gaussian-smoothed spectrum)**.** *Let $g_{\mu,\omega}(\nu) = (\sqrt{2\pi}\,\omega)^{-1} e^{-(\nu-\mu)^2/2\omega^2}$ be the $\mathcal{N}(\mu,\omega^2)$ density with a fixed mean μ, and $G_{\mu,\omega} = \frac{1}{2}(g_{\mu,\omega} + g_{-\mu,\omega})$ its symmetrization. Let $W \geq 0$ be even, continuous, integrable, and not identically zero. Then*

$$V(\omega) := \int_{\mathbb{R}} G_{\mu,\omega}(\nu)\, W(\nu)\, d\nu = \int_{\mathbb{R}} g_{\mu,\omega}(\nu)\, W(\nu)\, d\nu > 0$$

is smooth on $(0,\infty)$ with

$$V'(\omega) = \frac{V(\omega)}{\omega^3}\Big(\big\langle (\nu-\mu)^2 \big\rangle_\omega - \omega^2\Big), \qquad \langle f(\nu)\rangle_\omega := \frac{\int_{\mathbb{R}} f\, W g_{\mu,\omega}\, d\nu}{\int_{\mathbb{R}} W g_{\mu,\omega}\, d\nu},$$

and $V(\omega) \to W(\mu)$ as $\omega \to 0^+$ and $V(\omega) \to 0$ as $\omega \to \infty$.

Proof. Substituting $\nu \mapsto -\nu$ and using $g_{-\mu,\omega}(\nu) = g_{\mu,\omega}(-\nu)$ together with $W(-\nu) = W(\nu)$ gives $\int_{\mathbb{R}} g_{-\mu,\omega} W = \int_{\mathbb{R}} g_{\mu,\omega} W$, hence $\int_{\mathbb{R}} G_{\mu,\omega} W = \int_{\mathbb{R}} g_{\mu,\omega} W$; positivity follows from $W \geq 0$, $W \not\equiv 0$. Differentiating under the integral with $\partial_\omega g_{\mu,\omega} = \big((\nu-\mu)^2/\omega^3 - 1/\omega\big) g_{\mu,\omega}$ is justified because, on any $\omega \in [\omega_0, \omega_1] \subset (0,\infty)$, the substitution $u = (\nu-\mu)/\omega$ bounds the integrand by $C(\omega_0)\, W(\nu) \in L^1$, so dominated convergence applies and

$$V'(\omega) = \frac{1}{\omega^3} \int_{\mathbb{R}} \big((\nu-\mu)^2 - \omega^2\big) W g_{\mu,\omega}\, d\nu = \frac{V(\omega)}{\omega^3}\Big(\big\langle (\nu-\mu)^2 \big\rangle_\omega - \omega^2\Big).$$

As $\omega \to 0^+$ the Gaussian $g_{\mu,\omega} \to \delta_\mu$ weakly, so $V \to W(\mu)$ by continuity of W; as $\omega \to \infty$, $V \leq \|g_{\mu,\omega}\|_\infty \|W\|_1 = \|W\|_1/(\sqrt{2\pi}\,\omega) \to 0$. □

Theorem A.3 (Bandwidth-optimized signal-to-noise ratio – full statement)**.** *Let the scalar response function ϕ be i.i.d. stationary Gaussian with spectral measure $\mathcal{N}(\mu,\omega^2)$ with a fixed mean μ, so that its expected power spectrum is the symmetrized Gaussian*

$$\mathbb{E}|\hat{\phi}|^2 = G_{\mu,\omega} := \tfrac{1}{2}\big(g_{\mu,\omega} + g_{-\mu,\omega}\big), \qquad g_{\mu,\omega}(\nu) = \tfrac{1}{\sqrt{2\pi}\,\omega} e^{-(\nu-\mu)^2/2\omega^2}.$$

Let it drive $h_i = \phi * (x+n)$ under a signal x with power spectrum $S_x := |\hat{x}|^2$ and independent coloured noise n with power spectrum S_n, both non-negative, even, continuous, integrable, not identically zero and of finite second moment. Define $\mathrm{SNR}(\omega) := V_x(\omega)/V_n(\omega)$ with $V_x = \int_{\mathbb{R}} G_{\mu,\omega} S_x\, d\nu$ and $V_n = \int_{\mathbb{R}} G_{\mu,\omega} S_n\, d\nu$. Then every interior critical point $\omega^\star$ of SNR satisfies the matching condition

$$\frac{\int_{\mathbb{R}} (\nu-\mu)^2 S_x\, g_{\mu,\omega^\star}(\nu)\, d\nu}{\int_{\mathbb{R}} S_x\, g_{\mu,\omega^\star}(\nu)\, d\nu} = \frac{\int_{\mathbb{R}} (\nu-\mu)^2 S_n\, g_{\mu,\omega^\star}(\nu)\, d\nu}{\int_{\mathbb{R}} S_n\, g_{\mu,\omega^\star}(\nu)\, d\nu},$$

i.e. the $g_{\mu,\omega^\star}$-weighted RMS detunings of signal and noise from the mean coincide. If $S_n(\mu) > 0$, then $\mathrm{SNR}(0^+) = S_x(\mu)/S_n(\mu)$, while

$$\mathrm{SNR}(\infty) = \frac{\int_{\mathbb{R}} S_x\, d\nu}{\int_{\mathbb{R}} S_n\, d\nu} > 0,$$

so the large-bandwidth limit never vanishes and existence is settled by the RMS detunings $\rho^2 := \int_{\mathbb{R}} (\nu-\mu)^2 S\, d\nu / \int_{\mathbb{R}} S\, d\nu$ instead: if moreover $S_x(\mu) = 0$ and $\rho_x^2 < \rho_n^2$, an interior maximizer $\omega^\star \in (0,\infty)$ exists. For $S_x(\mu) \neq 0$, no interior optimum is guaranteed.

Proof of Theorem A.3. Since x and n are independent, $h_i = s_i + \eta_i$ with $s_i = \phi * x$ and $\eta_i = \phi * n$ uncorrelated, hence

$$\mathrm{Var}(h_i) = V_x(\omega) + V_n(\omega).$$

Lemma A.2 with $W = S_x$ and $W = S_n$ gives

$$V_x' = \frac{V_x}{\omega^3}\big(\langle (\nu-\mu)^2 \rangle_x - \omega^2\big), \qquad V_n' = \frac{V_n}{\omega^3}\big(\langle (\nu-\mu)^2 \rangle_n - \omega^2\big),$$

where $\langle\cdot\rangle_x$ and $\langle\cdot\rangle_n$ denote the weighted averages of the lemma taken with $W = S_x$ and $W = S_n$ respectively. Since $V_n > 0$,

$$\mathrm{SNR}' = \frac{V_n V_x' - V_x V_n'}{V_n^2} = \frac{\mathrm{SNR}}{\omega^3}\big(\langle(\nu-\mu)^2\rangle_x - \langle(\nu-\mu)^2\rangle_n\big),$$

the $-\omega^2$ terms cancelling, and as $\mathrm{SNR} > 0$ every interior critical point satisfies $\langle(\nu-\mu)^2\rangle_x = \langle(\nu-\mu)^2\rangle_n$, which is the matching condition. By the lemma $V_x \to S_x(\mu)$ and $V_n \to S_n(\mu)$ as $\omega \to 0^+$, giving $\mathrm{SNR}(0^+) = S_x(\mu)/S_n(\mu)$ when $S_n(\mu) > 0$. For the opposite endpoint, $g_{\mu,\omega}(\nu)\sqrt{2\pi}\,\omega \to 1$ pointwise and is bounded by 1, so dominated convergence applied separately to numerators and denominators gives, using the finite second moments,

$$\langle(\nu-\mu)^2\rangle_x \to \rho_x^2, \qquad \langle(\nu-\mu)^2\rangle_n \to \rho_n^2, \qquad \mathrm{SNR}(\infty) = \frac{\int_{\mathbb{R}} S_x}{\int_{\mathbb{R}} S_n} > 0 \qquad (\omega \to \infty).$$

Suppose now $S_x(\mu) = 0$ and $\rho_x^2 < \rho_n^2$. Then $\mathrm{SNR}(0^+) = 0$, and since SNR' carries the sign of $\langle(\nu-\mu)^2\rangle_x - \langle(\nu-\mu)^2\rangle_n \to \rho_x^2 - \rho_n^2 < 0$, there is an ω_1 beyond which SNR is strictly decreasing. Fix $\omega_0 \in (0, \omega_1)$ and let $M := \max_{[\omega_0,\omega_1]} \mathrm{SNR} > 0$. Since $\mathrm{SNR}(0^+) = 0$ there is an $\epsilon \in (0, \omega_0)$ with $\mathrm{SNR} < M$ on $(0, \epsilon)$, and $\mathrm{SNR} < \mathrm{SNR}(\omega_1) \le M$ on (ω_1, ∞), so the supremum over $(0, \infty)$ is attained at an interior $\omega^\star \in [\epsilon, \omega_1]$, necessarily a solution of the matching condition. If $S_x(\mu) \neq 0$, the low-bandwidth limit is positive and does not guarantee an interior maximum. For $\mu = 0$, $g_{0,\omega} = G_{0,\omega}$ and the matching condition equates the weighted root-mean-square frequencies of signal and noise. □

A.5 Induced Random Fourier Feature kernels

We derive the second-order statistics of Random Fourier Features under three choices of frequency distribution, which yield the squared-exponential and sinc kernels of the random features literature (Rahimi & Recht, 2007); *these are well-known results, which we include for completeness's sake*. Since $\mathbf{\Omega}_i, \mathbf{\Upsilon}_i \in \mathbb{R}^{k\times d}$ have i.i.d. components, the induced kernel is identical, componentwise, across all (i, j, l); it therefore suffices to treat a single scalar component, and we drop the matrix indices throughout this subsection.

Lemma A.4 (Induced kernel of Random Fourier Features)**.** *Let $\Omega_1, \ldots, \Omega_m$ be i.i.d. real random variables with characteristic function $\varphi(\nu) = \mathbb{E}[e^{i\nu\Omega}]$, let $\Upsilon_1, \ldots, \Upsilon_m \overset{\text{i.i.d.}}{\sim} U(0, 2\pi)$ be independent of $\{\Omega_i\}$, and define the scalar feature map*

$$\phi(t) = \sqrt{\frac{2}{m}} \sum_{i=1}^{m} \cos(\Omega_i t + \Upsilon_i).$$

Then, for every $m \ge 1$ and all $t, t' \in \mathbb{R}$, writing $\Delta t = t - t'$,

$$\mathbb{E}[\phi(t)] = 0, \qquad \mathbb{E}\big[\phi(t)\phi(t')\big] = \mathbb{E}_\Omega[\cos(\Omega\Delta t)] \;=\; \mathrm{Re}\,\varphi\big(\Delta t\big),$$

so in particular $\mathbb{E}[\phi(t)^2] = 1$, which fixes the normalization $\sqrt{2/m}$. Moreover, the empirical kernel is consistent: $\phi(t)\phi(t') \to \mathrm{Re}\,\varphi(\Delta t)$ as $m \to \infty$, for each fixed pair (t, t').

Proof. Since $\Upsilon_i \sim U(0, 2\pi)$ independently of Ω_i, for every fixed Ω_i

$$\mathbb{E}_{\Upsilon_i}[\cos(\Omega_i t + \Upsilon_i)] = \frac{1}{2\pi}\int_0^{2\pi} \cos(\Omega_i t + \upsilon)\, d\upsilon = 0,$$

so $\mathbb{E}_\Upsilon[\phi(t)] = 0$ unconditionally on $\{\Omega_i\}$. For the second moment, the product-to-sum identity $\cos A \cos B = \frac{1}{2}[\cos(A-B) + \cos(A+B)]$ applied to $A = \Omega_i t + \Upsilon_i$, $B = \Omega_i t' + \Upsilon_i$ gives

$$\phi(t)\phi(t') = \frac{2}{m}\sum_{i=1}^{m} \cos(\Omega_i t + \Upsilon_i)\cos(\Omega_i t' + \Upsilon_i) \tag{11}$$

$$= \frac{1}{m}\sum_{i=1}^{m}\Big[\cos\big(\Omega_i(\Delta t)\big) + \cos\big(\Omega_i(t+t') + 2\Upsilon_i\big)\Big]. \tag{12}$$

Taking the expectation over Υ_i removes the second term by the same full-period argument as above (the extra additive constant $\Omega_i(t+t')$ inside the cosine does not affect the vanishing of its average over $2\Upsilon_i$), leaving $\mathbb{E}_\Upsilon[\phi(t)\phi(t')] = \frac{1}{m}\sum_{i=1}^m \cos(\Omega_i \Delta t)$. Taking the expectation over the i.i.d. Ω_i gives $\mathbb{E}_\Omega[\cos(\Omega\Delta t)]$ for every m, which is by definition the real part of the characteristic function of Ω evaluated at Δt; setting $\Delta t = 0$ gives $\mathbb{E}[\phi(t)^2] = 1$. The summands are i.i.d. and bounded by 1, so the strong law of large numbers gives the stated convergence as $m \to \infty$. □

Corollary A.5 (Zero-mean Gaussian frequencies)**.** *If $\Omega_i \sim \mathcal{N}(0, \omega^2)$, then $\varphi(\nu) = e^{-(\omega)^2\nu^2/2}$ is real and*

$$\mathbb{E}_\Omega[\cos(\Omega\Delta t)] = e^{-(\omega)^2(\Delta t)^2/2}.$$

Proof. The characteristic function of $\mathcal{N}(0, \sigma^2)$ is $\varphi(\nu) = e^{-\sigma^2\nu^2/2}$; substituting $\sigma = \omega$, $\nu = \Delta t$ gives the claim. The distribution is symmetric about 0, so φ is already real and $\operatorname{Re}\varphi = \varphi$. □

Corollary A.6 (Gaussian frequencies)**.** *If $\Omega_i \sim \mathcal{N}(\mu_\omega, \omega^2)$, then*

$$\mathbb{E}_\Omega[\cos(\Omega\Delta t)] = \cos\big(\mu_\omega \Delta t\big)\, e^{-(\omega)^2(\Delta t)^2/2}.$$

Proof. The characteristic function of $\mathcal{N}(\mu, \sigma^2)$ is $\varphi(\nu) = e^{i\mu\nu - \sigma^2\nu^2/2}$, so $\operatorname{Re}\varphi(\nu) = \cos(\mu\nu)\, e^{-\sigma^2\nu^2/2}$; substituting $\mu = \mu_\omega$, $\sigma = \omega$, $\nu = \Delta t$ gives the claim. This reduces to Corollary A.5 when $\mu_\omega = 0$. □

Corollary A.7 (Uniform frequencies)**.** *If $\Omega_i \sim U(-\omega, \omega)$, then*

$$\mathbb{E}_\Omega[\cos(\Omega\Delta t)] = \frac{\sin\big(\omega\Delta t\big)}{\omega\Delta t}.$$

Proof. Integrating directly,

$$\mathbb{E}_\Omega[\cos(\Omega\Delta t)] = \frac{1}{2\omega}\int_{-\omega}^{\omega} \cos(\Omega\Delta t)\, d\Omega \tag{13}$$

$$= \frac{1}{2\omega}\left[\frac{\sin(\Omega\Delta t)}{\Delta t}\right]_{-\omega}^{\omega} \tag{14}$$

$$= \frac{\sin(\omega\Delta t)}{\omega\Delta t}. \tag{15}$$

The uniform distribution is symmetric about 0, so this value is already real and $\operatorname{Re}\varphi = \varphi$. □

A.6 The exact kernel of TUNE

In Section 3.2 the TUNE kernel was only characterized by its dependencies, $k_0(t+t')k_1(t^2 + t'^2)k_s(\Delta t)$; here we retrieve it exactly for the distributions used in TUNE. As in Appendix A.5 the channels are i.i.d., so we treat a single component $\bar\phi(t)$ of $\bar{\boldsymbol{\Phi}}_{\boldsymbol{\Psi}}(t)$ and drop the matrix indices:

$$\bar\phi(t) = \sqrt{\frac{2}{md}}\sum_{i=1}^{m} \tilde\Lambda_i^{\,t}\, e^{-\frac{1}{2}\Theta_i^2(t-\Xi)^2}\cos(\Omega_i t + \Upsilon_i), \tag{16}$$

where $\Omega_i \sim \mathcal{N}(\mu_\omega, \omega^2)$, $\Upsilon_i \sim U(0, 2\pi)$, $\tilde\Lambda_i \sim U(\lambda_{min}, \lambda_{max})$ and $\Theta_i \sim \mathcal{N}(\mu_\theta, \theta^2)$ are i.i.d. over the m features, and the biased time point $\boldsymbol{\Xi}_{i,j} \sim U(0, T)$ is sampled once per channel. We write $\bar t = \frac{1}{2}(t+t')$.

Theorem A.8 (Exact kernel of TUNE)**.** *For every $m \geq 1$, $\mathbb{E}[\bar\phi(t)] = 0$ and*

$$\mathbb{E}\big[\bar\phi(t)\bar\phi(t') \,\big|\, \Xi\big] = \frac{1}{d}\underbrace{\frac{\lambda_{max}^{t+t'+1} - \lambda_{min}^{t+t'+1}}{(\lambda_{max} - \lambda_{min})(t+t'+1)}}_{k_0(t+t')}\underbrace{\frac{e^{-\mu_\theta^2 a/(1+2\theta^2 a)}}{\sqrt{1+2\theta^2 a}}}_{k_1(t,t';\Xi)}\underbrace{\cos(\mu_\omega\Delta t)\, e^{-\frac{1}{2}\omega^2\Delta t^2}}_{k_2(\Delta t)}, \tag{17}$$

with $a = (\Xi - \bar t)^2 + \frac{1}{4}\Delta t^2$, and $k_0(t+t') = \lambda^{t+t'}$ if $\lambda_{min} = \lambda_{max} = \lambda$. The kernel therefore reads $\boldsymbol{K}(t,t') = k_0(t+t')\,\bar k_1(t,t')\,k_s(\Delta t)\,\boldsymbol{I}$, with $\bar k_1(t,t') = \frac{1}{T}\int_0^T k_1(t,t';\xi)\,d\xi$.

Proof. **Mean and cross terms.** Each phase Υ_i is independent of all other variables, so every feature has zero mean and every cross term $i \neq i'$ vanishes (Lemma A.4), also conditionally on the shared Ξ. The m remaining diagonal terms are identically distributed and cancel the $1/m$ of the normalization, for every m.

Factorization. Since $\mathbb{E}_\Upsilon[\cos(\Omega t + \Upsilon)\cos(\Omega t' + \Upsilon)] = \frac{1}{2}\cos(\Omega \Delta t)$ and Ω, $\tilde{\Lambda}$, Θ are independent, a diagonal term factorizes into $\mathbb{E}_\Omega[\cos(\Omega \Delta t)] = k_s(\Delta t)$ (Corollary A.6), $\mathbb{E}[\tilde{\Lambda}^{t+t'}] = \frac{1}{\lambda_{max}-\lambda_{min}} \int_{\lambda_{min}}^{\lambda_{max}} \lambda^{t+t'} d\lambda = k_0(t+t')$, and an envelope term.

Envelope. Since $(t-\Xi)^2 + (t'-\Xi)^2 = 2a$, the envelope term is $\mathbb{E}[e^{-a\Theta^2}]$; completing the square in the Gaussian integral over $\Theta \sim \mathcal{N}(\mu_\theta, \theta^2)$,

$$\mathbb{E}\big[e^{-a\Theta^2}\big] = \int_{\mathbb{R}} \frac{e^{-ax^2-(x-\mu_\theta)^2/(2\theta^2)}}{\sqrt{2\pi\theta^2}}\, dx = \frac{1}{\sqrt{1+2\theta^2 a}} \exp\Big(-\frac{\mu_\theta^2 a}{1+2\theta^2 a}\Big) = k_1(t,t';\Xi).$$

Averaging over Ξ and summing the d i.i.d. components of a row yields $\boldsymbol{K}(t,t')$. □

Corollary A.9 (Closed forms of $\bar{k}_1$)**.** *$\bar{k}_1$ depends on (t,t') only through $\bar{t}$ and Δt. For $\theta = 0$ and $\mu_\theta > 0$,*

$$\bar{k}_1(t,t') = e^{-\frac{1}{4}\mu_\theta^2 \Delta t^2}\, \frac{\sqrt{\pi}}{2\mu_\theta T}\Big[\operatorname{erf}\big(\mu_\theta(T-\bar{t})\big) + \operatorname{erf}\big(\mu_\theta \bar{t}\big)\Big], \tag{18}$$

and for $\mu_\theta = 0$ and $\theta > 0$, with $\beta = \sqrt{1+\frac{1}{2}\theta^2\Delta t^2}$,

$$\bar{k}_1(t,t') = \frac{1}{\sqrt{2}\,\theta T}\Big[\operatorname{arsinh}\Big(\frac{\sqrt{2}\,\theta(T-\bar{t})}{\beta}\Big) + \operatorname{arsinh}\Big(\frac{\sqrt{2}\,\theta\,\bar{t}}{\beta}\Big)\Big]. \tag{19}$$

Proof. a depends on (t,t') only through $\bar{t}$ and Δt. For $\theta = 0$, $k_1 = e^{-\frac{1}{4}\mu_\theta^2\Delta t^2} e^{-\mu_\theta^2(\Xi-\bar{t})^2}$, whose average over $\Xi \sim U(0,T)$ is an error function; for $\mu_\theta = 0$, $k_1 = \big(\beta^2 + 2\theta^2(\Xi-\bar{t})^2\big)^{-1/2}$, which integrates to arsinh after the substitution $u = \sqrt{2}\,\theta(\Xi-\bar{t})/\beta$. □

A.7 Echo State Property of TUNE

Theorem A.10 (Echo State Property)**.** *Let $\bar{\boldsymbol{\Phi}}_{\boldsymbol{\Psi}}(t) = \sqrt{\frac{2}{md}} \sum_{i=1}^m \boldsymbol{\Psi}_i(t) \circ \cos(\boldsymbol{\Omega}_i t + \boldsymbol{\Upsilon}_i)$ with $\boldsymbol{\Psi}_i(t) = \tilde{\boldsymbol{\Lambda}}_i^t e^{-\frac{1}{2}\boldsymbol{\Theta}_i^2(t-\boldsymbol{\Xi})^2}$ and $\lambda_{max} := \max_{i,j,l}(\tilde{\boldsymbol{\Lambda}}_i)_{j,l} < 1$. Then for every uniformly bounded input sequence the hidden state map $\boldsymbol{h}_t = c\sum_{t'=0}^\infty \bar{\boldsymbol{\Phi}}_{\boldsymbol{\Psi}}(t')\boldsymbol{x}_{t-t'}$ induced by this decaying non-stationary GP is well defined (the series converges absolutely) and uniformly state contracting, and therefore possesses the ESP.*

Proof. Fix a realization of $\boldsymbol{\Omega}_i, \boldsymbol{\Upsilon}_i, \boldsymbol{\Theta}_i, \tilde{\boldsymbol{\Lambda}}_i, \boldsymbol{\Xi}$. Since $|\cos(\cdot)| \leq 1$ and $\sup_{\boldsymbol{\Theta}_{i,j,l}, \boldsymbol{\Xi}_{j,l}} e^{-\frac{1}{2}\boldsymbol{\Theta}_{i,j,l}^2(t'-\boldsymbol{\Xi}_{j,l})^2} = 1$, every component of $\bar{\boldsymbol{\Phi}}_{\boldsymbol{\Psi}}(t')$ obeys

$$\big|(\bar{\boldsymbol{\Phi}}_{\boldsymbol{\Psi}}(t'))_{j,l}\big| \leq \sqrt{\tfrac{2}{md}} \sum_{i=1}^m \lambda_{max}^{t'} = \sqrt{\tfrac{2m}{d}}\, \lambda_{max}^{t'},$$

so the induced $\ell_\infty \to \ell_\infty$ operator norm, i.e. the maximum absolute row sum over the d columns, satisfies

$$\big\|\bar{\boldsymbol{\Phi}}_{\boldsymbol{\Psi}}(t')\big\|_\infty \leq C\lambda_{max}^{t'}, \qquad C := \sqrt{2md}.$$

Now let $\boldsymbol{x}, \boldsymbol{x}'$ be two left-infinite input sequences with $\|\boldsymbol{x}\|_\infty, \|\boldsymbol{x}'\|_\infty \leq M$ that agree on the last q steps, i.e. $\boldsymbol{x}_{t-t'} = \boldsymbol{x}'_{t-t'}$ for $0 \leq t' < q$ (we write q rather than k, as k denotes the reservoir size

throughout). Then only lags $t' \geq q$ contribute and

$$\left\|\boldsymbol{h}_t(\boldsymbol{x}) - \boldsymbol{h}_t(\boldsymbol{x}')\right\|_\infty = \left\| c \sum_{t'=q}^{\infty} \bar{\boldsymbol{\Phi}}_{\boldsymbol{\Psi}}(t') \left(\boldsymbol{x}_{t-t'} - \boldsymbol{x}'_{t-t'}\right) \right\|_\infty \tag{20}$$

$$\leq c \sum_{t'=q}^{\infty} \left\|\bar{\boldsymbol{\Phi}}_{\boldsymbol{\Psi}}(t')\right\|_\infty \left\|\boldsymbol{x}_{t-t'} - \boldsymbol{x}'_{t-t'}\right\|_\infty \tag{21}$$

$$\leq 2McC \sum_{t'=q}^{\infty} \lambda_{max}^{t'} = \frac{2McC}{1-\lambda_{max}} \lambda_{max}^{q}, \tag{22}$$

a bound that is uniform in t and in the two input sequences. Since $\lambda_{max} < 1$, for every $\delta > 0$ it suffices to take

$$q_\delta = \left\lceil \frac{1}{\ln(1/\lambda_{max})} \ln\left(\frac{2McC}{\delta\,(1-\lambda_{max})}\right) \right\rceil$$

so that

$$\left\|\boldsymbol{h}_t(\boldsymbol{x}) - \boldsymbol{h}_t(\boldsymbol{x}')\right\|_\infty < \delta \qquad \text{for all } t \text{ whenever the inputs agree over the last } q \geq q_\delta \text{ steps.}$$

Thus, $\boldsymbol{h}_t$ is uniformly state contracting. □

A.8 Generalization to multi-variable Gaussian Processes

Matrix-valued GPs can be constructed by treating the first (and potentially second) index as a real variable:

$$\bar{\Phi}(t, x) = \sqrt{\frac{2}{md}} \sum_{i=1}^{m} \cos(\Omega_i^t t + \Omega_i^x x + \Upsilon_i), \tag{23}$$

where $\Omega^t, \Omega^x \in \mathbb{R}^d$, which induces the kernel

$$\mathbb{E}[\bar{\Phi}(t, x)\bar{\Phi}(t', x')^\top] = \mathbb{E}_{\Omega^t, \Omega^x}\left[\cos\left(\Omega^t \Delta t + \Omega^x \Delta x\right)\right] \tag{24}$$

$$\sim k(\Delta t, \Delta x). \tag{25}$$

If $\mathrm{Var}(\Omega^x) \to \infty$, one retrieves independent samples drawn from the same approximate kernel. This perspective allows for kernels that are not achievable by i.i.d. $\boldsymbol{B}$ linear ESNs, because the spatial covariance $\mathbb{E}\left[(e^{\boldsymbol{A}t}\boldsymbol{B})_i (e^{\boldsymbol{A}t}\boldsymbol{B})_{:,i}^\top\right] = \delta(\Delta i)$ by i.i.d. sampling of $\boldsymbol{B}$. Note that this does not, in general, hold for $\mathbb{E}\left[(e^{\boldsymbol{A}t}\boldsymbol{B})_i (e^{\boldsymbol{A}t'}\boldsymbol{B})_{:,i'}^\top\right] \sim k(i, i', \Delta t)$, which demonstrates a causal dependence that is not present in the GP. To achieve spatio-temporal behaviour one needs to create correlations between the matrices acting on $\boldsymbol{x}$ and t.

B Non-linear Echo State Networks

In this study we compare the novel TUNE against linear ESNs and non-linear ESNs. The latter is only stated briefly in Section 2; we define it here in full[5]:

$$\boldsymbol{h}_t = (1-\alpha)\boldsymbol{h}_{t-1} + \alpha\sigma\left(\tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}_{t-1} + s\boldsymbol{B}\boldsymbol{x}_t + s\boldsymbol{b}\right), \tag{26}$$

$$\boldsymbol{o}_t = \boldsymbol{W}\boldsymbol{h}_t + \boldsymbol{d}, \tag{27}$$

where $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ is some input data with $\boldsymbol{x}_t \in \mathbb{R}^d$, the bias $\boldsymbol{b} \in \mathbb{R}^k$ has components $\boldsymbol{b}_i \sim \mathcal{N}(0, 1)$, s is a hyperparameter which we will refer to as the hidden state norm, α is a scalar that determines leakage, and σ is some bounded non-linear activation function. The scalar $\tilde{\lambda} \in [0, 1]$ determines the decay rate, $\tilde{\boldsymbol{A}} \in \mathbb{R}^{k\times k}$ is a transfer matrix with a spectral radius $\rho(\tilde{\boldsymbol{A}}) = 1$, and $\boldsymbol{B} \in \mathbb{R}^{k\times d}$ is the input matrix with $\boldsymbol{B}_{i,j} \sim \mathcal{N}(0, 1/d)$. Throughout the appendix, as in the result tables, "ESN" refers to this non-linear ESN.

[5]Continuous-time non-linear ESNs also exist, but are not of importance to this study.

B.1 Reservoir topologies of the ESN and linear ESN

Corollaries 3.2 and 3.3 are results on $\boldsymbol{A}$ rather than on the transfer matrix $\tilde{\boldsymbol{A}} = e^{\boldsymbol{A}}$, so we build every reservoir topology by drawing $\boldsymbol{A}$ with the structure named in the corresponding row of Table 1 and setting $\tilde{\boldsymbol{A}} = e^{\boldsymbol{A}}/\rho(e^{\boldsymbol{A}})$. Table 4 collects the four draws.

Reservoir topology	Draw of $\boldsymbol{A}$	Kernel
Random	$\boldsymbol{G}/\sqrt{k}$	non-stationary
Normal	$\boldsymbol{Q}\big(\bigoplus_m a_m \boldsymbol{I}_2 + b_m \mathbf{J}\big)\boldsymbol{Q}^\top$	locally stationary
Orthogonal	$(\boldsymbol{G} - \boldsymbol{G}^\top)/2\sqrt{k}$	stationary
Simple-cycle	not drawn: $\tilde{\boldsymbol{A}} = \boldsymbol{P}$	stationary, equidistant

Table 4: The four topologies, with $\boldsymbol{G}_{i,j} \sim \mathcal{N}(0,1)$ i.i.d., $\boldsymbol{Q}$ orthogonal, $\mathbf{J} = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$, $a_m, b_m \sim \mathcal{N}(0,1)$ i.i.d. for $m \leq \lfloor k/2 \rfloor$, and $\boldsymbol{P}$ the cyclic permutation. Normal sampling puts $\boldsymbol{H}$ and $\boldsymbol{S}$ in a shared eigenbasis, so $[\boldsymbol{H}, \boldsymbol{S}] = \mathbf{0}$ holds by construction and $\tilde{\boldsymbol{A}}$ has eigenvalues $e^{a_m} e^{\pm i b_m}$: one decay rate and one frequency per mode, drawn independently.

B.2 A novel hidden state initialization

Initialization stability is important for non-linear ESNs with $\tilde{\lambda} \approx 1$, since they propagate information of the transient phase, diluting information of early steps. This is not detrimental for time series, as this stabilizing period can be left out of the training data. In classification tasks, however, leaving out the transient phase directly removes potentially crucial information; we therefore seek an informed initialization of the hidden state $\boldsymbol{h}_0$. Stable dynamics implies that for some left-infinite sequence $\{\boldsymbol{x}_0, \boldsymbol{x}_{-1}, \dots\}$ with $\boldsymbol{x}_{-t} = \boldsymbol{x}_0$ we have $\boldsymbol{h}_{-t+1} = \boldsymbol{h}_{-t}$ for all $t \in \mathbb{N}$. We also assume that $\tilde{\lambda}$ and the hidden state norm s are jointly small, of comparable order, allowing for a perturbative treatment:

$$\boldsymbol{h}_{-t} = (1-\alpha)\boldsymbol{h}_{-t} + \alpha\sigma\left(\tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}_{-t} + s\boldsymbol{B}\boldsymbol{x}_{-t+1} + s\boldsymbol{b}\right) \tag{28}$$

$$= \sigma(\tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}_{-t} + s\boldsymbol{B}\boldsymbol{x}_{-t+1} + s\boldsymbol{b}) \tag{29}$$

Assuming an activation function σ satisfying $\sigma(0) = \mathbf{0}$, the exact fixed-point equation for $\boldsymbol{h}_0 = \boldsymbol{h}_{-1}$ becomes:

$$\boldsymbol{h}_0 = \sigma(\tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}_0 + s\boldsymbol{B}\boldsymbol{x}_0 + s\boldsymbol{b}) \tag{30}$$

Collecting the two $\mathcal{O}(s)$-forcing terms into $\boldsymbol{c} \equiv \boldsymbol{B}\boldsymbol{x}_0 + \boldsymbol{b}$:

$$\boldsymbol{h}_0 = \sigma\left(\tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}_0 + s\boldsymbol{c}\right) \tag{31}$$

Since $\tilde{\lambda}$ and s are both treated as small, we expand $\boldsymbol{h}_0$ by the combined degree n in $(\tilde{\lambda}, s)$:

$$\boldsymbol{h}_0 = \boldsymbol{h}^{(1)} + \boldsymbol{h}^{(2)} + \boldsymbol{h}^{(3)} + \mathcal{O}(4), \tag{32}$$

where $\boldsymbol{h}^{(n)}$ collects all monomials $\tilde{\lambda}^a s^b$ with $a + b = n$. Substituting into the fixed-point argument and grouping by combined degree gives $\boldsymbol{z} \equiv \tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}_0 + s\boldsymbol{c} = \boldsymbol{z}_1 + \boldsymbol{z}_2 + \boldsymbol{z}_3 + \mathcal{O}(4)$ with

$$\boldsymbol{z}_1 = s\boldsymbol{c}, \qquad \boldsymbol{z}_2 = \tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}^{(1)}, \qquad \boldsymbol{z}_3 = \tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}^{(2)}. \tag{33}$$

Using the Taylor series expansion of σ about $z = \mathbf{0}$ up to order 3, $\sigma(z) = \sigma'(0)\, z + \frac{1}{2}\sigma''(0)\, z^{\odot 2} + \frac{1}{6}\sigma'''(0)\, z^{\odot 3} + \mathcal{O}(z^{\odot 4})$, and substituting $\boldsymbol{z} = \boldsymbol{z}_1 + \boldsymbol{z}_2 + \boldsymbol{z}_3$ component-wise, the required Hadamard powers expand as

$$\boldsymbol{z}^{\odot 2} = \boldsymbol{z}_1^{\odot 2} + 2\,\boldsymbol{z}_1 \odot \boldsymbol{z}_2 + \mathcal{O}(4), \qquad \boldsymbol{z}^{\odot 3} = \boldsymbol{z}_1^{\odot 3} + \mathcal{O}(4), \tag{34}$$

so that, collecting terms of matching combined degree,

$$\boldsymbol{h}_0 \approx \underbrace{\sigma'(0)\,\boldsymbol{z}_1}_{\text{degree 1}} + \underbrace{\sigma'(0)\,\boldsymbol{z}_2 + \frac{\sigma''(0)}{2}\,\boldsymbol{z}_1^{\odot 2}}_{\text{degree 2}} + \underbrace{\sigma'(0)\,\boldsymbol{z}_3 + \sigma''(0)\,\boldsymbol{z}_1 \odot \boldsymbol{z}_2 + \frac{\sigma'''(0)}{6}\,\boldsymbol{z}_1^{\odot 3}}_{\text{degree 3}}. \tag{35}$$

Equating this term-by-term with $\boldsymbol{h}_0 = \boldsymbol{h}^{(1)} + \boldsymbol{h}^{(2)} + \boldsymbol{h}^{(3)}$:

$$\begin{aligned}
&\text{degree } 1: \ \boldsymbol{h}^{(1)} = \sigma'(0)\, s\boldsymbol{c} \\
&\text{degree } 2: \ \boldsymbol{h}^{(2)} = \sigma'(0)\, \tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}^{(1)} + \frac{\sigma''(0)}{2}\,(s\boldsymbol{c})^{\odot 2} \\
&\text{degree } 3: \ \boldsymbol{h}^{(3)} = \sigma'(0)\, \tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}^{(2)} + \sigma''(0)\,(s\boldsymbol{c}) \odot \left(\tilde{\lambda}\tilde{\boldsymbol{A}}\boldsymbol{h}^{(1)}\right) + \frac{\sigma'''(0)}{6}\,(s\boldsymbol{c})^{\odot 3}
\end{aligned} \tag{36}$$

This yields the solution up to combined order 3:

$$\begin{aligned}
\boldsymbol{h}_0 \approx{}& \sigma'(0)\, s\boldsymbol{c} + \sigma'(0)^2\tilde{\lambda} s\, \tilde{\boldsymbol{A}}\boldsymbol{c} + \frac{\sigma''(0)}{2} s^2 \boldsymbol{c}^{\odot 2} \\
&+ \sigma'(0)^3 \tilde{\lambda}^2 s\, (\tilde{\boldsymbol{A}})^2 \boldsymbol{c} + \frac{\sigma'(0)\sigma''(0)}{2} \tilde{\lambda} s^2\, \tilde{\boldsymbol{A}}\left(\boldsymbol{c}^{\odot 2}\right) && (37) \\
&+ \sigma'(0)\sigma''(0)\, \tilde{\lambda} s^2 \left(\boldsymbol{c} \odot (\tilde{\boldsymbol{A}}\boldsymbol{c})\right) + \frac{\sigma'''(0)}{6} s^3 \boldsymbol{c}^{\odot 3} && (38)
\end{aligned}$$

which is the fixed-point approximation for a jointly small $(\tilde{\lambda}, s)$, for a general activation function σ and general constant input $\boldsymbol{x}_0$. We now specialize to a hyperbolic tangent activation[6], $\sigma(z) = \tanh(z)$, for which $\sigma'(0) = 1$, $\sigma''(0) = 0$, $\sigma'''(0) = -2$, so that every term carrying $\sigma''(0)$ vanishes and $\boldsymbol{h}^{(2)} = \tilde{\lambda} s\, \tilde{\boldsymbol{A}}\boldsymbol{c}$. We further set $\boldsymbol{x}_0 = \boldsymbol{0}$, so that $\boldsymbol{c} = \boldsymbol{B}\boldsymbol{x}_0 + \boldsymbol{b} = \boldsymbol{b}$. Substituting these into the general result above, and truncating the result, gives

$$\boldsymbol{h}_0 \approx \mathrm{clip}_{-1}^{1}\left(s\left(\boldsymbol{I} + \tilde{\lambda}\tilde{\boldsymbol{A}} + \tilde{\lambda}^2(\tilde{\boldsymbol{A}})^2\right)\boldsymbol{b} \ - \ \frac{s^3}{3}\boldsymbol{b}^{\odot 3}\right). \tag{39}$$

In the experimental setting we let this initial hidden state approximation be forwarded 10 iterations with an all-zero input sequence; Figure 4 verifies the approximation.

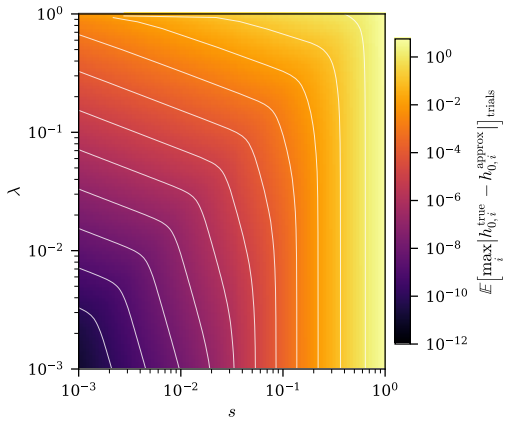


Figure 4: Experimental verification of the unclipped $\boldsymbol{h}_0$ initialization strategy for Gaussian random $\tilde{\boldsymbol{A}}$.

C ParalESN as a sampled Gaussian Process

The closest architecture to ours is the ParalESN of (Pinna et al., 2026), which deploys a diagonal transfer matrix. It is therefore worth stating precisely where it sits in the framework of Section 3.2. We first restate the model in our notation, then identify the kernel it samples, and finally explain why only its shallow variant is rerun here.

[6]Since $\tanh$ is odd, $\sigma''(0) = 0$ but $\sigma^{(4)}(0) = 0$ as well; nonetheless, because $\tilde{\lambda}$ is no longer resummed exactly, the next non-vanishing contribution appears at combined degree 4 rather than being absorbed as in the $O(1)$-$\tilde{\lambda}$ case, so this truncation is only accurate to $\mathcal{O}(4)$ in $(\tilde{\lambda}, s)$ jointly.

C.1 The model in our notation

A ParalESN layer of width k driven by $\boldsymbol{x}_t \in \mathbb{R}^d$ is given by Eqs. (4)–(6) of (Pinna et al., 2026). It has a diagonal complex transition matrix $\boldsymbol{\Lambda} = \mathrm{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_k)$ with

$$\tilde{\lambda}_n = \rho_n e^{i\theta_n}, \qquad \rho_n \sim U(\rho_{min}, \rho_{max}), \qquad \theta_n \sim U(\theta_{min}, \theta_{max}), \tag{40}$$

a leaky rate $\tau \in (0, 1]$ that folds into an effective transition $\boldsymbol{\lambda} = (1-\tau)\mathbf{1} + \tau\,\mathrm{diag}(\boldsymbol{\Lambda})$, a complex input matrix $\boldsymbol{B} \in \mathbb{C}^{k\times d}$ whose row n is scaled by $\sqrt{1-|\lambda_n|^2}$, and a recurrence bias $\boldsymbol{b}$. The hidden state and readout read

$$\begin{aligned} \boldsymbol{h}_t &= \boldsymbol{\lambda} \circ \boldsymbol{h}_{t-1} + \tau\left(\boldsymbol{B}\boldsymbol{x}_t + \boldsymbol{b}\right), \\ \boldsymbol{o}_t &= \boldsymbol{W}\tanh\Big(\mathrm{Re}\big(\boldsymbol{W}_{mix} * \boldsymbol{h}_t + b_{mix}\big)\Big) + \boldsymbol{d}, \end{aligned} \tag{41}$$

where $*$ is a one-dimensional convolution of width κ *across the hidden dimension*, shared over time.[7] Because $\boldsymbol{\Lambda}$ is diagonal, the recurrence is an component-wise causal convolution, which the original parallelizes with an associative scan; the solved form of Section 2 therefore applies verbatim with the response function

$$\boldsymbol{\Phi}_t = \tau\,\mathrm{diag}(\boldsymbol{\lambda})^t\boldsymbol{B}, \qquad \boldsymbol{h}_t = \sum_{t'=0}^{T} \boldsymbol{\Phi}_{t'}\boldsymbol{x}_{t-t'} + \tau\boldsymbol{b}\sum_{t'=0}^{T}\boldsymbol{\lambda}^{t'}. \tag{42}$$

Since $|\lambda_n| \le (1-\tau) + \tau|\tilde{\lambda}_n| < 1$ whenever $\rho_{max} < 1$, the effective transition is contracting for every leaky rate.

C.2 The Gaussian Process it samples

Writing $\boldsymbol{B}_{n,j} = |\boldsymbol{B}_{n,j}|e^{i\varphi_{n,j}}$ and taking the real part that the readout layer sees, component (n, j) of equation 42 is

$$\mathrm{Re}\left(\boldsymbol{\Phi}_t\right)_{n,j} = \tau\,|\boldsymbol{B}_{n,j}|\;\rho_n^t\cos(\theta_n t + \varphi_{n,j}), \tag{43}$$

which is one term of the expansion of Section 3.2.[8] Taking the expectation over the reservoir draw, the covariance factorizes as

$$\mathbb{E}\left[\boldsymbol{\Phi}_t\boldsymbol{\Phi}_{t'}^\top\right] \;\sim\; \underbrace{\frac{\rho_{max}^{t+t'+1} - \rho_{min}^{t+t'+1}}{(\rho_{max}-\rho_{min})(t+t'+1)}}_{k_0(t+t')}\;\underbrace{\frac{\sin(\theta_{max}\Delta t) - \sin(\theta_{min}\Delta t)}{(\theta_{max}-\theta_{min})\Delta t}}_{k_s(\Delta t)}\;\boldsymbol{I}, \tag{44}$$

that is, a kernel of the form $k_0(t+t')k_s(\Delta t)\boldsymbol{I}$, and hence a locally stationary process.

Remark C.1. *ParalESN samples a* locally stationary *GP with a band-limited (uniform) spectral measure: it is the $\boldsymbol{\Theta} = \mathbf{0}$, $m = 1$ special case of the response function of Section 3.3, and sits in the "Normal" row of Table 1, as a diagonal transition matrix is normal. Due to the channel-wise convolution $\boldsymbol{W}_{mix}$ imposed in the hidden state construction, an effective Monte-Carlo sampling $\bar{m} = \kappa \le k$ is induced.*

TUNE generalizes this construction in three respects. The spectral measure is Gaussian with a tuneable bandwidth and mean rather than uniform on a band, which is the quantity Theorem A.3 identifies as the object to tune. The envelope $\boldsymbol{\Psi}$ carries the Gaussian factor $e^{-\frac{1}{2}\boldsymbol{\Theta}^2(t-\boldsymbol{\Xi})^2}$, whose per-channel time bias $\boldsymbol{\Xi}$ produces the non-stationary component that a decay alone cannot express. Independently of both, each channel averages m features instead of one: the frequency content of a ParalESN channel is a single frequency, whereas that of a TUNE channel is a spectral density. The price is the m-fold construction cost of Table 5, which leaves both models linear in k.

[7] We write κ for the mixing width, which the original calls k; here k is the reservoir size throughout.

[8] The phases are the arguments of a complex matrix with real and imaginary parts uniform on $[-1, 1]$, hence uniform on a square rather than on a disc; the resulting phase distribution is close to, but not exactly, $U(0, 2\pi)$.

C.3 Scope of the comparison

We rerun the shallow ParalESN under the same search budget, dataset splits and timing procedure as our own models. In a deep stack, each layer consumes the $\tanh$-mixed output $\boldsymbol{z}_t^{(\ell-1)}$ of the layer below through the ring input matrix of equation 5 of (Pinna et al., 2026), so the composite map from input to final state is no longer linear in the input and admits no response function $\boldsymbol{\Phi}_t$. The analysis of Section 3, and with it every statement in this appendix, applies to one layer; the deep variant is a different object, and is reported as its authors measured it.

D Computational complexity

Every model of this study is used in three stages. Its untrained part is *constructed* once per draw: the transfer matrix of the ESN, or the response function $\boldsymbol{\Phi}$ of the three linear models. Every window of the data is then passed through it, a *feature pass* that yields the state the readout sees. Finally, the ridge *readout* is fitted. The readout is the same for every model, so the models differ only in the first two stages; Table 5 collects their cost, which the rest of this section derives. Throughout, k is the reservoir size, d the input dimension, T the window length, m the number of random features of TUNE and n the number of readout outputs.

Construction

The ESN draws $\boldsymbol{A}$ with the structure of its reservoir topology and normalizes $\tilde{\boldsymbol{A}} = e^{\boldsymbol{A}}/\rho(e^{\boldsymbol{A}})$; the matrix exponential and the eigenvalues each cost $\mathcal{O}(k^3)$, and only $\tilde{\boldsymbol{A}}$ and $\boldsymbol{B}$ are kept, since its state is computed by the recurrence itself. The linear ESN makes the same draw and then unrolls it into the response function, $\boldsymbol{\Phi}_t = \tilde{\boldsymbol{A}}\boldsymbol{\Phi}_{t-1}$ for $t < T$: T products of a $k \times k$ with a $k \times d$ matrix, so $\mathcal{O}(k^3 + Tdk^2)$ in all and $\mathcal{O}(Tdk)$ to store. ParalESN avoids the square matrix altogether. Its transition is diagonal, so its impulse response is k scalar power series, $\mathcal{O}(Tk)$, projected onto the input in $\mathcal{O}(Tdk)$, plus the $\mathcal{O}(Tk \log T)$ transform of its sequence path. TUNE draws m features and sums them, each a component-wise product over the $T \times d \times k$ response: $\mathcal{O}(mTdk)$, with no matrix product of any kind.

Feature pass

The three linear models read a window out in the same way: $\boldsymbol{h}_T = c \sum_{t'=0}^{T} \boldsymbol{\Phi}_{t'} \boldsymbol{x}_{T-t'}$ contracts the Td components of the window against the response, $\mathcal{O}(Tdk)$ work in a single reduction of depth $\mathcal{O}(\log Td)$. ParalESN adds its mixing convolution across the hidden dimension, $\mathcal{O}(\kappa k)$, which does not change the order. The ESN must instead run its recurrence, T dependent updates of $\mathcal{O}(k(k+d))$ each, so its depth grows linearly with the window. For forecasting, the linear models cut the series into N overlapping windows and read out the final state of each, $\mathcal{O}(NTdk)$ in total, whereas the ESN carries its state through the series in a single pass of $\mathcal{O}(Nk(k+d))$: fewer operations whenever $Td > k + d$, but none of them parallel over time.

Readout

The readout has $(k + 1)n$ trainable parameters. Its normal equations are accumulated over the N_{tr} training windows in chunks, $\mathcal{O}(N_{tr}k(k + n))$ time in $\mathcal{O}(k(k + n))$ memory, so no design matrix is ever held whole. A single symmetric eigendecomposition, $\mathcal{O}(k^3)$, then serves every penalty the inner search tries; each costs $\mathcal{O}(k^2 n + N_{val}kn)$ to solve and to score on the N_{val} validation windows, for at most $S = 20$ penalties. The linear models stream their features and so compute them a second time for the training error, whereas the ESN, whose features come out of one recurrent pass, holds them instead, at $\mathcal{O}(Nk)$ memory. For the multi-horizon forecasts $n = Hd$, and the $N_{tr}kn$ term is what makes a long horizon on a high-dimensional series expensive.

Model difference

The feature pass is the same for the three linear models, so at a given width their inference costs the same and their difference lies in construction. TUNE is linear in k and involves no square matrix. The linear ESN pays $\mathcal{O}(k^3)$ to form $\tilde{\boldsymbol{A}}$ and a propagation that costs k/m times TUNE's sampling;

the cubic term dominates the propagation as soon as $k > Td$, which holds at the two largest widths of Table 2. A diagonal transition avoids the square matrix as well, which puts ParalESN in the same class as TUNE. The non-linear ESN is the only model whose feature pass is sequential in T.

	Construction (once per draw)		Feature pass (per window)	
Model	Time	Memory	Work	Depth
ESN	$\mathcal{O}(k^3)$	$\mathcal{O}(k(k+d))$	$\mathcal{O}(Tk(k+d))$	$\mathcal{O}(T \log k)$
Linear ESN	$\mathcal{O}(k^3+Tdk^2)$	$\mathcal{O}(Tdk)$	$\mathcal{O}(Tdk)$	$\mathcal{O}(\log Td)$
ParalESN	$\mathcal{O}(Tk(d+\log T))$	$\mathcal{O}(Tdk)$	$\mathcal{O}(Tdk)$	$\mathcal{O}(\log Td)$
TUNE	$\mathcal{O}(mTdk)$	$\mathcal{O}(Tdk)$	$\mathcal{O}(Tdk)$	$\mathcal{O}(\log Td)$
Readout (all)	$\mathcal{O}\big(N_{tr}k(k+n) + k^3 + S(k^2n + N_{val}kn)\big)$ time, $\mathcal{O}(k(k+n))$ memory			

Table 5: Cost of each model for reservoir size k, input dimension d, window length T and m Monte-Carlo features. *Construction* builds the untrained part once per draw; the *feature pass* reads one window out through it, and *depth* is that computation's critical path. The readout is shared by all models: $(k+1)n$ parameters fitted over N_{tr} training windows, with at most $S = 20$ ridge penalties scored on N_{val} validation windows.

E Experimental setup and datasets

E.1 Hardware and software environment

Every number reported in this paper, in the main text and in the appendix, was produced on a single desktop workstation: an AMD Ryzen 7 7700 (8 cores, 16 threads), 32 GB of system memory and one NVIDIA GeForce RTX 4070 Ti SUPER, running Windows 11. The experiments were conducted in Python 3.12 with PyTorch 2.12 against CUDA 13.0. All four architectures: the ESN, the linear ESN, ParalESN and TUNE, are implemented as dense tensors, and have their ridge readout trained on the GPU in the default `float32`; the memory capacity measurement of Appendix F.3 is the sole exception and runs in `float64`, because floating point errors did have noticeable impact.

E.2 Training and evaluation protocol

For `sMNIST` and `psMNIST` the official 60000 training and 10000 test images are pooled and re-split 90/10, stratified by class; the permutation of `psMNIST` is drawn once from a fixed seed, so every model sees the same permuted pixel order. The ETT datasets are split chronologically after the convention of the long-term forecasting literature: the first 12 months are training data and the last 4 months are test data, with the intervening months discarded, and every channel is standardized with the mean and standard deviation of the training portion alone. Within the training set the last 20% is held out as a fixed validation split, not reshuffled, and is the only data a trial ever scores itself on; the test set stays untouched.

A trial draws one hyperparameter configuration, builds the response function from it, fits the ridge readout, and returns the validation metric: accuracy for classification and mean squared error for forecasting. Each trial is run 5 times on independent draws of the same configuration and split, and is scored by the mean of those fits.

Once the search is over, its winning configuration is re-initialized, refitted 5 times and evaluated on the held-out test set. The reported $\pm$ is the standard deviation of the test scores of those five models. A cell whose single fit already exceeds a 10s budget is reported *o.o.t.* rather than searched, which is how the $\sim$50K and $\sim$160K ESN and linear ESN components of Table 2 are decided.

E.3 Ridge regression

In this study, Echo State Networks are trained via ridge regression, which has an exact closed-form solution. Let $\boldsymbol{Z} = [\boldsymbol{O} \quad \mathbf{1}^{N\times 1}] \in \mathbb{R}^{N\times(k+1)}$ be the feature matrix containing the non-linearly transformed hidden states $\boldsymbol{O}$ concatenated with a unit column for the bias across N time steps/samples, and let $\boldsymbol{Y} \in \mathbb{R}^{N\times n}$ be the ground truth target matrix. The readout weight matrix

$\hat{\boldsymbol{W}} = [\boldsymbol{W} \quad \boldsymbol{d}] \in \mathbb{R}^{(k+1)\times n}$ is obtained by solving:

$$\hat{\boldsymbol{W}} = \left(\boldsymbol{Z}^\top \boldsymbol{Z} + \gamma \boldsymbol{I}_{k+1}\right)^{-1} \boldsymbol{Z}^\top \boldsymbol{Y} \tag{45}$$

where $\gamma > 0$ is the Tikhonov regularization hyperparameter (the ridge penalty). In this study the optimal regularization hyperparameter γ is approximated via heuristic optimization, specifically via Brent's algorithm (Brent, 1973). The search is run over $\log_{10}\gamma \in [-10, 2]$ against the validation split, to a tolerance of 10^{-3} and at most 20 iterations, and is repeated inside every fit; γ is therefore never a hyperparameter of the outer search.

E.4 Hyperparameter ranges

The outer search is a Bayesian optimizer, run for 500 trials per cell; the sampler is multivariate, it evaluates 100 expected-improvement candidates per step, and it treats the best 10% of trials as the good set. Its warmup is 50 random trials; a search whose best value has not improved for 100 trials is stopped early.

Table 6 lists the searched ranges. Every model carries the hidden state norm s, and the ESN and linear ESN carry the reservoir topology as a categorical hyperparameter, so that the linear ESN is compared at its own best reservoir topology. For classification and regression the decay is confined to the top of its range, since a readout taken at $t = T$ alone needs a response function that has not already forgotten the start of the window; for forecasting, where the readout is per-step, the decay is searched over its full range.

Model	Hyperparameter	Symbol	Range
TUNE	frequency bandwidth	ω	$[10^{-4}, 10^{1}]$, log
	frequency mean	μ_ω	$[10^{-4}, 10^{1}]$, log
	envelope bandwidth	θ	$[10^{-4}, 10^{0}]$, log
	envelope mean	μ_θ	$[10^{-4}, 10^{0}]$, log
	decay range, lower	λ_{min}	$[0.9, 1]$ $([10^{-2}, 1])$
	decay range, upper	λ_{max}	1 $([10^{-2}, 1])$
	random features	m	$\{1, \ldots, 1000\}$
	hidden state norm	s	$[10^{-3}, 10^{2}]$, log
Linear ESN	decay rate	λ	1 $([10^{-2}, 1]$, log)
	topology	–	random, orthogonal, normal, permutation
	hidden state norm	s	$[10^{-3}, 10^{2}]$, log
ESN	decay rate	λ	$[0.9, 1]$ $([10^{-2}, 1]$, log)
	leak rate	α	$[0, 1]$
	topology	–	random, orthogonal, normal, permutation
	hidden state norm	s	$[10^{-5}, 10^{-1}]$, log
ParalESN	eigenvalue magnitude	ρ_{min}, ρ_{max}	$[0.9, 1]$ $([10^{-2}, 1])$
	eigenvalue phase	$\theta_{min}, \theta_{max}$	$[0, \pi]$
	leaky rate	τ	$[0.1, 1]$
	recurrence bias scale	ω_b	$[10^{-3}, 10^{1}]$, log
	mixing kernel scale	ω_{mix}	$[10^{-2}, 10^{1}]$, log
	mixing bias scale	ω_{mixb}	$[10^{-3}, 10^{1}]$, log
	mixing width	κ	$\{1, \ldots, 7\}$
	hidden state norm	s	$[10^{-3}, 10^{2}]$, log
all	ridge penalty	γ	$\log_{10}\gamma \in [-10, 2]$, inner search

Table 6: Searched hyperparameter ranges, at 500 trials per cell. A bracketed alternative is the range used for forecasting, where the readout is per-step; the unbracketed range is the one used for classification and regression, where the readout sees $\boldsymbol{h}_T$ alone. γ is fitted inside every trial rather than by the outer search.

E.5 TIMING AND ENERGY MEASUREMENT

The times reported beside every accuracy and every mean squared error are *per training iteration* of one controlled configuration; one fit of the winning configuration, measured from the start of the readout fit to its end and averaged over the fits the search performed. Energy and emissions are tracked over the whole search from CPU, GPU and RAM power draw with CodeCarbon (Courty et al., 2024), and then divided by the number of fits that search performed, so that they stand on the same per-iteration footing as the time.

E.6 BENCHMARK DATASETS

The benchmarks of Section 4 are described here.

E.6.1 KBUMPS

A collection of N random signals with localized information, generated by sampling an i.i.d. Gaussian vector of length T, which is subsequently added to $\{0, \dots, K_{max}\}$ random rectangular pulses of various widths:

$$x_i = \sum_{k=0}^{K_i} \text{rect}(|j - a_{i,k}| < b_{i,k}) + \epsilon_i, \tag{46}$$

where $i \in \{0, \dots, N-1\}$, $j \in \{0, \dots, T-1\}$, $\epsilon_{i,j} \sim \mathcal{N}(0, s^2)$, $K_i \sim \text{round}(U(0, K_{max}))$, $a_{i,k} \sim U(0, T)$, $b_{i,k} \sim U(w_{min}, w_{max})$, and $s \in \mathbb{R}^+$ is the noise level. The pulses are generated such that they do not overlap: a minimum separation $d_{i,k}$ between all pulse centers $|a_{i,j} - a_{i,k}|$ is imposed. The classification goal is to predict K_i from x_i, which is a non-trivial task since most of the sequence is filled with informationless noise. A performant model should be able to discern noise from information in the memorization process. Table 7 lists the dataset parameters.

Parameter	Dataset value
N	100000
N_{train}	$0.5N$
T	1000
w_{min}, w_{max}	20,100
s	1
K_{max}	3
$d_{i,k}$	$1.1w_{max}$

Table 7: `KBumps` dataset parameters.

E.6.2 BUMPDISTANCE

Similar to `KBumps`, `BumpDistance` is a collection of N random signals with localized information, generated by sampling an i.i.d. Gaussian vector of length T, which is subsequently added to two random rectangular pulses of various widths:

$$x_i = \text{rect}(|j - a_{i,0}| < b_{i,0}) + \text{rect}(|j - a_{i,1}| < b_{i,1}) + \epsilon_i, \tag{47}$$

where $i \in \{0, \dots, N-1\}$, $j \in \{0, \dots, T-1\}$, $\epsilon_{i,j} \sim \mathcal{N}(0, s^2)$, $a_{i,0;1} \sim U(0, T)$, $b_{i,0;1} \sim U(w_{min}, w_{max})$, and $s \in \mathbb{R}^+$ is the noise level. The pulses are generated such that they do not overlap: a minimum separation d_i between the pulse centers $|a_{i,0} - a_{i,1}|$ is imposed. The regression goal is to return the distance $|a_{i,0} - a_{i,1}|$ from the input signal x_i, which is again a non-trivial task since most of the sequence is filled with informationless noise. A performant model should be able to discern noise from information in the memorization process. Table 8 lists the dataset parameters.

E.6.3 SMNIST AND PSMNIST

The 70000 MNIST images are scaled to $[0, 1]$ and flattened row-major into scalar sequences, giving $d = 1$, $T_{\text{seq}} = 784$ and 10 classes; this is sequential MNIST, in which a model must carry pixel information across the full 784 steps to a readout taken at $t = T$ alone. The permuted variant

Parameter	Dataset value
N	100000
N_{train}	$0.5N$
T	1000
w_{min}, w_{max}	20,100
s	1
d_i	$1.1w_{max}$

Table 8: `BumpDistance` dataset parameters.

applies one fixed random permutation of the 784 positions to every image, which destroys the local correlation structure of the image and leaves a task that cannot be solved by short-range features alone. The reservoir size sets the trainable parameter count, $(k+1)n$, and the four budgets of Table 2 are $k = 100 \ll T_{\text{seq}}$, $k = 784 = T_{\text{seq}}$, $k = 0.8N_{\text{train}}/10$ and $k = 16000 \gg T_{\text{seq}}$.

E.6.4 ETT

The `Electricity` Transformer Temperature benchmark consists of four datasets recorded from two transformers: `ETTh1` and `ETTh2` at hourly resolution (17420 steps) and `ETTm1` and `ETTm2` at 15-minute resolution (69680 steps). Each carries $d = 7$ channels, six load features and the oil temperature. The task is autoregressive: from the state at step t the model predicts the next $1, \dots, H$ states jointly, so the target carries Hd columns, and the horizons of Table 3 are $H \in \{1, 96, 192\}$. All models use $k = 100$ and a response-function window $T = 0.01T_{\text{seq}}$, which is 86 steps on the hourly datasets and 346 on 15-minute ones.

F Additional experiments

F.1 Classification

Beyond `sMNIST` and `psMNIST`, the linear ESN, ParalESN and TUNE are evaluated on five time series classification datasets from the UEA (Bagnall et al., 2018) & UCR (Dau et al., 2019) archives. ParalESN has been previously evaluated on these classification datasets (Pinna et al., 2026). They are chosen to differ in length, dimensionality, training set size, and class balance. Each dataset uses the archive's own train/test split rather than a re-split. Table 9 reports the results.

Blink: EEG recorded with a four-electrode headband while subjects blinked, labeled by whether the blink was short or long: $d = 4$ channels of length $T_{\text{seq}} = 510$, 2 balanced classes, 500 training and 450 test time series. It is the only multivariate time series of the five, and the one with the fewest training time series.

FordA and FordB: Measurements of engine noise, labeled by whether a symptom is present in an automotive subsystem ($d = 1$, $T_{\text{seq}} = 500$, 2 classes). `FordA` (3601 training, 1320 test time series) records both splits under typical operating conditions. `FordB` (3636 training, 810 test time series) records its training time series under typical conditions and its test time series under noisy ones, so it measures how well a fitted model transfers across a shift between training and test time series.

StarLightCurves: Phase-folded brightness curves of variable stars, classified as Cepheid, RR Lyrae or eclipsing binary ($d = 1$, $T_{\text{seq}} = 1024$, 3 classes). Only 1000 time series are available for training against 8236 for testing, and the training classes are imbalanced (152, 275 and 573 time series).

FaultDetectionA: Signals recorded from an electromechanical drive system, classified by the state of its rolling bearing: undamaged, damaged inner ring or damaged outer ring ($d = 1$, 3 classes, 10912 training and 2728 test time series). At $T_{\text{seq}} = 5120$ its time series are the longest of any dataset in this study, which makes it the longest-range memory test of the five; its training classes are imbalanced as well (992 undamaged time series against 4960 of each damage type).

Width	Dataset	Linear ESN		ParalESN		TUNE	
		acc (%)	t (s)	acc (%)	t (s)	acc (%)	t (s)
$k = 100$	Blink	73.7 ± 7.8	**0.01**	$\underline{95.6} \pm 0.8$	$\underline{0.03}$	**96.3** ± 0.9	0.03
	FordA	65.0 ± 2.6	**0.02**	**74.4** ± 1.5	0.03	$\underline{70.0} \pm 0.5$	$\underline{0.03}$
	FordB	56.4 ± 2.3	**0.02**	$\underline{57.0} \pm 0.2$	0.03	**58.1** ± 0.5	$\underline{0.03}$
	StarLightCurves	87.4 ± 1.1	**0.01**	**89.1** ± 1.2	$\underline{0.03}$	$\underline{88.4} \pm 1.3$	0.03
	FaultDetectionA	58.8 ± 2.7	$\underline{0.05}$	$\underline{73.0} \pm 0.3$	**0.05**	**74.9** ± 0.7	0.07
$k = 1000$	Blink	95.8 ± 2.1	**0.03**	**97.5** ± 0.5	$\underline{0.04}$	$\underline{96.7} \pm 0.3$	0.14
	FordA	70.0 ± 0.7	**0.04**	$\underline{85.3} \pm 4.3$	0.04	**85.5** ± 4.6	$\underline{0.04}$
	FordB	56.4 ± 2.4	**0.04**	$\underline{70.4} \pm 0.8$	0.04	**76.1** ± 1.7	$\underline{0.04}$
	StarLightCurves	73.1 ± 25.5	**0.04**	$\underline{91.8} \pm 0.3$	$\underline{0.04}$	**95.7** ± 0.2	0.06
	FaultDetectionA	71.3 ± 1.7	$\underline{0.07}$	$\underline{75.5} \pm 0.6$	**0.06**	**87.0** ± 0.5	0.31

Table 9: Classification performance (test accuracy and training time per iteration) on the Blink, FordA, FordB, StarLightCurves and FaultDetectionA datasets, at a small ($k = 100$) and a large ($k = 1000$) reservoir size; every model uses the archive's own train/test split. **Bold** indicates the best result, underline the second best.

F.2 FORECASTING

The results of Section 4 in the main text on the ETT dataset collection are extended here by additional forecasting datasets: five real-world multivariate time series from the long-term forecasting literature, the NARMA10 task and five chaotic systems from the dysts benchmark. As for ETT, the model predicts the next $1, \dots, H$ states jointly from the state at step t, and every channel of a real-world time series is standardized with the mean and standard deviation of its training portion alone. These time series have no partition of their own and are split chronologically, the first 80% for training and the remaining 20% for testing. All models use a response-function window $T = 0.01 T_{\text{seq}}$, which ranges from 59 steps on Exchange to 420 on Weather and Solar. Weather and Exchange forecast $H = 96$ steps ahead, the shortest horizon of the long-term forecasting literature. The three high-dimensional time series forecast $H = 8$ steps ahead; the target carries Hd columns. NARMA10 and the chaotic systems are one-step ahead prediction tasks, and their errors are several orders of magnitude below those of the real-world time series. Table 10 reports the results.

Weather: 21 meteorological indicators, among them air temperature and humidity, recorded every 10 minutes throughout 2020 (52696 time steps) (Wu et al., 2022); the moderate multivariate case.

Exchange: Daily exchange rates of eight currencies from 1990 to 2016 (7588 time steps) (Wu et al., 2022). Close to a random walk and non-stationary, it is the dataset on which a long memory of the past is least useful.

Solar: Power output of 137 photovoltaic plants in Alabama, recorded every 10 minutes in 2006 (52560 time steps) (Wu et al., 2022). It is dominated by a daily cycle with a flat zero every night, and is the most strongly periodic of the benchmarks.

Electricity and Traffic: Hourly electricity consumption of 321 clients from 2012 to 2014 (26304 time steps), and hourly occupancy rates of 862 road sensors on San Francisco Bay Area freeways from 2015 to 2016 (17544 time steps) (Wu et al., 2022). Together they form the high-dimensional end of the evaluation, where the input dimension is comparable to or larger than the reservoir size.

NARMA10: The tenth-order non-linear autoregressive moving average task, the standard test of whether a reservoir combines memory with non-linearity. An input $u_t \sim \mathcal{U}(0, 0.5)$ drives

$$y_{t+1} = 0.3\, y_t + 0.05\, y_t \sum_{i=0}^{9} y_{t-i} + 1.5\, u_{t-10}\, u_t + 0.1, \tag{48}$$

and the model, having read $u_1, \dots, u_t$, outputs y_t at every step, so the target depends on inputs a dozen steps back, partly through a product of two of them. The training sequence has 13000 time steps, the first 1000 of which are discarded as washout, and the test time series is an independent run of 1000 time steps. Input and target are not normalized.

CHAOTIC SYSTEMS

Five systems from the dysts benchmark (Gilpin, 2021) of chaotic dynamical systems, each integrated with an explicit Runge–Kutta solver and resampled to 100 time steps per dominant period. A trajectory of 20000 time steps is standardized with the statistics of its first 10000 time steps, which form the training set, and its remaining 10000 time steps are the test set. The task is one-step prediction of the full state, the domain in which ESNs have been shown to lie near the efficiency-performance Pareto front (Gilpin, 2023). The five datasets are ordered by their maximum Lyapunov exponent λ_l, the rate at which two nearby trajectories separate and so the rate at which a forecast's error is amplified.

Torus ($\lambda_l = 0.01$)**:** A quasiperiodic trajectory on a torus,

$$\begin{aligned} \dot{x} &= -an\sin(nt)\cos t - (r + a\cos(nt))\sin t, \\ \dot{y} &= -an\sin(nt)\sin t + (r + a\cos(nt))\cos t, \\ \dot{z} &= an\cos(nt), \end{aligned} \tag{49}$$

with $a = 0.5$, $n = 15.3$ and $r = 1$. Its Lyapunov spectrum is identically zero, so it is not chaotic at all: the motion is a superposition of two frequencies, and it is the case an oscillating response function should resolve exactly. It is the easy anchor of the five.

Rossler ($\lambda_l = 0.15$)**:** The Rossler attractor,

$$\dot{x} = -y - z, \qquad \dot{y} = x + ay, \qquad \dot{z} = b + z(x - c), \tag{50}$$

with $a = b = 0.2$ and $c = 5.7$. A single folded band with one weakly positive exponent, it is the mildly chaotic textbook flow.

Lorenz ($\lambda_l = 0.89$)**:** The Lorenz system,

$$\dot{x} = \sigma(y - x), \qquad \dot{y} = x(\rho - z) - y, \qquad \dot{z} = xy - \beta z, \tag{51}$$

with $\sigma = 10$, $\rho = 28$ and $\beta = 2.667$. Dissipative chaos on the two-lobed attractor, with roughly six times the exponent of Rossler.

Lorenz96 ($\lambda_l = 1.34$)**:** The spatially extended weather toy model,

$$\dot{x}_i = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F, \qquad i = 1, \ldots, d, \tag{52}$$

with cyclic indices, $d = 4$ and forcing $F = 20$. Unlike the preceding flows, its channels are coupled cells of one spatial lattice rather than the coordinates of a single point, so the readout must track an interaction between channels rather than a trajectory in a fixed frame.

DoublePendulum ($\lambda_l = 3.99$)**:** The planar double pendulum in Hamiltonian form, with state $(\theta_1, \theta_2, p_1, p_2)$ of two angles and their conjugate momenta, unit masses and lengths $m_1 = m_2 = l_1 = l_2 = 1$ and $g = 9.81$. It is conservative rather than dissipative: there is no attractor, so no part of an error the model makes is damped by the dynamics. It is the hardest of the five considered chaotic dynamical systems.

Width	Dataset	Linear ESN MSE	ParalESN MSE	TUNE MSE
$k = 100$	Weather	0.296 ± 0.007	$\underline{0.175} \pm 0.004$	$\mathbf{0.161} \pm 0.001$
	Exchange	0.103 ± 0.007	$\underline{0.103} \pm 0.003$	$\mathbf{0.102} \pm 0.006$
	Solar	0.054 ± 0.001	$\mathbf{0.054} \pm 0.000$	$\underline{0.054} \pm 0.001$
	Electricity	0.418 ± 0.031	$\mathbf{0.347} \pm 0.005$	$\underline{0.347} \pm 0.013$
	Trafic	$\underline{0.594} \pm 0.010$	0.597 ± 0.018	$\mathbf{0.579} \pm 0.016$
	NARMA10	$\underline{0.001} \pm 0.000$	0.001 ± 0.000	$\mathbf{0.001} \pm 0.000$
	Torus	$(\underline{4.90} \pm 1.11) \times 10^{-7}$	$(7.85 \pm 1.80) \times 10^{-7}$	$(\mathbf{4.60} \pm 1.13) \times 10^{-7}$
	Rossler	$(1.04 \pm 0.30) \times 10^{-6}$	$(\mathbf{4.76} \pm 2.20) \times 10^{-7}$	$(\underline{8.52} \pm 2.30) \times 10^{-7}$
	Lorenz	$(\underline{5.62} \pm 1.78) \times 10^{-7}$	$(6.72 \pm 2.81) \times 10^{-7}$	$(\mathbf{5.56} \pm 4.42) \times 10^{-7}$
	Lorenz96	$(8.15 \pm 1.14) \times 10^{-6}$	$(\mathbf{2.70} \pm 0.16) \times 10^{-6}$	$(\underline{6.80} \pm 1.40) \times 10^{-6}$
	DoublePendulum	$(1.87 \pm 0.10) \times 10^{-4}$	$(\mathbf{1.15} \pm 0.18) \times 10^{-4}$	$(\underline{1.79} \pm 0.12) \times 10^{-4}$

Table 10: Forecasting performance (test MSE) on Weather and Exchange, forecast $H = 96$ steps ahead, on the high-dimensional Solar, Electricity and Trafic, forecast $H = 8$ steps ahead, and on NARMA10 and five chaotic systems from the dysts benchmark, one step ahead, at a small ($k = 100$) reservoir size; every model uses $T = 0.01 T_{\text{seq}}$. **Bold** indicates the best result, underline the second best.

F.3 MEMORY CAPACITY AND ABLATION

Memory capacity (MC) is an important metric for analyzing the short-term memorization capabilities of ESNs. In Figure 5 we provide an ablation plot for various $\mathbf{\Omega}$–$\mathbf{\Theta}$ interactions. The most important observation is the stationary regime for high ω values: TUNE achieves a perfect capacity $\mathrm{MC} = 100$, which is equivalent in performance to linear ESNs with an orthogonal transfer matrix. This result supports our claims of equivalence between stationary GP response functions and linear ESNs with orthogonal transfer matrices. When decay is active, a peculiar effect comes into play

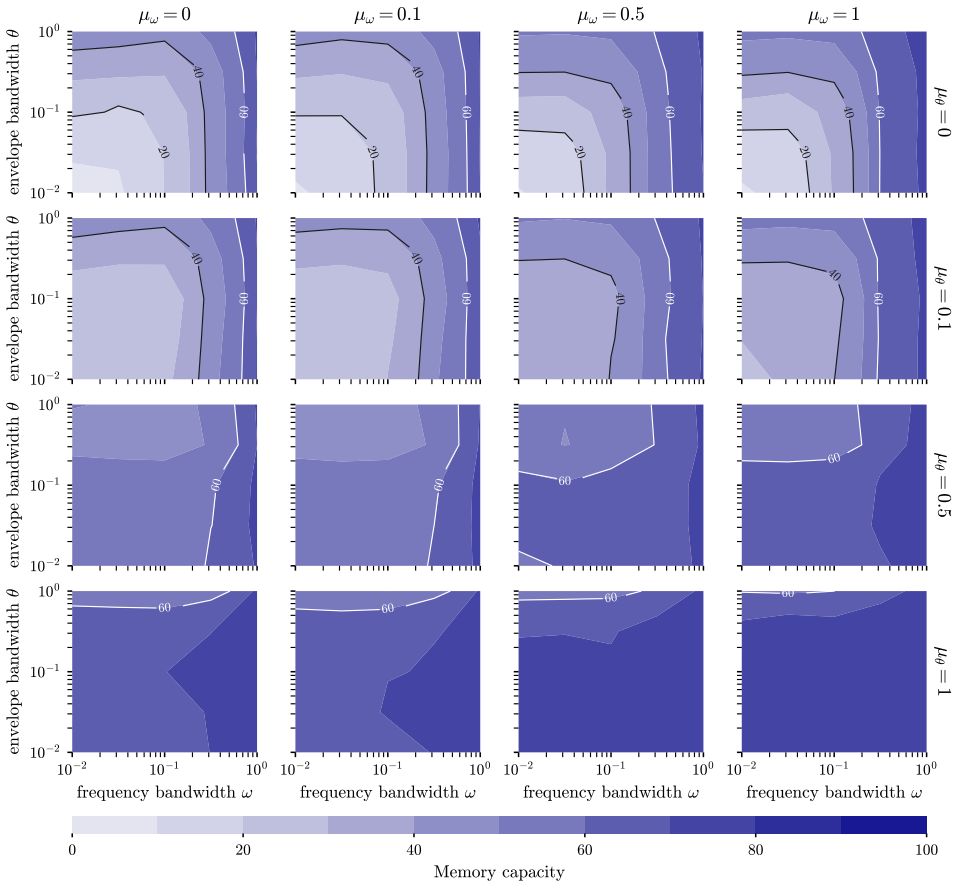


Figure 5: Memory capacity, MC, of the TUNE response function at width $k = 100$ over its frequency bandwidth ω and envelope bandwidth θ, for frequency means μ_ω (columns) and envelope means μ_θ (rows), with no decay ($\lambda_{min} = \lambda_{max} = 1$), $m = k$ random features and a linear readout; averaged over 5 trials. MC is summed over the delays $1, \ldots, 200$ on i.i.d. noise, and the colour scale ends at k, the most a width-k reservoir can hold. Darker is higher MC; each contour line is labeled with the capacity it traces.

In the same and the opposite regime, we show the memory curve for varying values of θ, ω and k, with zero means, in Figure 6. Increasing θ is not equivalent to raising ω: because of the singular combined peaks of the non-stationary envelope, memorization becomes unstable, since it is unlikely that the whole region is equally covered for $k \leq T$; this large variance reduces when $k > T$.

for the stationary mechanism, visible in Figure 7: for larger bandwidths the total memory capacity is always lower, but the memory curve reaches further lag values, likely because the process mimics a low-pass filter. Increasing k saturates this effect and removes it completely. Such behavior is not present for the non-stationary mechanism, which is mostly width-invariant, except for larger bandwidths, which can now saturate the autocorrelation.

Table 11 reports the complete ablation of the four terms of the TUNE response function on `sMNIST` and `ETTh1`, and Figure 8 reports the effect of the number of Monte-Carlo features m.

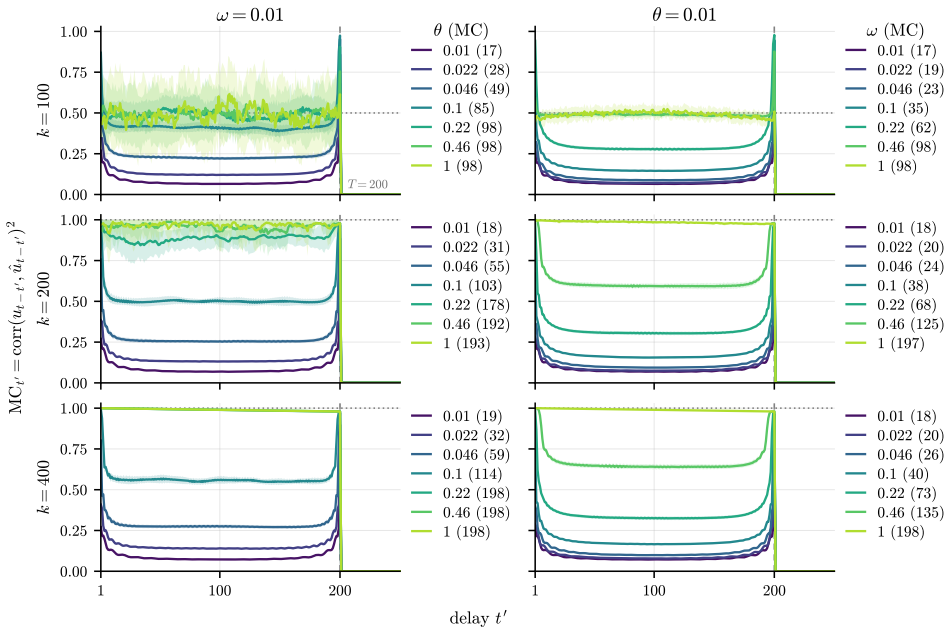


Figure 6: MC curve of the TUNE response function at width $k \in \{100, 200, 400\}$, window $T = 200$ and $\lambda_{min} = \lambda_{max} = 1$, averaged over 30 trials: the squared correlation MC_k between the recalled and the true input k steps back, on a sequence of i.i.d. noise. MC is the sum of a curve over the delays.

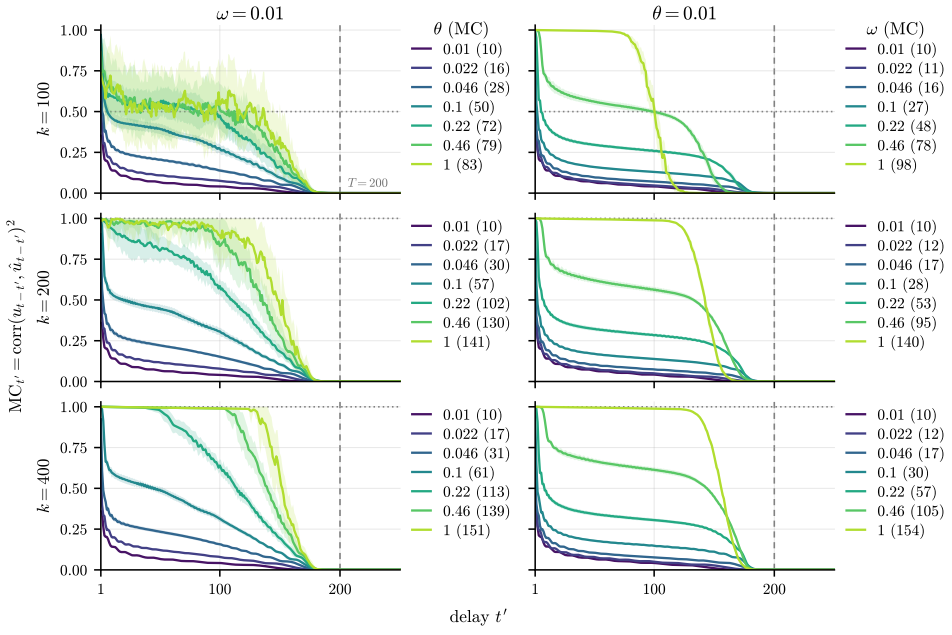


Figure 7: MC curve of the TUNE response function at width $k \in \{100, 200, 400\}$, window $T = 200$ and $\lambda_{min} = \lambda_{max} = 0.9$, averaged over 30 trials: the squared correlation MC_k between the recalled and the true input t' steps back, on a sequence of i.i.d. noise. The memory capacity is the sum of a curve over the delays.

Ablation	sMNIST Test accuracy (%) ↑		ETTh1 Test MSE ↓		
	$k = 200$	$k = 2000$	$H = 1$	$H = 96$	$H = 192$
full	90.4 ± 0.3	$\underline{96.9} \pm 0.1$	$\mathbf{0.197} \pm 0.001$	0.611 ± 0.003	0.781 ± 0.005
no Ω	89.2 ± 0.2	82.1 ± 16.8	0.199 ± 0.004	0.611 ± 0.001	$\mathbf{0.767} \pm 0.003$
no Θ	90.1 ± 0.4	96.6 ± 0.0	$\underline{0.197} \pm 0.001$	0.619 ± 0.002	0.768 ± 0.001
no Λ	$\mathbf{90.8} \pm 0.2$	$\mathbf{97.0} \pm 0.1$	0.257 ± 0.010	0.612 ± 0.002	0.773 ± 0.003
no m	89.4 ± 0.5	95.2 ± 0.1	0.202 ± 0.004	$\mathbf{0.610} \pm 0.002$	0.770 ± 0.004
no Ω+Θ	65.3 ± 0.6	53.5 ± 10.8	0.231 ± 0.002	0.733 ± 0.009	0.886 ± 0.003
no Ω+Λ	89.5 ± 0.3	87.4 ± 4.4	0.262 ± 0.007	0.612 ± 0.002	0.773 ± 0.002
no Ω+m	88.7 ± 0.2	91.0 ± 0.8	0.212 ± 0.002	0.614 ± 0.002	0.771 ± 0.002
no Θ+Λ	$\underline{90.7} \pm 0.2$	96.7 ± 0.1	0.245 ± 0.003	0.611 ± 0.003	0.770 ± 0.001
no Θ+m	87.1 ± 0.7	87.4 ± 8.5	0.203 ± 0.002	0.619 ± 0.010	$\underline{0.767} \pm 0.002$
no Λ+m	89.8 ± 0.6	90.9 ± 6.4	0.248 ± 0.003	0.611 ± 0.001	0.785 ± 0.009
no Ω+Θ+Λ	22.8 ± 0.8	23.2 ± 0.1	1.09 ± 0.00	1.18 ± 0.00	1.33 ± 0.00
no Ω+Θ+m	58.9 ± 0.4	57.1 ± 1.4	0.227 ± 0.001	0.740 ± 0.013	0.890 ± 0.009
no Ω+Λ+m	89.6 ± 0.5	89.5 ± 1.7	0.264 ± 0.010	0.613 ± 0.002	0.772 ± 0.002
no Θ+Λ+m	87.5 ± 0.5	91.9 ± 0.0	0.251 ± 0.005	$\underline{0.610} \pm 0.002$	0.779 ± 0.003
no Ω+Θ+Λ+m	22.0 ± 0.5	23.0 ± 0.3	1.09 ± 0.00	1.18 ± 0.00	1.33 ± 0.00

Table 11: Complete ablation of the TUNE response function on sMNIST classification (test accuracy, width $k \in \{200, 2000\}$) and ETTh1 forecasting (test MSE, width $k = 100$ and $T = 0.01 T_{\text{seq}}$, across horizons H). Each row removes a subset of the four terms: the oscillation ($\Omega = 0$), the envelope ($\Theta = 0$), the decay ($\Lambda = 1$) and the sum over Monte-Carlo features ($m = 1$), and searches every hyperparameter it does not pin under the same budget as the full model. All 2^4 subsets are run, so the table shows interactions and not only one-at-a-time knockouts. **Bold** indicates the best result in each column, underline the second best.

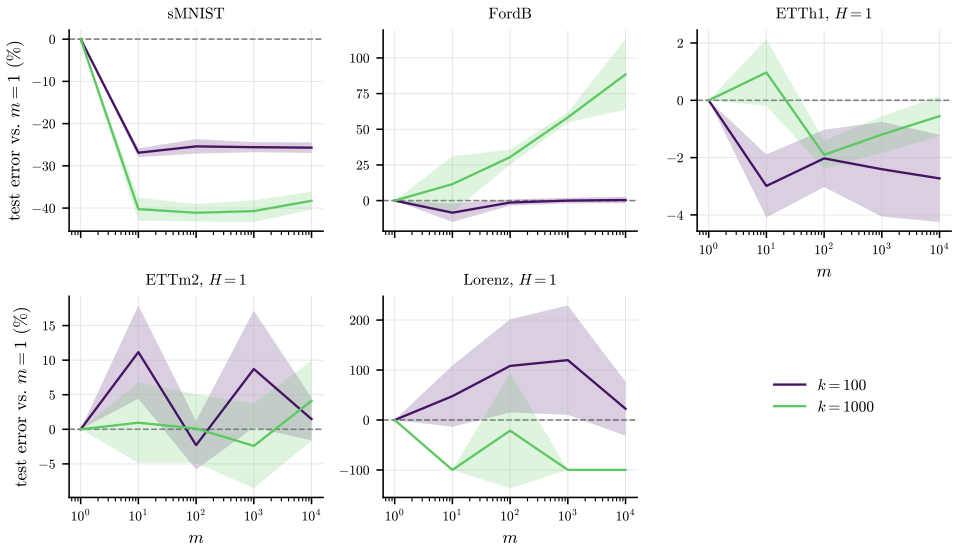


Figure 8: Relative change of the test error (MSE) of a stationary TUNE against $m = 1$, over the number of random features m at widths $k \in \{100, 1000\}$. Shaded is one standard deviation over the 5 re-initializations of each winning configuration, propagated through the ratio. The optimal m is clearly dataset- and k-dependent.

F.4 Optimal SNR bandwidth in relation to optimal accuracy bandwidth

Theorem 3.4 predicts the frequency bandwidth $\omega^\star$ that maximizes the signal-to-noise ratio (SNR) of a single hidden state component at a given mean μ. Whether this is also the bandwidth that maximizes accuracy is a separate question, which we test on `sMNIST`, `psMNIST`, `Blink`, `FordA`, `FordB`, `StarLightCurves` and `FaultDetectionA` (all classification benchmarks in this study). It is important to note that these datasets have their own intrinsic noise, which was not separated, so the reported results are noisy, yet still informative. Each benchmark is standardized by the mean and standard deviation of its training series but otherwise used as provided, and S_x is the mean periodogram of those series. The series are corrupted by a stationary GP of the response function's own family, with frequencies drawn from $\mathcal{N}(\mu_n, \omega_n^2)$ and a white floor carrying 10% of its power, at three times the root-mean-square of the clean signal, so that S_n is known exactly. Every combination of $\mu_n \in \{0.1, 0.3, 1\}\pi$ and $\omega_n \in \{0.1, 0.3\}\pi$ is paired with every mean $\mu \in \{0.01, 0.05, 0.3\}\pi$ of the response function, which gives 126 runs. For each run, $\omega^\star$ maximizes $\mathrm{SNR}(\omega)$ by quadrature before any model is trained; a prediction at an end of the range is the branch of the theorem without an interior optimum.

We disable every TUNE's term outside the theorem by letting $\Theta = 0$, $\mu_\theta = 0$ and $\lambda_{min} = \lambda_{max} = 1$, and use $k = 100$, $m = 1000$, at most 6000 training and 2000 test time series, and a readout on $\boldsymbol{h}_T$ alone. We sweep ω over 30 log-spaced values in $[0.001, 1]$ at 30 trials each, and locate the accuracy peak $\hat{\omega}^\star$ by fitting a parabola in $\log\omega$ to its best points, with one bootstrap standard deviation over trials. Figures 9 and 10 and Table 12 report the sweep; 99 of the 126 predictions are interior, and 68 runs have an interior accuracy peak. Where the theorem predicts an interior optimum, the measured peak is centered on it, with a median $\hat{\omega}^\star/\omega^\star$ of 1.06 and 22 of 64 predictions within one standard deviation of it; where the prediction lies at an end of the range, only 4 runs have a peak at all and it lies a median factor of 10.9 away, none of them within. The agreement differs strongly between benchmarks. On `StarLightCurves` the measured peak tracks the prediction with a slope of 1.04 in $\log\omega$ and 10 of its 11 peaks within one standard deviation, on `FaultDetectionA` with a slope of 0.97, a rank correlation of 0.96 and 5 of 15, and on `Blink` with a slope of 0.82 and 6 of 11; their median distances are 0.3, 1.2 and 0.9 standard deviations. On `sMNIST` and `psMNIST` no prediction falls within one standard deviation and the peak lies a median factor of 8.8 and 14 from it, and on `FordA` and `FordB` the accuracy stays near chance (a median peak accuracy of 0.51 over runs), so that only 5 and 3 of their 18 runs have a peak to compare at all. Table 13 gives what the prediction costs in accuracy rather than in bandwidth: tuning to $\omega^\star$ instead of to $\hat{\omega}^\star$ gives up at most 1.7% on `Blink`, `FordA`, `FordB`, `StarLightCurves` and `FaultDetectionA`, and up to 31.0% and 19.7% on `sMNIST` and `psMNIST`. A few entries are slightly negative, where the predicted bandwidth edges out the fitted peak within the noise of the curves. The SNR of a single component therefore sets the accuracy-optimal bandwidth on some benchmarks but not on all; we do not isolate the cause here, and as the noise is stationary, Conjecture 3.5 remains untested. In general, we find that SNR-optimized frequency distributions give good estimates of the accuracy-optimized frequency distributions on several of the datasets, even with already-present unknown noise, and miss them on the rest. It is important to note that one, in general, does not have access to the underlying noise spectral distribution, so Theorem A.3 cannot easily provide a stepping stone to zero-shot hyperparameter optimization.

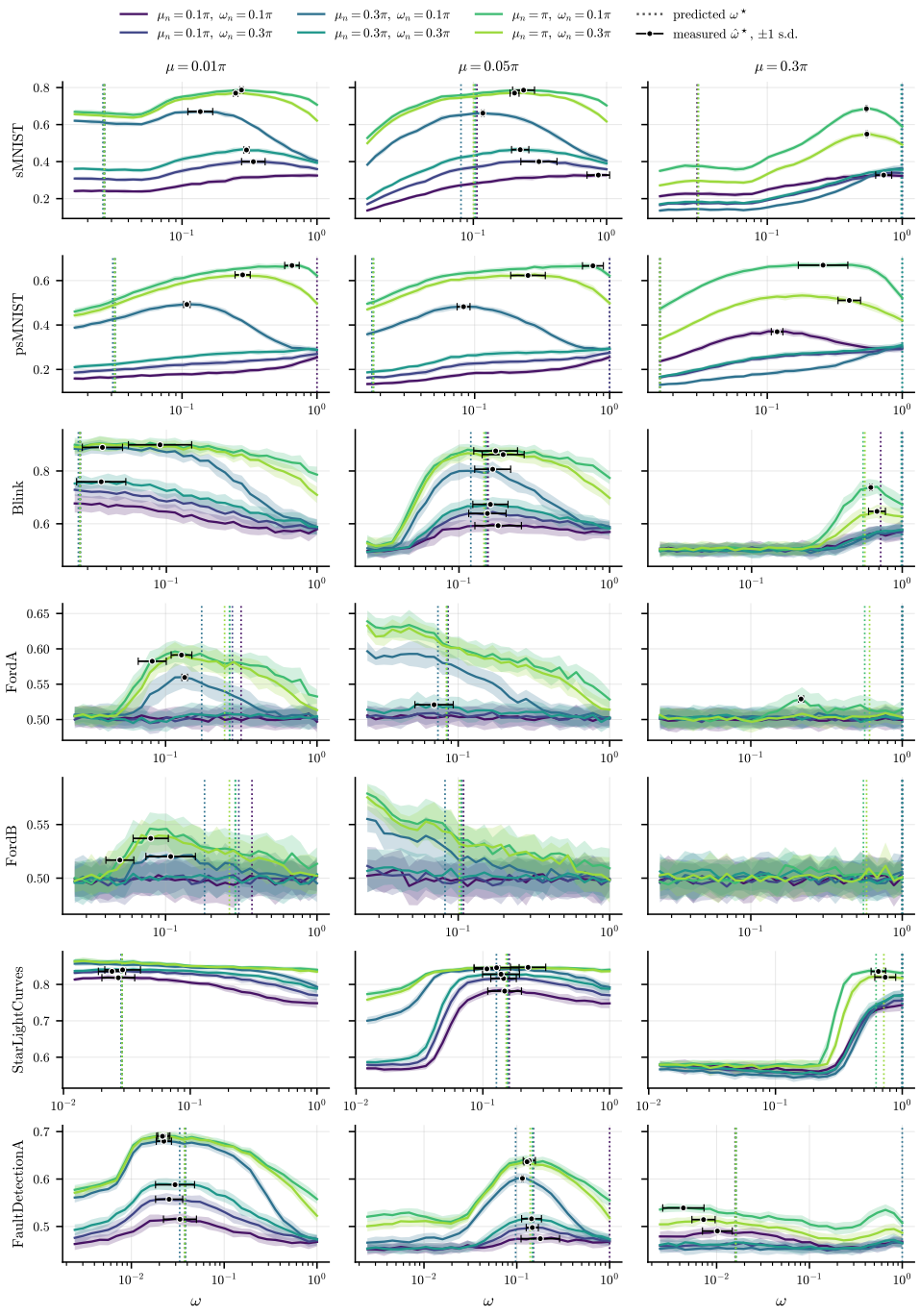


Figure 9: Test accuracy averaged over 30 trials per benchmark (rows), mean μ of the response function (columns) and noise (colour), shaded by one standard deviation over those trials, with the bandwidth predicted by Theorem A.3 dotted in the matching colour and the measured peak $\hat{\omega}^\star$ marked in black on each curve, with one bootstrap standard deviation of its location over trials.

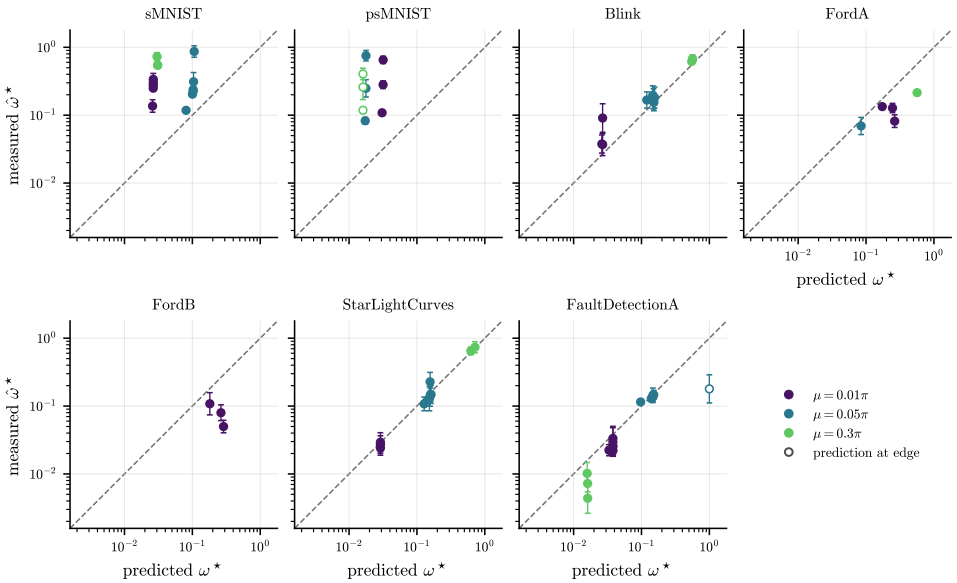


Figure 10: Measured against predicted bandwidth, one panel per benchmark, with bars of one bootstrap standard deviation over 30 trials. Colour is the mean μ of the response function, hollow markers are predictions at an end of the resolvable range, and the dashed line is the identity.

Dataset	μ	$\mu_n = 0.1\pi$		$\mu_n = 0.3\pi$		$\mu_n = \pi$	
		0.1π	0.3π	0.1π	0.3π	0.1π	0.3π
sMNIST	0.01π	–	13	7.8	55	82	59
	0.05π	11	3.5	17	5.2	4.4	9.1
	0.3π	23	–†	–†	–	320	235
psMNIST	0.01π	–†	–	23	–	24	17
	0.05π	–†	–†	14	–	21	8.9
	0.3π	20†	–†	–†	–†	6.6†	17†
Blink	0.01π	–	–	1.2	0.9	2.5	–
	0.05π	0.4	0.0	1.2	0.3	0.5	0.9
	0.3π	–	–†	–†	–	6.5	1.4
FordA	0.01π	–	–	6.2	–	5.5	4.2
	0.05π	–	–	–	0.7	–	–
	0.3π	–†	–†	–†	–†	25	–
FordB	0.01π	–	–	1.4	–	8.3	4.5
	0.05π	–	–	–	–	–	–
	0.3π	–†	–†	–†	–†	–	–
StarLightCurves	0.01π	0.2	0.7	–	0.0	–	–
	0.05π	0.3	0.4	0.7	0.3	1.2	0.4
	0.3π	–†	–†	–†	–†	0.3	0.1
FaultDetectionA	0.01π	0.3	1.2	2.1	0.5	3.5	3.8
	0.05π	3.6†	0.2	8.9	0.1	0.5	1.0
	0.3π	1.2	–	–†	–	2.5	2.8

Table 12: Distance between the measured accuracy peak $\hat{\omega}^\star$ and the bandwidth $\omega^\star$ predicted by Theorem A.3, in bootstrap standard deviations of $\log \hat{\omega}^\star$, per benchmark, response-function mean μ and noise with frequency mean μ_n and bandwidth ω_n (second header row). † marks a prediction at an end of the resolvable range rather than an interior optimum; a dash marks a run without an interior accuracy peak or with a zero-width band.

Dataset	μ	$\mu_n = 0.1\pi$		$\mu_n = 0.3\pi$		$\mu_n = \pi$	
		0.1π	0.3π	0.1π	0.3π	0.1π	0.3π
sMNIST	0.01π	–	9.4	5.8	10.7	12.5	12.2
	0.05π	4.4	2.9	1.0	3.0	1.9	1.9
	0.3π	10.1	–†	–†	–	31.0	25.2
psMNIST	0.01π	–†	–	6.7	–	15.6	13.5
	0.05π	–†	–†	11.8	–	16.2	14.3
	0.3π	$13.3^\dagger$	–†	–†	–†	$19.7^\dagger$	$17.6^\dagger$
Blink	0.01π	–	–	0.6	0.3	0.4	–
	0.05π	-0.1	0.1	0.7	0.3	0.4	0.7
	0.3π	–	–†	–†	–	0.6	1.3
FordA	0.01π	–	–	0.8	–	0.2	1.7
	0.05π	–	–	–	0.3	–	–
	0.3π	–†	–†	–†	–†	1.2	–
FordB	0.01π	–	–	0.7	–	-0.9	1.3
	0.05π	–	–	–	–	–	–
	0.3π	–†	–†	–†	–†	–	–
StarLightCurves	0.01π	-0.0	-0.1	–	-0.0	–	–
	0.05π	0.4	0.3	0.2	0.1	0.2	0.0
	0.3π	–†	–†	–†	–†	-0.0	-0.0
FaultDetectionA	0.01π	0.2	0.2	0.3	-0.1	0.5	0.7
	0.05π	$0.6^\dagger$	-0.0	0.2	0.0	0.3	0.2
	0.3π	0.5	–	–†	–	1.1	0.6

Table 13: Test accuracy given up by tuning the bandwidth to the prediction $\omega^\star$ of Theorem A.3 instead of to the measured peak $\hat{\omega}^\star$, in points of accuracy averaged over 30 trials and paired within a seed, per benchmark, response-function mean μ and noise with frequency mean μ_n and bandwidth ω_n (second header row). † marks a prediction at an end of the resolvable range rather than an interior optimum; a dash marks a run without an interior accuracy peak.