LogSig-SSM: Time-Series Modelling with Multi-Scale Log-Signature Compression for State-Space Models

Felix Oury
Department of Computing
Imperial College London
felix.oury19@imperial.ac.uk

Nicolas Calvo Peiro
Department of Bioengineering
Imperial College London
nicolas.calvo-peiro19@imperial.ac.uk

Reiko J. Tanaka
Department of Bioengineering
Imperial College London
r.tanaka@imperial.ac.uk

Abstract

Time-series data are often sampled irregularly at high frequencies and exhibit long-range dependencies, which makes long-horizon modelling difficult. Continuous-time models such as neural controlled differential equations (NCDEs) and neural rough differential equations (NRDEs) can handle irregular sampling, but they scale poorly to long sequences. Selective state-space models (SSMs) such as Mamba scale linearly with sequence length, but they provide limited recurrent mixing across hidden dimensions within a single block. We propose LogSig-SSM (Log-Signature Compression for State-Space Models), which first compresses long multivariate time series into a shorter sequence of tokens using multi-scale windowed log-signatures, and then processes these tokens with a selective SSM backbone. LogSig-SSM is scalable and robust to irregular sampling, combining log-signature tokens that capture higher-order cross-channel interactions with a selective SSM that models long-range dependencies. The model also admits a continuous-time interpretation as an NCDE/NRDE-style system driven by a log-signature-based input, in which selectivity induces an input-dependent rescaling of the latent dynamics. Across four benchmarks, namely long-sequence classification on UEA, high-frequency physiological regression on PPG-DaLiA, multivariate weather forecasting, and irregularly sampled clinical prediction on PhysioNet Sepsis, LogSig-SSM outperforms or matches strong SSM and continuous-time baselines while training up to $30\times$ faster and using up to $37\times$ less GPU memory than Mamba on the longest sequences.[1]

1 Introduction

Time-series modelling has numerous real-world applications in domains such as healthcare [9] and finance [16]. These applications typically involve signals, such as ECG or transaction prices, that are irregularly sampled at high frequencies and contain information spanning long time scales. Processing these long sequences creates memory and computational bottlenecks which motivate the development of models that scale to long sequences while remaining reliable under irregular sampling rates. Two widely-used approaches are continuous-time models and State-Space-Models (SSMs).

Continuous-time models such as Neural Ordinary Differential Equations (NODEs) [5] and Neural Controlled Differential Equations (NCDEs) [19] treat observed time-series data as discrete-time

[1] Code: https://github.com/felix-oury/LogSigSSM.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

samples from an underlying continuous-time process. NCDEs operate on continuous interpolations of the data and naturally accommodate irregularly-sampled observations. However, training and inference can become costly on longer sequences because numerical solvers often require more integration steps. To address this computational bottleneck, Neural Rough Differential Equations (NRDEs) [25] and Log-NCDEs [38] employ the log-ODE method [6, 11] which summarises information between time intervals with truncated log-signatures, enabling the solver to take fewer, larger steps. Signature-based summarisation has also been adapted to improve computational efficiency and handle irregular sampling in discrete models such as RNNs [20] and transformers [24].

In parallel, structured SSMs [4, 8, 14, 15, 31, 34] have emerged as an efficient alternative to attention-based models. Unlike transformers, whose computational cost scales quadratically [36], modern SSMs achieve linear-time processing, constant-memory inference, and efficient GPU-parallel training. SSMs have also shown competitive performance across various sequence modelling tasks [21, 32, 33, 43]. Recent selective SSMs, such as Mamba [14], introduced input-dependent gating enabling the model to filter context content rather than uniformly compressing it. Additionally, novel theoretical work [7] has established that SSMs admit a continuous-time interpretation as linear controlled differential equations (CDEs), connecting them to the continuous-time modelling framework.

However, recent empirical studies highlight two key limitations of Mamba. First, performance can degrade on very long sequences, where Mamba-based models may underperform attention-based baselines [37]. Existing approaches mitigate this limitation through token pruning or decimation [40, 3]. Second, Mamba's diagonal state transition admits no within-block mixing across hidden dimensions. As Cirone et al. [7, Cor 4.4] show, the closure of a single Mamba block decomposes as a sum of per-channel functionals. This means genuinely multivariate path functionals, such as Lévy areas, which depend on the joint behaviour of two channels, lie outside this closure. Two mitigations exist for this limitation, but come with costs. Stacking blocks with linear inter-block mixing eventually recovers full expressivity (Prop. 4.5 in [7]), but only as both depth and hidden-state size grow without bound, i.e. cross-channel structure is paid for in model size. Replacing the diagonal state transition with richer structure [39, 35] avoids the depth cost, but at non-trivial efficiency overhead (PD-SSM reports approximately $6\times$ slower runtime than Mamba [35]). These limitations motivate a complementary strategy: compressing the effective sequence length while injecting explicit higher-order cross-channel information at the input, leaving the efficient diagonal scan intact.

Log-signature features [23] provide exactly such input-level functionals. They encode iterated integrals of the path, including Lévy areas, while inheriting invariance to monotone time reparameterisations and thus offering robustness to irregular sampling. Building on this idea, we introduce LogSig-SSM, a two-stage architecture that achieves both accuracy and efficiency on long sequences. First, a long multivariate sequence is mapped to a shorter token sequence by computing truncated log-signatures over strided windows at multiple scales, yielding compact descriptors that explicitly capture cross-channel dependencies. Second, these tokens are processed by a diagonal selective SSM, which efficiently models long-range dependencies over the reduced sequence length.

Our contributions are:

- We introduce LogSig-SSM, a model that integrates log-signature tokenisation with selective SSMs. The tokeniser compresses long, irregularly-sampled multivariate sequences into compact summaries by computing truncated log-signatures over multi-scale windows.
- We provide a continuous-time interpretation of LogSig-SSM by extending the linear-CDE view of SSMs [7], interpreting the model as an NCDE/NRDE-style system driven by log-signature-derived controls with input-dependent time rescaling of the latent dynamics.
- We establish a depth–truncation tradeoff. A single LogSig-SSM block followed by an MLP can approximate any continuous functional of the windowed log-signature up to depth m, while the same class of functionals lies outside the closure of a single diagonal Mamba block (even with a pointwise MLP, which cannot recover features absent from the block state) and requires chained blocks to recover.
- We evaluate LogSig-SSM on four benchmarks: long-sequence time-series classification (UEA), high-frequency physiological signal regression (PPG-DaLiA), multivariate weather forecasting, and irregularly sampled early sepsis prediction (PhysioNet Sepsis). LogSig-SSM matches or exceeds strong SSM and continuous-time baselines and reduces training time by $30\times$ and GPU memory by $37\times$ relative to standard Mamba on the longest sequences.

2 Theoretical Background

We review the theoretical foundations of LogSig-SSM: (i) structured SSMs, which serve as the sequential processing backbone, (ii) their continuous-time interpretation as linear CDEs, and (iii) log-signatures, which provide compact summaries of multivariate time-series over time intervals.

2.1 Structured State-Space Models

Modern SSM sequence layers can be viewed as discretisations of a continuous-time linear state-space system. Let $X : [0,1] \to \mathbb{R}^d$ be an input signal observed at time points $0 \le t_0 < \cdots < t_L \le 1$. We write $x_\ell := X(t_\ell)$ for the resulting discrete sequence and $x_{i,\ell} := X_i(t_\ell)$ for its ith channel. A channel-wise linear SSM processes each channel X_i with its own latent state $H_i : [0,1] \to \mathbb{K}^N$ and produces a scalar output $y_i : [0,1] \to \mathbb{R}$:

$$\frac{dH_i(t)}{dt} = A_i H_i(t) + B_i\, X_i(t), \quad y_i(t) = C_i H_i(t), \tag{1}$$

where $A_i \in \mathbb{K}^{N\times N}$ is structured (often diagonal), $B_i \in \mathbb{K}^N$, $C_i \in \mathbb{K}^{1\times N}$, and $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. To apply SSMs to discrete sequence data, we discretise the continuous-time dynamics using a step size $\Delta_i > 0$. Under a zero-order hold (ZOH) discretisation with step Δ_i:

$$H_{i,\ell} = \overline{A}_i H_{i,\ell-1} + \overline{B}_i x_{i,\ell}, \quad \overline{A}_i = e^{\Delta_i A_i}, \quad \overline{B}_i = (\Delta_i A_i)^{-1}(e^{\Delta_i A_i} - I)\Delta_i B_i. \tag{2}$$

A key design choice in modern SSMs is how the continuous-dynamics are discretised. Linear Time-Invariant (LTI) SSMs, such as S4 [15] and S5 [34], use a fixed step size Δ_i learned at training time, giving time-invariant operators $(\overline{A}_i, \overline{B}_i)$. Selective SSMs, such as Mamba [14], instead make the step size input-dependent. This induces input-dependent selectivity, meaning the model can adapt its update rate across the sequence, focusing on important information while down-weighting irrelevant inputs. In addition, the input and output projections $s_B : \mathbb{R}^d \to \mathbb{R}^N$ and $s_C : \mathbb{R}^d \to \mathbb{R}^{1\times N}$ are learned functions of the current sample x_ℓ. In Mamba, the channel-wise internal step is defined as

$$\Delta_{i,\ell} = \mathrm{softplus}(\alpha_i^\top x_\ell + \beta_i), \qquad \alpha_i \in \mathbb{R}^d,\ \beta_i \in \mathbb{R}, \tag{3}$$

yielding input-dependent operators,

$$\overline{A}_{i,\ell} = \exp(\Delta_{i,\ell} A_i), \quad \overline{B}_{i,\ell} = (\Delta_{i,\ell} A_i)^{-1}\big(\exp(\Delta_{i,\ell} A_i) - I\big)\,\Delta_{i,\ell}\, s_B(x_\ell), \quad \overline{C}_{i,\ell} = s_C(x_\ell). \tag{4}$$

For small $\Delta_{i,\ell}$, $\overline{B}_{i,\ell} \approx \Delta_{i,\ell} s_B(x_\ell)$, so selectivity acts as input-dependent scaling of the state update.

2.2 Unifying View: Linear CDEs and SSMs

SSM layers admit a continuous-time interpretation as linear CDEs [7], a connection we extend in Section 3.2.1 to derive LogSig-SSM's continuous-time form. Let $H_t \in \mathbb{R}^N$ denote the hidden state and let $\omega : [0,1] \to \mathbb{R}^{d_\omega}$ and $\xi : [0,1] \to \mathbb{R}^{d_\xi}$ be control paths. We consider the linear CDE

$$dH_t \;=\; \sum_{i=1}^{d_\omega} A_i H_t\, d\omega_t^i \;+\; W_B\, d\xi_t, \qquad H_0 \in \mathbb{R}^N, \tag{5}$$

for learnable matrices $A_i \in \mathbb{R}^{N\times N}$ and $W_B \in \mathbb{R}^{N\times d_\xi}$[2]. Selective SSMs correspond to input-dependent driving paths (ω^X, ξ^X). Viewed in this continuous-time framework, selectivity can be interpreted as an input-dependent adaptive time rescaling: the model dynamically adjusts how quickly its latent state evolves based on the characteristics of the input [7].

2.3 Path Signatures and Log-Signatures

The signature transform represents a path through its hierarchy of iterated integrals [23, 6]. Formally, let $X \in C^1([0,1]; \mathbb{R}^d)$ be a path and fix $0 \le a < b \le 1$. The path signature of X over $[a,b]$ is the (infinite) collection of iterated integrals:

$$\mathrm{Sig}(X)_{a,b} := \Big(1, \mathbf{X}^1_{a,b}, \mathbf{X}^2_{a,b}, \dots\Big), \quad \mathbf{X}^m_{a,b} := \int_{a<u_1<\cdots<u_m<b} dX_{u_1} \otimes \cdots \otimes dX_{u_m}, \tag{6}$$

[2]To avoid overloading notation, we use W_B for the time-invariant injection matrix in the linear CDE view (denoted B in [7]). We reserve $s_B(\cdot)$ for Mamba's input-dependent projection in selective scans, written as B in [14].

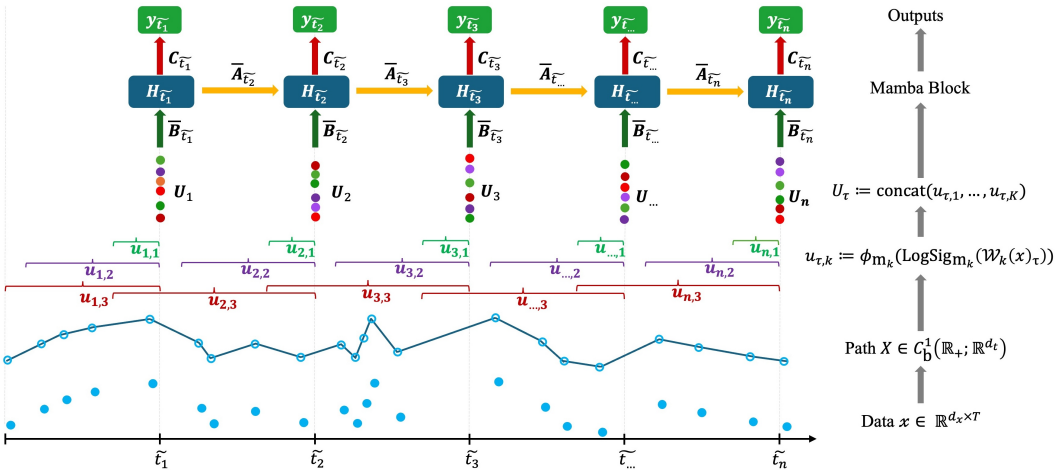


Figure 1: Discrete-time LogSig-SSM pipeline. A multivariate time series $x \in \mathbb{R}^{d_x \times T}$ is summarised at strided anchors $\tilde{t}_\tau := t_{\tau s}$. For each scale k, a windowed segment $\mathcal{W}_k(x)_\tau$ is extracted, and its truncated log-signature coordinate vector, $u_{\tau,k} := \phi_{m_k}\Big(\mathrm{LogSig}_{m_k}\big(\mathcal{W}_k(x)_\tau\big)\Big) \in \mathbb{R}^{D_k}$, is computed. These are concatenated across scales to form the token $U_\tau := \mathrm{concat}\big(u_{\tau,1}, \ldots, u_{\tau,K}\big) \in \mathbb{R}^{D_{\mathrm{tot}}}$. Finally, U_τ is projected to the scan width d via a learned linear embedding to form $\widetilde{U}_\tau = W_{\mathrm{in}} U_\tau$, which serves as the input to the Mamba block.

where $\mathbf{X}^m_{a,b} \in (\mathbb{R}^d)^{\otimes m}$. In practice, we truncate the signature at depth M and write $\mathrm{Sig}_M(X)_{a,b} := \big(1, \mathbf{X}^1_{a,b}, \ldots, \mathbf{X}^M_{a,b}\big)$. The signature is invariant under monotone time reparameterisations [23]. If $\gamma : [0,1] \to [a,b]$ is a continuous, non-decreasing, surjective reparameterisation, then

$$\mathrm{Sig}(X)_{a,b} = \mathrm{Sig}(X \circ \gamma)_{0,1}. \tag{7}$$

This ensures the signature depends on the geometric path traced over $[a,b]$ rather than the sampling density, providing natural robustness to irregular sampling once a path interpolation is fixed.

The log-signature is the tensor-algebra logarithm $\mathrm{LogSig}(X)_{a,b} := \log(\mathrm{Sig}(X)_{a,b})$, which lies in the free Lie algebra and, after choosing a Lie basis, provides a compact coordinate representation of the same information. Its truncation inherits the same reparameterisation invariance:

$$\mathrm{LogSig}_M(X)_{a,b} = \mathrm{LogSig}_M(X \circ \gamma)_{0,1}. \tag{8}$$

In LogSig-SSM (Section 3.1), we compute truncated log-signatures on overlapping windows at multiple scales to form a compressed token stream.

3 LogSig-SSM Methodology

We introduce LogSig-SSM, a selective SSM driven by log-signature tokenisation. LogSig-SSM (i) compresses a multivariate input sequence into a sequence of interval descriptors using multi-scale windowed log-signatures, and (ii) processes the resulting token stream with a selective SSM backbone for efficient long-range propagation. Figure 1 summarises the discrete-time implementation.

3.1 Discrete-Time Implementation of LogSig-SSM

Let $x \in \mathbb{R}^{d_x \times T}$ be a multivariate time-series with samples $\{x_j\}_{j=0}^{T-1}$ observed at times $0 = t_0 < \cdots < t_{T-1}$ (not necessarily uniformly spaced). Fix a stride $s \in \mathbb{N}$ and define anchor indices $\tau \in \{1, \ldots, T'\}$ with $T' := \lfloor \frac{T-1}{s} \rfloor$ and anchor times $\tilde{t}_\tau := t_{\tau s}$. Fix $K \in \mathbb{N}$ window scales, specified by hyperparameters $\{L_k\}_{k=1}^K$ (window lengths) and $\{m_k\}_{k=1}^K$ (log-signature truncation depths), where the index k enumerates scales. For each scale k and anchor τ, let $W_{k,\tau}$ denote the windowed samples, $W_{k,\tau} := \{(t_j, x_j) : \max(0, \tilde{t}_\tau - L_k) \le t_j \le \tilde{t}_\tau\}$. Let $\mathcal{W}_k(x)_\tau : [\max(0, \tilde{t}_\tau - L_k), \tilde{t}_\tau] \to \mathbb{R}^{d_x}$ be the bounded-variation path obtained by piecewise-linear interpolation of $W_{k,\tau}$. Define its truncated

log-signature by $\mathrm{LogSig}_{m_k}\big(\mathcal{W}_k(x)_\tau\big) \in \mathfrak{g}^{(\le m_k)}(\mathbb{R}^{d_x})$, where $\mathfrak{g}^{(\le m)}(\mathbb{R}^{d_x})$ is the step-m truncated free Lie algebra over $\mathbb{R}^{d_x}$. We tokenise with the log-signature rather than the signature. The two are related by a polynomial bijection and carry the same information at equal depth (Section 2.3). The log-signature is the redundancy-free coordinate system for that content and the driver of the log-ODE method underlying the NRDE and Log-NCDE models we build on. In implementation the log-signature is represented by its coordinate vector in a fixed basis: let $\phi_{m_k} : \mathfrak{g}^{(\le m_k)}(\mathbb{R}^{d_x}) \to \mathbb{R}^{D_k}$ be the corresponding coordinate map, and define

$$u_{\tau,k} := \phi_{m_k}\Big(\mathrm{LogSig}_{m_k}\big(\mathcal{W}_k(x)_\tau\big)\Big) \in \mathbb{R}^{D_k}.$$

Stacking across K scales yields the LogSig-SSM token

$$U_\tau := \mathrm{concat}\big(u_{\tau,1}, \dots, u_{\tau,K}\big) \in \mathbb{R}^{D_{\mathrm{tot}}}, \qquad D_{\mathrm{tot}} := \sum_{k=1}^{K} D_k. \tag{9}$$

The model dimension (scan width) d is a design choice and may differ from D_{tot}. Accordingly, the scan input is obtained by applying a learned linear embedding to each token,

$$\widetilde{U}_\tau := W_{\mathrm{in}}\, U_\tau \in \mathbb{R}^d, \qquad W_{\mathrm{in}} \in \mathbb{R}^{d \times D_{\mathrm{tot}}}. \tag{10}$$

The resulting sequence $\widetilde{U}_{1:T'}$ is viewed as d channels, indexed by $i \in \{1, \dots, d\}$. The token stream is then processed by a stack of Mamba blocks [14], each governed by the selective scan from Section 2.1. The discretisation step $\Delta_{i,\tau} \in \mathbb{R}_+$, the input projection $s_B(\widetilde{U}_\tau) \in \mathbb{R}^N$, and the output projection $s_C(\widetilde{U}_\tau) \in \mathbb{R}^{1\times N}$ are all functions of the current input $\widetilde{U}_\tau \in \mathbb{R}^d$, where N denotes the state size.

Referencing the ZOH recurrence derived in (2), the per-channel state $H_{i,\tau}$ evolves by

$$H_{i,\tau} = \overline{A}_{i,\tau}\, H_{i,\tau-1} + \overline{B}_{i,\tau}\, \widetilde{U}_{i,\tau}, \qquad y_{i,\tau} = s_C(\widetilde{U}_\tau)\, H_{i,\tau}, \tag{11}$$

where $\overline{A}_{i,\tau} = \exp(\Delta_{i,\tau} A_i)$. Using the small-step approximation from Section 2.1, the driving term simplifies to: $\overline{B}_{i,\tau}\, \widetilde{U}_{i,\tau} \;\approx\; \Delta_{i,\tau}\, s_B(\widetilde{U}_\tau)\, \widetilde{U}_{i,\tau}$.

3.2 Continuous-time Interpretation of LogSig-SSM

3.2.1 LogSig-SSM as a Linear CDE

To obtain a continuous-time interpretation, we adopt the small-step approximation (Section 2.1) standard in selective SSM analysis [7]. Consider the sequence of tokens $\{U_\tau\}_{\tau=1}^{T'}$ from (9). We define a piecewise-constant control path $U : [0,1] \to \mathbb{R}^{D_{\mathrm{tot}}}$ by choosing a partition $0 = r_0 < \cdots < r_{T'} = 1$ and setting:

$$U(t) = U_\tau, \qquad t \in [r_{\tau-1}, r_\tau) \quad \text{for } \tau = 1, \dots, T'. \tag{12}$$

The effective driving signal is then the projection $\widetilde{U}(t) := W_{\mathrm{in}}\, U(t) \in \mathbb{R}^d$. As shown in Appendix A.1, LogSig-SSM admits the Linear CDE representation

$$dH^{\widetilde{U}}_{i;t} = A_i H^{\widetilde{U}}_{i;t}\, d\omega^{\widetilde{U}}_{i;t} + W_B\, d\xi^{\widetilde{U}}_{i;t}, \qquad H^{\widetilde{U}}_{i;0} = h_{i,0}, \qquad i = 1, \dots, d, \tag{13}$$

where $W_B \in \mathbb{R}^{N\times d}$ is the injection matrix, consistent with the discrete-time projection $s_B(u) = W_B\, u$. The system is driven by

$$\omega^{\widetilde{U}}_{i;t} := \int_0^t \sigma(\widetilde{\alpha}_i^\top \widetilde{U}_s + \widetilde{\beta}_i)\, ds \in \mathbb{R}, \qquad \xi^{\widetilde{U}}_{i;t} := \int_0^t \widetilde{U}_s\, \widetilde{U}^i_s\, \sigma(\widetilde{\alpha}_i^\top \widetilde{U}_s + \widetilde{\beta}_i)\, ds \in \mathbb{R}^d, \tag{14}$$

for parameters $\widetilde{\alpha}_i \in \mathbb{R}^d$ and $\widetilde{\beta}_i \in \mathbb{R}$. Since $\widetilde{U}(t) = W_{\mathrm{in}}\, U(t)$ and U is constructed from multi-scale windowed log-signatures of the original path, the drivers remain log-signature-induced functionals of the original trajectory.

3.2.2 Relation to Neural RDEs: Interval-Driven Dynamics

Under the ZOH extension (12), the drivers $(\omega_i^{\widetilde{U}}, \xi_i^{\widetilde{U}})$ are affine on each interval with constant derivatives since $\widetilde{U}_s = \widetilde{U}_\tau$ for $s \in [r_{\tau-1}, r_\tau)$. Summarising each interval by the driver increments $\Lambda_{i,\tau}^{\widetilde{U}} := (\Delta\omega_{i,\tau}^{\widetilde{U}},\ \Delta\xi_{i,\tau}^{\widetilde{U}})^\top$, equation (13) reduces to the NRDE-style integral form for $t \in (r_{\tau-1}, r_\tau)$:

$$H_{i;t}^{\widetilde{U}} = H_{i;r_{\tau-1}}^{\widetilde{U}} + \int_{r_{\tau-1}}^{t} \left[\, A_i H_{i;s}^{\widetilde{U}},\ W_B \,\right] \frac{1}{\Delta r_\tau} \Lambda_{i,\tau}^{\widetilde{U}}\, ds. \tag{15}$$

In particular, $\Delta\omega_{i,\tau}^{\widetilde{U}} = \sigma(\widetilde{\alpha}_i^\top \widetilde{U}_\tau + \widetilde{\beta}_i)\Delta r_\tau$ acts as a content-dependent internal-time allocation per interval. LogSig-SSM's construction is inspired by NRDEs [25] but introduces two key differences. First, NRDEs use the log-signature of the raw path over a single window per interval. LogSig-SSM instead uses a multi-scale token U_τ concatenating log-signatures at K different window lengths, providing descriptors at complementary resolutions within a single token. Second, the drivers $\Lambda_{i,\tau}^{\widetilde{U}}$ are content-dependent, unlike the fixed interval parametrisation in NRDEs. Therefore, LogSig-SSM learns to allocate internal time selectively across the compressed sequence. The derivation of (15) from (13) is provided in Appendix A.2.

4 Theoretical Analysis

We highlight three theoretical attributes of LogSig-SSM. First, we establish a depth-truncation tradeoff by showing that injecting cross-channel path geometry directly into the tokens allows LogSig-SSM to expose non-commutative multivariate features in a single diagonal block that would otherwise require deep, chained Mamba blocks [7]. Second, tokenisation reduces the effective sequence length, cutting selective scan complexity by a factor of T/T' relative to Mamba. Third, LogSig-SSM inherits within-window time-reparameterisation invariance from the log-signature transform, providing robustness to irregular sampling.

4.1 Single-layer expressivity via a log-signature driver lift

A single diagonal Mamba block cannot represent non-commutative cross-channel iterated integrals of its input path; recovering them requires chained Mamba blocks (Cor. 4.4 and Sec. C.2.1 of Cirone et al. [7]). LogSig-SSM bypasses this depth cost by lifting the input before the scan. By construction, the token U_τ contains every log-signature coordinate of the windowed path $\mathcal{W}_k(X)_\tau$ up to depth m_k. The following proposition formalises the resulting depth–truncation tradeoff.

Proposition 4.1 (Depth–truncation tradeoff; Proof in App. A.3)**.** *Fix a compact family $\mathcal{X} \subset C_b^1([0, T_{\mathrm{end}}]; \mathbb{R}^{d_x})$ with $d_x \geq 2$, a window scale (L_k, m_k), an anchor τ, and $2 \leq m \leq m_k$. Let $F : \mathcal{X} \to \mathbb{R}$ be a continuous functional that factors as $F(X) = f(\mathrm{LogSig}_m(\mathcal{W}_k(X)_\tau))$ for some continuous f. Then: **(i)** for every $\varepsilon > 0$, a single LogSig-SSM block followed by a post-scan MLP approximates F uniformly on $\mathcal{X}$ to error ε; **(ii)** there exist such F with $F \notin \mathcal{F}^{(1)}_{\mathrm{Mamba}}[\mathcal{X}]$, where $\mathcal{F}^{(1)}_{\mathrm{Mamba}}[\mathcal{X}]$ denotes the uniform closure on $\mathcal{X}$ of functionals computable by a single diagonal Mamba block on the raw input path (Cor. 4.4 of [7]); **(iii)** any raw-input Mamba approximation of such F requires chained diagonal blocks; recovering depth-m cross-channel features in particular requires chains of depth growing with m in the construction of Cirone et al. [7] (Sec. C.2.1, Prop. 4.5), whereas LogSig-SSM realises F in a single block whenever $m_k \geq m$.*

Remark 4.2 (Multi-scale and asymptotic universality). Proposition 4.1 reflects a depth–truncation tradeoff, not an expressivity gap. The benefit is that LogSig-SSM moves the cost of cross-channel geometry from the online selective scan to an offline, parallel tokeniser. Across scales, any depth-$\leq m_k$ coordinate at scale k enters U_τ linearly; thus, a single LogSig-SSM block captures features that would otherwise require a Mamba chain whose depth grows with $\max_k m_k$. By signature universality [23], in the limit $m_k \to \infty$ a single LogSig-SSM block approximates any continuous functional of windowed paths, consistent with the asymptotic universality of chained diagonal Mamba on raw X (Prop. 4.5 of Cirone et al. [7]).

4.2 Computational Efficiency and Scan Complexity

The selective scan scales linearly with sequence length and constitutes the main time-sequential component of a Mamba layer [14]. Recent studies indicate that Mamba-style models can degrade

at extreme context lengths, which has been linked to length-dependent state decay and a reduced effective memory horizon [3, 37, 40]. Consequently, reducing the effective sequence length is a critical design component for selective SSMs. LogSig-SSM performs this compression explicitly by replacing the length-T input sequence with a token sequence of length $T' \approx T/s$, where $s \in \mathbb{N}$ is the stride. Since the selective scan cost scales as $O(B_{\text{sz}} \cdot T \cdot N \cdot D)$, compression from T to T' reduces scan FLOPs by a factor of $T/T' \approx s$. Additionally, log-signature tokenisation is a one-time offline pre-processing step: precomputed features are not in the training-time backpropagation graph, and computation is highly optimised in libraries such as `iisignature` [29] and `Signatory` [18].

4.3 Robustness to Irregular Sampling

Since log-signatures are invariant under monotone time reparameterisations (Section 2.3), tokens computed from windowed paths encode path geometry rather than sampling density. Under a fixed interpolation, two observation sequences whose windowed paths coincide up to a monotone reparameterisation yield identical tokens and therefore identical model outputs.

Proposition 4.3 (Within-window invariance to monotone time reparameterisation)**.** *Let $\widehat{X}, \widehat{X}' : [0, T_{\text{end}}] \to \mathbb{R}^{d_x}$ be two bounded-variation interpolants and $I_{\tau,k} := [\tilde{t}_\tau - L_k, \tilde{t}_\tau]$ the time interval for window (τ, k). For every scale k and anchor τ, assume there exists a continuous, non-decreasing, surjective map $\gamma_{\tau,k} : I_{\tau,k} \to I_{\tau,k}$ such that $\mathcal{W}_k(\widehat{X}')_\tau(t) = \mathcal{W}_k(\widehat{X})_\tau(\gamma_{\tau,k}(t)), \quad \forall t \in I_{\tau,k}$. Then $U_\tau(\widehat{X}) = U_\tau(\widehat{X}')$ for all τ, and* $\text{LogSigSSM}(\widehat{X}) = \text{LogSigSSM}(\widehat{X}')$.

Proof. The reparameterisation invariance of the signature (cf. (7)) gives $\text{Sig}(\mathcal{W}_k(\widehat{X})_\tau) = \text{Sig}(\mathcal{W}_k(\widehat{X}')_\tau)$. Taking the truncated log-signature and applying the fixed coordinate map ϕ_{m_k} yields $u_{\tau,k}(\widehat{X}) = u_{\tau,k}(\widehat{X}')$. Concatenating across scales gives $U_\tau(\widehat{X}) = U_\tau(\widehat{X}')$, and the claim follows from the determinism of the selective scan. □

The theoretical guarantee is local to each window. However, Section 5.4 provides empirical evidence that this local invariance is sufficient in practice: LogSig-SSM is highly competitive with specialised irregular-time methods on PhysioNet Sepsis, a dataset where only 10.3% of values are observed.

5 Experimental Evaluation

We empirically test the performance of LogSig-SSM across four settings: long-range multivariate classification, high-frequency physiological regression, multivariate weather forecasting, and irregularly sampled clinical prediction. Baselines span SSM/recurrent models, continuous-time models, and general time-series methods. We follow the Log-NCDE protocol [38] for UEA and PPG-DaLiA, the LinOSS protocol [31] for weather forecasting, and the Neural CDE protocol [19] for PhysioNet Sepsis. All experiments were done on a NVIDIA RTX 4090 GPU. Full runtime, memory, and hyperparameter details are in Appendix B and F.

5.1 UEA time-series classification

Following the Log-NCDE protocol [38], we evaluate on the six longest-sequence datasets of the UEA Archive [2], split 70:15:15 over five random seeds. LogSig-SSM achieves the best average accuracy across the sixteen models (71.7%) and the best average rank (2.7) (Table 1).

LogSig-SSM demonstrates a distinct advantage on signal types where standard diagonal SSMs severely struggle, most notably on the ETHANOLCONCENTRATION task (4 classes; chance 25%). While diagonal SSMs fail to surpass 30% accuracy and the best non-SSM baseline reaches 40.5%, LogSig-SSM reaches the highest accuracy of 51.4%. This signal is an absorption curve whose classes differ by smooth shifts in peak shape over extended regions rather than by high-frequency variation (Figure 2), a global trajectory geometry that multi-scale log-signature tokens capture directly and that diagonal SSM backbones cannot resolve on their own.

Beyond accuracy, LogSig-SSM delivers large efficiency gains, summarised for EIGENWORMS in Table 2. On EIGENWORMS (17,984 time points), LogSig-SSM trains in 4 s per 1000 steps versus 122 s for Mamba ($30\times$ faster), while using only 362 MB GPU memory versus 13,486 MB for Mamba ($37\times$ less). Log-signature tokenisation is a one-time offline step taking 1.6 s for the full dataset, and 6 s for all six UEA datasets together, about 1% of a single training run per dataset (Table 6). These

gains come at no accuracy cost: LogSig-SSM leads on EIGENWORMS while requiring $10\times$ less memory than LinOSS-IM.

Table 1: UEA long-sequence classification accuracy (% ↑), mean ± std over five random splits, and average rank over the six datasets (↓, competition ranking, missing entries ranked last). Bold = best, underline = second best, italic = third best per column (ties share a mark). †Transformer OOM on EIGENWORMS; TSPulse not evaluated on EIGENWORMS; *Model evaluated on 1 seed.

Model	EigenWorms	SCP1	SCP2	Ethanol	Heartbeat	Motor	Avg	Rank
Seq. length	17,984	896	1,152	1,751	405	3,000		
# classes	5	2	2	4	2	2		
Continuous-time models								
NRDE [25]	83.9±7.3	80.9±2.5	53.7±6.9	25.3±1.8	72.9±4.8	47.0±5.7	60.6	11.5
NCDE [19]	75.0±3.9	79.8±5.6	53.0±2.8	29.9±6.5	73.9±2.6	49.5±2.8	60.2	11.3
Log-NCDE [38]	85.6±5.1	83.1±2.8	53.7±4.1	34.4±6.4	75.2±4.6	53.7±5.3	64.3	7.7
General time-series models								
Transformer† [24]	OOM	84.3±6.3	49.1±2.5	40.5±6.3	70.5±0.1	50.5±3.0	59.0	10.3
RFormer [24]	90.3±0.0	81.2±2.0	52.3±3.0	34.7±4.0	72.5±0.0	55.8±6.0	64.5	8.0
TS2Vec* [41]	84.7	81.2	57.8	30.8	68.3	51.0	62.3	9.3
MOMENT* [13]	80.9	84.0	47.8	*35.7*	72.2	50.0	61.8	10.2
TSPulse†* [10]	—	83.6	51.1	24.7	70.2	58.0	57.5	11.7
SSM / recurrent models								
LRU [27]	87.8±2.8	82.6±3.4	51.2±3.6	21.5±2.1	78.4±6.7	48.4±5.0	61.7	9.7
S5 [34]	81.1±3.7	**89.9±4.6**	50.5±2.6	24.1±4.3	*77.7±5.5*	47.7±5.5	61.8	9.3
Mamba [14]	70.9±15.8	80.7±1.4	48.2±3.9	27.9±4.5	76.2±3.8	47.7±4.5	58.6	12.5
LinOSS-IMEX [31]	80.0±2.7	87.5±4.0	**58.9±8.1**	29.9±1.0	75.5±4.3	57.9±5.3	65.0	6.7
LinOSS-IM [31]	95.0±4.4	87.8±2.6	58.2±6.9	29.9±0.6	75.8±3.7	60.0±7.5	*67.8*	*4.3*
D-LinOSS [4]	*93.9±3.2*	88.9±3.0	58.6±2.3	29.9±0.6	75.8±4.9	**61.1±2.0**	68.0	3.7
PD-SSM [35]	90.0±5.7	80.9±2.0	56.1±8.6	34.7±4.0	**80.0±2.6**	60.0±3.7	67.0	5.2
LogSig-SSM	**95.3±3.6**	*88.4±1.9*	*58.4±2.9*	**51.4±6.9**	76.9±9.7	59.9±3.4	**71.7**	**2.7**

Table 2: Efficiency on EIGENWORMS ($T = 17{,}984$): GPU memory (MB), wall-clock time per 1000 training steps (s) on one RTX 4090, and test accuracy (%). Full results in Appendix B.

	NCDE	NRDE	Log-NCDE	Mamba	S5	LinOSS-IM	D-LinOSS	**LogSig-SSM**
Mem. (MB)	2484	2506	2510	13486	6646	3488	3488	**362**
Time (s/1k steps)	24595	5386	1956	122	31	14	14	**4**
Accuracy (%)	75.0	83.9	85.6	70.9	81.1	95.0	93.9	**95.3**

5.2 PPG-DaLiA time-series regression

PPG-DALIA [28] is a multimodal physiological regression benchmark for heart-rate estimation from wrist-worn sensors. It contains recordings from 15 subjects during daily activities, each with approximately 150 minutes of data sampled at up to 128 Hz. Following the Log-NCDE protocol [38], recordings are converted into long sequences via a sliding window of length $49{,}920$ and stride $4{,}992$. LogSig-SSM attains MSE 4.46×10^{-2}, a 28% improvement over D-LinOSS [4] (6.16×10^{-2}) and 58% over Mamba [14] (10.65×10^{-2}), the strongest relative gain across all four benchmarks (Table 3, left). On these ultra-long sequences ($49{,}920$ time steps), log-signature tokenisation directly alleviates the long-context degradation that limits standard selective SSMs, making training feasible in regimes where several baselines are memory-constrained.

5.3 Weather Forecasting

Following the precedent set by prior state-space models [15, 31], we evaluate LogSig-SSM's capacity to serve as a general-purpose sequence-to-sequence model on long-horizon forecasting tasks. We focus on the challenging weather prediction benchmark introduced by Zhou et al. [42], specifically targeting the hardest configuration: predicting the future 720 timesteps (one month of hourly multivariate climatological data) based directly on the preceding 720 timesteps. Without incorporating any domain-specific inductive biases or specialised forecasting architectures, LogSig-SSM achieves an MAE of 0.432 (Table 3), reducing the best prior result from D-LinOSS (0.486) by 11% and outperforming all other baselines by a larger margin. These results demonstrate that multi-scale log-signature tokenisation effectively generalises to standard long-horizon forecasting settings.

Table 3: Results on regression, forecasting, and irregular-sampling benchmarks. Left: PPG-DaLiA heart-rate regression, MSE $\times 10^{-2}$ ($\downarrow$), mean $\pm$ std over five runs [28]. Centre: weather forecasting MAE ($\downarrow$), predict 30 days of hourly data from the preceding 30 days; Informer† uses canonical self-attention. Right: PhysioNet Sepsis AUROC ($\uparrow$), 10.3% of values observed; OI = with observation indicators [30]. Bold = best, underline = second best per sub-table.

Regression (PPG-DaLiA)		**Forecasting (Weather)**		**Irregular Sampling (PhysioNet Sepsis)**		
Model	MSE$\times 10^{-2}$	Model	MAE	Method	OI	No OI
NRDE	$9.90_{\pm 0.97}$	Informer	0.731	GRU-Δt	$87.8_{\pm 0.6}$	$84.0_{\pm 0.7}$
NCDE	$13.54_{\pm 0.69}$	Informer†	0.741	GRU-D	$87.1_{\pm 2.2}$	$85.0_{\pm 1.3}$
Log-NCDE	$9.56_{\pm 0.59}$	LogTrans	0.773	GRU-ODE	$85.2_{\pm 1.0}$	$77.1_{\pm 2.4}$
LRU	$12.17_{\pm 0.49}$	Reformer	1.575	ODE-RNN	$87.4_{\pm 1.6}$	$83.3_{\pm 2.0}$
S5	$12.63_{\pm 1.25}$	LSTMa	1.109	Neural CDE	$88.0_{\pm 0.6}$	$77.6_{\pm 0.9}$
S6	$12.88_{\pm 2.05}$	LSTnet	0.757	ANCDE	$90.0_{\pm 0.2}$	$82.3_{\pm 0.3}$
Mamba	$10.65_{\pm 2.20}$	S4	0.578	Neural LSDE	$\underline{90.9_{\pm 0.4}}$	$87.9_{\pm 0.8}$
LinOSS-IMEX	$7.50_{\pm 0.46}$	LinOSS-IMEX	0.508	Neural LNSDE	$\mathbf{91.1_{\pm 0.2}}$	$\underline{88.1_{\pm 0.2}}$
LinOSS-IM	$6.40_{\pm 0.23}$	LinOSS-IM	0.528	Neural GSDE	$\underline{90.9_{\pm 0.1}}$	$\mathbf{88.4_{\pm 0.2}}$
D-LinOSS	$\underline{6.16_{\pm 0.73}}$	D-LinOSS	$\underline{0.486}$			
LogSig-SSM	$\mathbf{4.46_{\pm 0.60}}$	LogSig-SSM	$\mathbf{0.432}$	LogSig-SSM	$90.4_{\pm 0.4}$	$87.4_{\pm 0.8}$

5.4 Sepsis Prediction on PhysioNet (Irregular Sampling)

We evaluate on the PhysioNet 2019 early sepsis prediction challenge [30, 12]. Comprising 40,335 variable-length ICU stays with 34 features, this dataset is extremely sparse (only 10.3% observed values), providing a direct test of the irregular-sampling robustness established in Section 4.3. Predicting sepsis occurrence based on the first 72 hours, we evaluate AUROC both with and without appended observation indicators (OI) (Table 3, right).

LogSig-SSM achieves an AUROC of $90.4/87.4$ (with/without OI). It performs competitively with purpose-built irregular-time models (Neural LSDE, LNSDE, GSDE [26]) and outperforms Neural CDE ($88.0/77.6$) [19], ANCDE ($90.0/82.3$) [17], and all GRU variants. This empirically confirms that within-window reparameterisation invariance yields practical robustness under extreme sparsity.

5.5 Ablation Studies

Table 4 isolates the contributions of LogSig-SSM along two axes, the tokeniser and the backbone, with every variant tuned under the same protocol and budget. Read top to bottom, the descriptor rows decompose the 13.1 pp gap between raw Mamba (58.6%) and LogSig-SSM (71.7%). Compression alone contributes little, $+0.3$ pp with Catch24 statistics that discard the path and $+1.5$ pp with a learned per-channel descriptor that keeps it. Linear channel mixing adds $+2.8$ pp (62.9%), multi-scale spectral content another $+4.7$ pp (67.6%), multi-scale increments, the depth-one log-signature, a further $+1.9$ pp (69.5%), and the second-order terms the final step to 71.7%. This last term, the cross-channel geometry, is the one that neither linear projections nor spectral summaries express. With the tokeniser fixed, S5 and Transformer backbones both far exceed raw Mamba, so the tokens are useful regardless of backbone, while selective recurrence still leads on each of the six datasets.

Truncation depth. The depth-one row of Table 4 is a matched-budget arm that re-searches the tokeniser with the depth fixed at one (Appendix D). Forcing depth one costs 2.3 points on average, almost entirely on the three datasets with few channels (7.2, 4.3 and 2.8 on EIGENWORMS, SCP1 and SCP2). Wavelet scattering makes the content of the second-order terms visible: on those three datasets the depth-one arm is level with Wst-Mamba (-1.0, -1.8, $+2.5$ points) while depth two leads it by 6.2, 2.5 and 5.3, so the margin over the strongest alternative descriptor is exactly the cross-channel geometry of Proposition 4.1. Signature tokens of the same depth and configuration cost 3.1 points on average over the same three datasets and roughly double tokenisation time (Appendix C.1).

Search-free tokeniser. The tokeniser still improves performance without hyperparameter tuning. Keeping the Mamba backbone of the reported LogSig-SSM configuration and replacing the searched tokeniser hyperparameters by a rule based on the sequence length alone ($s = T/100$, lengths $[T/25, T/100]$, depths $[2, 1]$), LogSig-SSM still reaches 64.0% average accuracy, above S5

(61.8), LRU (61.7) and Mamba (58.6), and searching the tokeniser adds the remaining 7.7 points (Appendix F).

Table 4: UEA ablations, average accuracy over the six datasets (% ↑). All variants share the 200-trial Optuna protocol and budget of LogSig-SSM, and its window positions and token dimension where applicable; Δ is relative to raw Mamba. Per-dataset results in Appendix C, Table 7.

Configuration	Avg. acc. (%)	Δ (pp)	What it adds
Window descriptor (all rows compress the sequence), Mamba backbone fixed			
Mamba (raw input)	58.6	–	no compression, no channel mixing
Catch24 statistics [22]	58.9	+0.3	path discarded, summary statistics only
Depthwise convolution	60.1	+1.5	channels kept separate
Standard convolution	62.9	+4.3	first-order (linear) channel mixing
Wavelet scattering	67.6	+9.0	multi-scale spectral content, no cross-channel geometry
Log-signature, depth one	69.5	+10.9	multi-scale increments, no cross-channel content
Log-signature, depth two (**LogSig-SSM**)	**71.7**	**+13.1**	second-order terms: cross-channel geometry, Lévy areas (Prop. 4.1)
Backbone, log-signature tokeniser fixed			
LogSig-S5	63.0	+4.4	LTI recurrence, no selectivity
LogSig-Transformer	65.4	+6.8	attention, no recurrence

6 Conclusion

We introduce LogSig-SSM, which combines multi-scale windowed log-signature tokenisation with a selective SSM backbone for long-horizon time-series modelling. By replacing pointwise inputs with interval descriptors, LogSig-SSM compresses long, irregularly sampled sequences into a shorter token stream while injecting higher-order cross-channel information through log-signature features, enabling scan-efficient long-range propagation while increasing single-block multivariate expressivity.

Beyond its discrete-time formulation, LogSig-SSM extends the linear CDE view of selective SSMs [7] to log-signature-derived driving paths, yielding an NRDE-style interval-driven form in which selectivity allocates internal time across the compressed sequence in a content-dependent manner. Empirically, LogSig-SSM achieves the best or competitive performance across the four benchmarks above while substantially reducing training time and GPU memory relative to Mamba-style baselines. Ablation studies confirm that the gains arise from the geometric content of log-signature tokens, with selective recurrence providing additional margin on tasks requiring content-dependent aggregation.

A limitation for LogSig-SSM is that for genuinely single-channel paths, log-signature tokens reduce to path increments and the higher-order cross-channel structure becomes degenerate, although augmenting the path with time as an additional channel restores non-trivial structure. The token width grows as $O(d_x^m/m)$, so tokenisation time and the size of the input embedding increase rapidly with the number of channels. Finally, window lengths and stride are still selected per dataset (Appendix F). Learned or adaptive window selection is a natural direction for reducing this overhead.

Acknowledgments and Disclosure of Funding

We thank Helena Latorre Moreno for her feedback on the mathematical proofs and Foucault de Jerphanion for his help in the early stages of this work. This work was supported by the UKRI AI Centre for Doctoral Training in Digital Healthcare (grant number EP/Y030974/1).

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework, July 2019. URL http://arxiv.org/abs/1907.10902. arXiv:1907.10902 [cs].

[2] 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, October 2018. URL http://arxiv.org/abs/1811.00075. arXiv:1811.00075 [cs].

[3] Assaf Ben-Kish, Itamar Zimerman, Shady Abu-Hussein, Nadav Cohen, Amir Globerson, Lior Wolf, and Raja Giryes. Decimamba: Exploring the length extrapolation potential of mamba, 2025. URL https://arxiv.org/abs/2406.14528.

[4] Jared Boyer, T. Konstantin Rusch, and Daniela Rus. Learning to Dissipate Energy in Oscillatory State-Space Models, September 2025. URL http://arxiv.org/abs/2505.12171. arXiv:2505.12171 [cs].

[5] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019. URL https://arxiv.org/abs/1806.07366.

[6] Ilya Chevyrev and Andrey Kormilitzin. A Primer on the Signature Method in Machine Learning, January 2016. URL http://arxiv.org/abs/1603.03788. arXiv:1603.03788 [stat].

[7] Nicola Muca Cirone, Antonio Orvieto, Benjamin Walker, Cristopher Salvi, and Terry Lyons. Theoretical Foundations of Deep Selective State-Space Models, January 2025. URL http://arxiv.org/abs/2402.19047. arXiv:2402.19047 [cs].

[8] Tri Dao and Albert Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality, May 2024. URL http://arxiv.org/abs/2405.21060. arXiv:2405.21060 [cs].

[9] Jessilyn Dunn, Lukasz Kidzinski, Ryan Runge, Daniel Witt, Jennifer L. Hicks, Sophia Miryam Schüssler-Fiorenza Rose, Xiao Li, Amir Bahmani, Scott L. Delp, Trevor Hastie, and Michael P. Snyder. Wearable sensors enable personalized predictions of clinical laboratory measurements. *Nature Medicine*, 27(6):1105–1112, June 2021. ISSN 1546-170X. doi: 10.1038/s41591-021-01339-0. URL https://www.nature.com/articles/s41591-021-01339-0.

[10] Vijay Ekambaram, Subodh Kumar, Arindam Jati, Sumanta Mukherjee, Tomoya Sakai, Pankaj Dayama, Wesley M. Gifford, and Jayant Kalagnanam. Tspulse: Tiny pre-trained models with disentangled representations for rapid time-series analysis, 2026. URL https://arxiv.org/abs/2505.13033.

[11] Peter K. Friz and Nicolas B. Victoir. *Multidimensional Stochastic Processes as Rough Paths: Theory and Applications*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2010.

[12] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23):e215–e220, 2000. doi: 10.1161/01.CIR.101.23.e215. URL https://www.ahajournals.org/doi/abs/10.1161/01.CIR.101.23.e215.

[13] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models, 2024. URL https://arxiv.org/abs/2402.03885.

[14] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, May 2024. URL http://arxiv.org/abs/2312.00752. arXiv:2312.00752 [cs].

[15] Albert Gu, Karan Goel, and Christopher Ré. Efficiently Modeling Long Sequences with Structured State Spaces, August 2022. URL http://arxiv.org/abs/2111.00396. arXiv:2111.00396 [cs].

[16] Nikolaus Hautsch. *Modelling Irregularly Spaced Financial Data*, volume 539 of *Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, Heidelberg, 2004. ISBN 978-3-540-21134-1 978-3-642-17015-7. doi: 10.1007/978-3-642-17015-7. URL http://link.springer.com/10.1007/978-3-642-17015-7.

[17] Sheo Yon Jhin, Heejoo Shin, Sujie Kim, Seoyoung Hong, Minju Jo, Solhee Park, Noseong Park, Seungbeom Lee, Hwiyoung Maeng, and Seungmin Jeon. Attentive neural controlled differential equations for time-series classification and forecasting. *Knowl. Inf. Syst.*, 66(3): 1885–1915, November 2023. ISSN 0219-1377. doi: 10.1007/s10115-023-01977-5. URL https://doi.org/10.1007/s10115-023-01977-5.

[18] Patrick Kidger and Terry Lyons. Signatory: differentiable computations of the signature and logsignature transforms, on both cpu and gpu, 2021. URL https://arxiv.org/abs/2001.00706.

[19] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural Controlled Differential Equations for Irregular Time Series. In *Advances in Neural Information Processing Systems*, volume 33, pages 6696–6707. Curran Associates, Inc., 2020.

[20] Shujian Liao, Terry J. Lyons, Weixin Yang, Kevin Schlegel, and Hao Ni. Logsig-rnn: a novel network for robust and efficient skeleton-based action recognition. *CoRR*, abs/2110.13008, 2021. URL https://arxiv.org/abs/2110.13008.

[21] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid transformer-mamba language model, 2024. URL https://arxiv.org/abs/2403.19887.

[22] Carl H. Lubba, Sarab S. Sethi, Philip Knaute, Simon R. Schultz, Ben D. Fulcher, and Nick S. Jones. catch22: CAnonical Time-series CHaracteristics, January 2019. URL http://arxiv.org/abs/1901.10200. arXiv:1901.10200 [cs].

[23] Terry J Lyons, Michael Caruana, and Thierry Lévy. *Differential equations driven by rough paths: Ecole d'Eté de Probabilités de Saint-Flour XXXIV-2004*. Springer, 2007.

[24] Fernando Moreno-Pino, Alvaro Arroyo, Harrison Waldon, Xiaowen Dong, and Álvaro Cartea. Rough Transformers for Continuous and Efficient Time-Series Modelling, March 2024. URL http://arxiv.org/abs/2403.10288. arXiv:2403.10288 [stat].

[25] James Morrill, Cristopher Salvi, Patrick Kidger, James Foster, and Terry Lyons. Neural Rough Differential Equations for Long Time Series, June 2021. URL http://arxiv.org/abs/2009.08295. arXiv:2009.08295 [cs].

[26] YongKyung Oh, Dong-Young Lim, and Sungil Kim. Stable neural stochastic differential equations in analyzing irregular time series data, 2025. URL https://arxiv.org/abs/2402.14989.

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

[28] Attila Reiss, Ina Indlekofer, Philip Schmidt, and Kristof Van Laerhoven. Deep PPG: Large-scale heart rate estimation with convolutional neural networks. *Sensors*, 19(14):3079, 2019. doi: 10.3390/s19143079. URL https://doi.org/10.3390/s19143079.

[29] Jeremy Reizenstein and Benjamin Graham. The iisignature library: efficient calculation of iterated-integral signatures and log signatures, 2018. URL https://arxiv.org/abs/1802.08252.

[30] Matthew Reyna, Chris Josef, Russell Jeter, Supreeth Shashikumar, Benjamin Moody, M. Brandon Westover, Ashish Sharma, Shamim Nemati, and Gari D. Clifford. Early Prediction of Sepsis from Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019. *PhysioNet*, August 2019. doi: 10.13026/v64v-d857. URL https://doi.org/10.13026/v64v-d857. Version 1.0.0.

[31] T. Konstantin Rusch and Daniela Rus. Oscillatory State-Space Models, June 2025. URL http://arxiv.org/abs/2410.03943. arXiv:2410.03943 [cs].

[32] Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-directional equivariant long-range dna sequence modeling, 2024. URL https://arxiv.org/abs/2403.03234.

[33] Siavash Shams, Sukru Samet Dindar, Xilin Jiang, and Nima Mesgarani. Ssamba: Self-supervised audio representation learning with mamba state space model. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, page 1053–1059. IEEE, December 2024. doi: 10.1109/slt61566.2024.10832304. URL http://dx.doi.org/10.1109/SLT61566.2024.10832304.

[34] Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified State Space Layers for Sequence Modeling, March 2023. URL http://arxiv.org/abs/2208.04933. arXiv:2208.04933 [cs].

[35] Aleksandar Terzić, Nicolas Menet, Michael Hersche, Thomas Hofmann, and Abbas Rahimi. Structured sparse transition matrices to enable state tracking in state-space models, 2025. URL https://arxiv.org/abs/2509.22284.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

[37] Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. An empirical study of mamba-based language models, 2024. URL https://arxiv.org/abs/2406.07887.

[38] Benjamin Walker, Andrew D. McLeod, Tiexin Qin, Yichuan Cheng, Haoliang Li, and Terry Lyons. Log Neural Controlled Differential Equations: The Lie Brackets Make a Difference, October 2024. URL http://arxiv.org/abs/2402.18512. arXiv:2402.18512 [cs].

[39] Benjamin Walker, Lingyi Yang, Nicola Muca Cirone, Cristopher Salvi, and Terry Lyons. Structured linear cdes: Maximally expressive and parallel-in-time sequence models, 2025. URL https://arxiv.org/abs/2505.17761.

[40] Zhifan Ye, Kejing Xia, Yonggan Fu, Xin Dong, Jihoon Hong, Xiangchi Yuan, Shizhe Diao, Jan Kautz, Pavlo Molchanov, and Yingyan Celine Lin. Longmamba: Enhancing mamba's long context capabilities via training-free receptive field enlargement, 2025. URL https://arxiv.org/abs/2504.16053.

[41] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series, 2022. URL https://arxiv.org/abs/2106.10466.

[42] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting, 2021. URL https://arxiv.org/abs/2012.07436.

[43] Jingwei Zuo, Maksim Velikanov, Dhia Eddine Rhaiem, Ilyas Chahed, Younes Belkada, Guillaume Kunsch, and Hakim Hacid. Falcon mamba: The first competitive attention-free 7b language model, 2024. URL https://arxiv.org/abs/2410.05355.

A Additional Derivations

A.1 Derivation of the Linear CDE Interpretation of the Selective Scan

We provide the derivation underlying the continuous-time interpretation stated in Section 3.2.1, following the standard small-step approximation used in selective SSM analyses [7] and replacing the pointwise input by the embedded token path $\widetilde{U}$.

In the continuous-time formulation we take $W_B \in \mathbb{R}^{N\times d}$ and identify the discrete-time input projection by $s_B(\widetilde{U}) := W_B\widetilde{U} \in \mathbb{R}^N$. Fix a partition $0 = r_0 < \cdots < r_{T'} = 1$ and extend the embedded token sequence $\{\widetilde{U}_\tau\}_{\tau=1}^{T'}$ to a ZOH path $\widetilde{U} : [0,1] \to \mathbb{R}^d$ by (12). Introduce a base step $\delta > 0$ and rescale

$$\widetilde{\alpha}_i := \alpha_i/\delta, \qquad \widetilde{\beta}_i := \beta_i/\delta, \qquad i = 1,\dots,d. \tag{16}$$

Following [7], we further approximate the softplus gate by a ReLU, $\sigma(x) = \mathrm{ReLU}(x) \approx \log(1+e^x)$, which yields for small δ,

$$\Delta_i(\widetilde{U}) = \mathrm{softplus}(\alpha_i^\top\widetilde{U} + \beta_i) = \mathrm{softplus}\big(\delta(\widetilde{\alpha}_i^\top\widetilde{U} + \widetilde{\beta}_i)\big) \;\approx\; \delta\,\sigma(\widetilde{\alpha}_i^\top\widetilde{U} + \widetilde{\beta}_i). \tag{17}$$

Consequently, as $\delta \to 0$, a first-order expansion of the matrix exponential gives

$$\overline{A}_i(\widetilde{U}) = \exp(\Delta_i(\widetilde{U})A_i) = \exp\big(\delta\,\sigma(\widetilde{\alpha}_i^\top\widetilde{U} + \widetilde{\beta}_i)\,A_i\big) \;\approx\; I + \delta\,\sigma(\widetilde{\alpha}_i^\top\widetilde{U} + \widetilde{\beta}_i)\,A_i. \tag{18}$$

For the injection, we use the standard small-step approximation consistent with the Mamba-style selective scan:

$$\overline{B}_i(\widetilde{U}_\tau)\,\widetilde{U}_\tau^i \;\approx\; s_B(\widetilde{U}_\tau)\,\Delta_i(\widetilde{U}_\tau)\,\widetilde{U}_\tau^i \;\approx\; s_B(\widetilde{U}_\tau)\,\sigma(\widetilde{\alpha}_i^\top\widetilde{U}_\tau + \widetilde{\beta}_i)\,\delta\,\widetilde{U}_\tau^i. \tag{19}$$

Substituting (18)–(19) into the selective recurrence (11) gives the Euler-type update

$$H_{i,\tau} = H_{i,\tau-1} + A_iH_{i,\tau-1}\,\sigma(\widetilde{\alpha}_i^\top\widetilde{U}_\tau + \widetilde{\beta}_i)\,\delta + s_B(\widetilde{U}_\tau)\,\widetilde{U}_\tau^i\,\sigma(\widetilde{\alpha}_i^\top\widetilde{U}_\tau + \widetilde{\beta}_i)\,\delta, \tag{20}$$

which is the forward Euler discretisation of the ODE

$$\dot{H}_{i;t}^{\widetilde{U}} = A_iH_{i;t}^{\widetilde{U}}\,\sigma(\widetilde{\alpha}_i^\top\widetilde{U}_t + \widetilde{\beta}_i) + s_B(\widetilde{U}_t)\,\widetilde{U}_t^i\,\sigma(\widetilde{\alpha}_i^\top\widetilde{U}_t + \widetilde{\beta}_i), \qquad i = 1,\dots,d. \tag{21}$$

Define the drivers

$$\omega_{i;t}^{\widetilde{U}} := \int_0^t \sigma(\widetilde{\alpha}_i^\top\widetilde{U}_s + \widetilde{\beta}_i)\,ds \in \mathbb{R}, \qquad \xi_{i;t}^{\widetilde{U}} := \int_0^t \widetilde{U}_s\,\widetilde{U}_s^i\,\sigma(\widetilde{\alpha}_i^\top\widetilde{U}_s + \widetilde{\beta}_i)\,ds \in \mathbb{R}^d. \tag{22}$$

Then (21) can be rewritten in linear CDE form as

$$dH_{i;t}^{\widetilde{U}} = A_iH_{i;t}^{\widetilde{U}}\,d\omega_{i;t}^{\widetilde{U}} + W_B\,d\xi_{i;t}^{\widetilde{U}}, \qquad H_{i;0}^{\widetilde{U}} = h_{i,0}, \qquad i = 1,\dots,d, \tag{23}$$

which matches (13) in the main text.

For completeness, note that if $\widetilde{U}(t)$ is held constant on $[r_{\tau-1}, r_\tau)$ and the system is evolved for an internal increment $\Delta_{i,\tau}$, the exact ZOH discretisation gives:

$$H_{i,\tau} = \exp(\Delta_{i,\tau}A_i)\,H_{i,\tau-1} + (\Delta_{i,\tau}A_i)^{-1}\big(\exp(\Delta_{i,\tau}A_i) - I\big)\,\big(\Delta_{i,\tau}\,s_B(\widetilde{U}_\tau)\big)\,\widetilde{U}_\tau^i. \tag{24}$$

In particular, taking $\Delta_{i,\tau} = \Delta_i(\widetilde{U}_\tau)$ yields the Mamba-style selective recurrence

$$H_{i,\tau} = \overline{A}_{i,\tau}H_{i,\tau-1} + \overline{B}_{i,\tau}\,\widetilde{U}_\tau^i,$$

with $\overline{A}_{i,\tau} = \exp(\Delta_{i,\tau}A_i)$ and $\overline{B}_{i,\tau} = (\Delta_{i,\tau}A_i)^{-1}\big(\exp(\Delta_{i,\tau}A_i) - I\big)\,(\Delta_{i,\tau}s_B(\widetilde{U}_\tau))$.

A.2 Derivation of the NRDE Interval Form

We derive the interval-driven dynamics stated in Section 3.2.1, starting from the linear CDE (13) under the ZOH extension (12).

Since $\widetilde{U}(t) = \widetilde{U}_\tau$ is constant on $[r_{\tau-1}, r_\tau)$, the drivers defined in (14) are affine on each interval with constant derivatives:

$$\frac{d}{dt}\omega^{\widetilde{U}}_{i;t} = \sigma(\widetilde{\alpha}_i^\top \widetilde{U}_\tau + \widetilde{\beta}_i), \qquad \frac{d}{dt}\xi^{\widetilde{U}}_{i;t} = \widetilde{U}_\tau\, \widetilde{U}^i_\tau\, \sigma(\widetilde{\alpha}_i^\top \widetilde{U}_\tau + \widetilde{\beta}_i), \qquad t \in [r_{\tau-1}, r_\tau). \tag{25}$$

Substituting into (13) gives the ODE on the interval

$$\frac{d}{dt}H^{\widetilde{U}}_{i;t} = A_i H^{\widetilde{U}}_{i;t}\, \sigma(\widetilde{\alpha}_i^\top \widetilde{U}_\tau + \widetilde{\beta}_i) + W_B\, \widetilde{U}_\tau\, \widetilde{U}^i_\tau\, \sigma(\widetilde{\alpha}_i^\top \widetilde{U}_\tau + \widetilde{\beta}_i). \tag{26}$$

Defining the interval increments

$$\Lambda^{\widetilde{U}}_{i,\tau} := \begin{pmatrix} \Delta\omega^{\widetilde{U}}_{i,\tau} \\ \Delta\xi^{\widetilde{U}}_{i,\tau} \end{pmatrix} = \begin{pmatrix} \sigma(\widetilde{\alpha}_i^\top \widetilde{U}_\tau + \widetilde{\beta}_i)\, \Delta r_\tau \\ \widetilde{U}_\tau\, \widetilde{U}^i_\tau\, \sigma(\widetilde{\alpha}_i^\top \widetilde{U}_\tau + \widetilde{\beta}_i)\, \Delta r_\tau \end{pmatrix}, \qquad \Delta r_\tau := r_\tau - r_{\tau-1}, \tag{27}$$

and writing the ODE in the form $\frac{d}{dt}H^{\widetilde{U}}_{i;t} = [A_i H^{\widetilde{U}}_{i;t},\, W_B]\, \frac{1}{\Delta r_\tau}\Lambda^{\widetilde{U}}_{i,\tau}$, integration over $[r_{\tau-1}, t]$ yields

$$H^{\widetilde{U}}_{i;t} = H^{\widetilde{U}}_{i;r_{\tau-1}} + \int_{r_{\tau-1}}^{t} \big[A_i H^{\widetilde{U}}_{i;s},\, W_B\big]\, \frac{1}{\Delta r_\tau}\Lambda^{\widetilde{U}}_{i,\tau}\, ds, \qquad t \in (r_{\tau-1}, r_\tau), \tag{28}$$

which is equation (15) in the main text. The term $\Delta\omega^{\widetilde{U}}_{i,\tau} = \sigma(\widetilde{\alpha}_i^\top \widetilde{U}_\tau + \widetilde{\beta}_i)\Delta r_\tau$ depends on the input content $\widetilde{U}_\tau$, making the internal-time allocation content-dependent rather than fixed as in standard NRDEs.

A.3 Proof of Proposition 4.1

Closure of a single Mamba block. For a generic input path γ, the uniform closure of a single diagonal Mamba block (Cor. 4.4 of Cirone et al. [7]) takes the form

$$\mathcal{F}^{(1)}_{\mathrm{Mamba}}[\mathcal{X}] := \overline{\Big\{(\gamma, t) \mapsto \textstyle\sum_{i=1}^d \psi_i(\omega_t^{\gamma,i}) + \sum_{i=1}^d \int_0^t \phi_i(\omega_t^{\gamma,i} - \omega_s^{\gamma,i})\, d\xi_s^{\gamma,i}\Big\}_{\gamma\in\mathcal{X}}}, \tag{29}$$

with continuous $\psi_i, \phi_i : \mathbb{R} \to \mathbb{R}$ (notation as in Cirone et al. [7]). Each summand depends on γ only through scalar gate paths $\omega^{\gamma,i}$ and ds-driven integrators $\xi^{\gamma,i}$; in particular, no $d\gamma$-driven iterated integral appears.

Proof of (i). By construction, the token $U_\tau \in \mathbb{R}^{D_{\mathrm{tot}}}$ contains every log-signature coordinate of $\mathcal{W}_k(X)_\tau$ up to depth $m_k \geq m$ (Eq. 9). After fixing a Lie basis on the truncated free Lie algebra, the assumed factorisation of F gives $F(X) = \tilde{f}(U_\tau(X))$ for some continuous $\tilde{f} : \mathbb{R}^{D_{\mathrm{tot}}} \to \mathbb{R}$. The map $X \mapsto U_\tau(X)$ is continuous and $\mathcal{X}$ is compact, so the image $\mathcal{U} := \{U_\tau(X) : X \in \mathcal{X}\} \subset \mathbb{R}^{D_{\mathrm{tot}}}$ is compact. Take state dimension $d \geq D_{\mathrm{tot}}$ and $W_{\mathrm{in}} \in \mathbb{R}^{d\times D_{\mathrm{tot}}}$ injective; the image $\widetilde{\mathcal{U}} := W_{\mathrm{in}}\mathcal{U}$ is also compact.

We adopt the small-step parameterisation $\overline{B}_{i,\tau} = \Delta_{i,\tau}\, s_B(\widetilde{U}_\tau)$, implemented in Mamba [14] and used in the linear-CDE bridge of Cirone et al. [7] (cf. Section 3.2.1). Here $s_B(\widetilde{U}) = W_B\widetilde{U}$ and $s_C(\widetilde{U}) = W_C\widetilde{U}$ are linear in $\widetilde{U}$ with learnable parameters W_B, W_C, while $\overline{B}_{i,\tau}$ and $s_C(\widetilde{U}_\tau)$ vary in τ through their input dependence. Since softplus is continuous and strictly positive and $\widetilde{\mathcal{U}}$ is compact, $\Delta_{i,\tau} = \mathrm{softplus}(\alpha_i^\top \widetilde{U}_\tau + \beta_i)$ is uniformly bounded above and below by constants $\Delta_{\max} \geq \Delta_{\min} > 0$, so $\overline{B}_{i,\tau}$ is uniformly bounded. Writing

$$A_i = -\operatorname{diag}(\lambda_{i,1}, \ldots, \lambda_{i,N}), \qquad \lambda_{i,k} \geq \lambda_{\min} > 0,$$

yields

$$\overline{A}_{i,\tau} = \operatorname{diag}\big(e^{-\lambda_{i,1}\Delta_{i,\tau}}, \ldots, e^{-\lambda_{i,N}\Delta_{i,\tau}}\big), \qquad \|\overline{A}_{i,\tau}\| \leq e^{-\lambda_{\min}\Delta_{\min}} \xrightarrow{\lambda_{\min}\to\infty} 0,$$

uniformly on $\widetilde{\mathcal{U}}$.

The recurrence $H_{i,\tau} = \overline{A}_{i,\tau} H_{i,\tau-1} + \overline{B}_{i,\tau}\, \widetilde{U}^i_\tau$ keeps H uniformly bounded; taking $\lambda_{\min} \to \infty$,

$$H_{i,\tau} \;\to\; \Delta_{i,\tau}\, W_B \widetilde{U}_\tau\, \widetilde{U}^i_\tau,$$

being a continuous function of $\widetilde{U}_\tau$ alone. The limiting per-anchor readout vector $\{y_{i,\tau} := s_C(\widetilde{U}_\tau)\, H_{i,\tau}\}_{i=1}^d$ is then a smooth map $g : \widetilde{\mathcal{U}} \to \mathbb{R}^d$, polynomial in $\widetilde{U}_\tau$ modulated by the softplus gate.

For $d \geq D_{\text{tot}}$, g is a smooth map between spaces of matching ambient dimension. There exist parameters $(W_B, W_C, \alpha, \beta, A)$ such that g is injective on the compact set $\widetilde{\mathcal{U}}$. Combined with injectivity of W_{in}, the composed map $U_\tau \mapsto \{y_{i,\tau}\}_{i=1}^d$ is injective on $\mathcal{U}$. Standard MLP universality on the compact image then approximates $\tilde{f}$ uniformly, hence F, to error ε.

Proof of (ii). By Cor. 4.4 of Cirone et al. [7], every feature in $\mathcal{F}^{(1)}_{\text{Mamba}}[\mathcal{X}]$ factorises through scalar gates $\omega^{X,i}$ and ds-driven integrators $\xi^{X,i}$ as in (29); in Mamba, $d\xi^{X,i}_s = X_s X^i_s\, \sigma(\widetilde{\alpha}_i \cdot X_s + \widetilde{\beta}_i)\, ds$ is ds-driven, depending on X only through pointwise values X_s. A pointwise post-scan MLP composed with such block outputs produces only continuous functions of $\mathcal{F}^{(1)}_{\text{Mamba}}[\mathcal{X}]$; since closure features are ds-driven and scalar-gated, no dX-driven iterated integral lies in this composed family. The Lévy area

$$A^{12}_t = \tfrac{1}{2}\int_0^t \left(X^1_s\, dX^2_s - X^2_s\, dX^1_s\right)$$

is a dX-driven iterated integral, antisymmetric under $1 \leftrightarrow 2$; no finite combination of ds-driven features in the closure can reproduce it, so $A^{12} \notin \mathcal{F}^{(1)}_{\text{Mamba}}[\mathcal{X}]$.

Proof of (iii). Direct from Sec. C.2.1 of Cirone et al. [7]: chained diagonal blocks recover progressively richer iterated-integral features as the chain depth grows, populating the state at each anchor with the cross-channel structure that is unavailable at chain depth one. The asymptotic universality of Cirone et al. [7] (Prop. 4.5) requires chain depth growing with the maximum iterated-integral depth being recovered, whereas LogSig-SSM realises any such feature in a single block provided $\exists\, k : m_k \geq m$. □

B Memory and Runtime

Table 5 summarises the compute requirements of each model on the UEA classification benchmarks of Section 5.1. Baseline numbers are taken from [38, 4], and we report parameter count $|\theta|$, GPU memory usage (MB), and wall-clock time per 1000 training steps (s), all measured on a single NVIDIA RTX 4090.

LogSig-SSM is the most compute-efficient approach. It has the lowest memory usage on all six datasets, often an order of magnitude below Mamba, and it is the fastest (or joint fastest) model on all six datasets, with large gains over the continuous-time baselines. These gains follow from the architecture: log-signature tokenisation shortens the effective sequence before the selective scan, reducing both the number of recurrent updates and the activations stored for backpropagation. On the two widest datasets (Heartbeat, 61 channels; MotorImagery, 64 channels) the depth-two tokens are 1,891- and 4,160-dimensional, which raises the parameter count of the input embedding but leaves the footprint at 54 MB and 78 MB.

Tokenisation is a one-time offline computation and is not part of the training graph. Table 6 reports its cost with `iisignature` [29] on CPU, with the Lie basis built once per dataset rather than once per series (the emitted features are identical): all six datasets are tokenised in 6 s in total, about 1% of one training run per dataset. Tokenisation is linear in the sequence length and in the number of series, parallel over both, and the work at each anchor is set by $\max_k L_k$ and $\max_k m_k$ rather than by the length of the stream, so it also runs online: tokens are emitted every s samples with latency bounded by s and a buffer of $\max_k L_k$ samples, and the optional global branch can be maintained at constant cost per anchor through Chen's identity. The binding constraint is channel width, since the depth-m token has $O(d_x^m/m)$ coordinates.

Table 5: Compute requirements for the UEA classification tasks of Section 5. **Bold** indicates the lowest value per dataset for parameter count ($|\theta|$), GPU memory usage (MB), and time (seconds) per 1000 steps. All statistics were recorded on an NVIDIA RTX 4090.

		NRDE	NCDE	Log-NCDE	LRU	S5	Mamba	S6	LinOSS-IMEX	LinOSS-IM	D-LinOSS	LogSig-SSM
EigenWorms	$\|\theta\|$	105110	166789	37977	101129	22007	27381	**15045**	26119	134279	134279	52933
	mem.	2506	2484	2510	10716	6646	13486	7922	6556	3488	3488	**362**
	time	5386	24595	1956	94	31	122	68	37	14	14	**4**
SCP1	$\|\theta\|$	117187	166274	91557	25892	226328	184194	**24898**	447944	991240	992776	310786
	mem.	716	694	724	960	1798	1110	904	4768	4772	4790	**202**
	time	1014	973	635	9	17	7	**3**	42	38	38	**3**
SCP2	$\|\theta\|$	200707	182914	36379	26020	**5652**	356290	26018	448072	399112	399496	352642
	mem.	712	692	714	954	762	2460	1222	4772	2724	2736	**76**
	time	1404	1251	583	9	9	32	7	55	22	22	**2**
Ethanol	$\|\theta\|$	93212	133252	31452	76522	76214	1032772	**5780**	70088	6728	71112	158516
	mem.	712	692	710	1988	1520	4876	938	4766	1182	4774	**82**
	time	2256	2217	2056	16	9	255	**4**	48	8	37	**4**
Heartbeat	$\|\theta\|$	15657742	1098114	168320	338820	158310	1034242	6674	29444	10936	**4356**	396546
	mem.	6860	1462	2774	1466	1548	1650	606	922	928	672	**54**
	time	9539	1177	826	8	11	34	4	4	7	4	**2**
Motor	$\|\theta\|$	1134395	186962	81391	107544	**17496**	228226	52802	106024	91844	20598	70034
	mem.	4552	4534	4566	8646	4616	3120	4056	12708	4510	4518	**78**
	time	7616	3778	730	51	16	35	34	128	11	20	**5**

Table 6: Log-signature tokenisation cost for LogSig-SSM on the UEA datasets, using `iisignature` [29] on CPU: wall-clock seconds for the whole dataset (mean $\pm$ std over five seeds) and the same time as a percentage of one training run.

Dataset	d_x	T	Tokenisation (s)	% of one training run
EigenWorms	6	17,984	1.587 ± 0.004	1.2
SCP1	6	896	0.887 ± 0.051	0.8
SCP2	7	1,152	0.052 ± 0.026	0.1
Ethanol	3	1,751	0.113 ± 0.031	0.1
Heartbeat	61	405	0.959 ± 0.026	0.9
Motor	64	3,000	2.469 ± 0.103	1.5
All six			6.07	

C Ablation Studies on UEA

Table 7 gives the per-dataset results behind Table 4. Every variant uses the same two-stage Optuna protocol and budget as LogSig-SSM (Appendix F), with backbone-appropriate ranges substituted where needed.

Tokeniser variants (Mamba backbone fixed).

- **Mamba (no tokenisation):** the raw input sequence is processed directly by Mamba, with no compression.
- **ConvDepthwise-Mamba:** a 1-D depthwise (per-channel) convolutional layer over each window, projecting to the same token dimension as LogSig-SSM. This isolates compression with no cross-channel mixing.
- **ConvMixing-Mamba:** a standard 1-D convolutional layer over each window, adding linear cross-channel mixing on top of compression.
- **Catch24-Mamba**: Catch24 statistical summaries per window, i.e. the 22 catch22 features [22] together with the mean and standard deviation of the window, computed separately for each channel.
- **Wst-Mamba:** 1-D wavelet-scattering coefficients (Kymatio `Scattering1D` with Morlet wavelets) per window, capturing multi-scale spectral content.

Wst-Mamba is the only variant ahead of LogSig-SSM on any dataset, on EthanolConcentration (53.5 against 51.4, within one standard deviation).

Backbone variants (log-signature tokeniser fixed). The same log-signature tokens are fed to Mamba (selective recurrence), S5 (LTI recurrence) and a standard Transformer (attention). LogSig-SSM (Mamba) leads on each of the six datasets, with the largest margin on EthanolConcentration, where discriminative information is concentrated in content-defined regions of the sequence.

Table 7: UEA ablations (% accuracy ↑, mean $\pm$ std over five splits). Top: tokeniser variants with the Mamba backbone fixed. Bottom: backbone variants with the log-signature tokeniser fixed. All variants share the same window positions, token dimensions where applicable, and Optuna budget (Appendix F). Bold = best per column.

Model	EigenWorms	SCP1	SCP2	Ethanol	Heartbeat	Motor	Avg
Tokeniser variants (Mamba backbone)							
Mamba (raw input)	$70.9_{\pm 15.8}$	$80.7_{\pm 1.4}$	$48.2_{\pm 3.9}$	$27.9_{\pm 4.5}$	$76.2_{\pm 3.8}$	$47.7_{\pm 4.5}$	58.6
ConvDepthwise-Mamba	$78.4_{\pm 3.0}$	$83.9_{\pm 5.8}$	$45.2_{\pm 3.8}$	$35.8_{\pm 12.5}$	$68.7_{\pm 4.3}$	$48.6_{\pm 2.4}$	60.1
ConvMixing-Mamba	$85.9_{\pm 11.8}$	$81.4_{\pm 3.4}$	$51.0_{\pm 3.1}$	$33.6_{\pm 5.6}$	$74.9_{\pm 10.3}$	$50.6_{\pm 3.8}$	62.9
Catch24-Mamba	$72.5_{\pm 11.0}$	$82.4_{\pm 4.7}$	$44.5_{\pm 4.2}$	$34.5_{\pm 5.7}$	$66.8_{\pm 6.1}$	$52.8_{\pm 7.0}$	58.9
Wst-Mamba	$89.1_{\pm 12.5}$	$85.9_{\pm 1.2}$	$53.1_{\pm 5.6}$	$\mathbf{53.5_{\pm 4.0}}$	$72.9_{\pm 6.3}$	$51.0_{\pm 9.5}$	67.6
Backbone variants (log-signature tokeniser)							
LogSig-S5	$85.0_{\pm 4.5}$	$87.5_{\pm 2.3}$	$54.4_{\pm 3.8}$	$25.6_{\pm 2.7}$	$71.6_{\pm 8.5}$	$53.7_{\pm 4.7}$	63.0
LogSig-Transformer	$93.1_{\pm 4.9}$	$83.9_{\pm 3.8}$	$53.7_{\pm 3.1}$	$36.0_{\pm 3.4}$	$72.1_{\pm 8.0}$	$53.7_{\pm 4.6}$	65.4
LogSig-SSM (ours)	$\mathbf{95.3_{\pm 3.6}}$	$\mathbf{88.4_{\pm 1.9}}$	$\mathbf{58.4_{\pm 2.9}}$	$51.4_{\pm 6.9}$	$\mathbf{76.9_{\pm 9.7}}$	$\mathbf{59.9_{\pm 3.4}}$	**71.7**

C.1 Signature versus log-signature tokens

The truncated signature and log-signature are related by a polynomial bijection, so at equal depth they carry identical information and any continuous functional of one is a continuous functional of the other. What the log-signature gives up is universality for linear readouts, which the nonlinear readout of LogSig-SSM does not need. We use the log-signature because it is the driver with which the models we build on are defined (the log-ODE method underlying NRDEs [25] and Log-NCDEs [38]), because it is the redundancy-free coordinate system for that content, and because `iisignature` accumulates it directly in a Lie basis rather than taking a logarithm of the signature. Table 8 compares the two token types directly in accuracy and tokenisation time on EigenWorms, SCP1 and SCP2, the three datasets on which the second-order terms are load-bearing (Appendix D): the log-signature coordinates of the reported configuration are replaced by signature coordinates of the same depth with every other hyperparameter fixed, and since the two coincide at depth one the comparison isolates how the higher levels are coordinatised. The log-signature leads on all three, by 3.1 points on average, and is about twice as fast to compute.

Table 8: Signature against log-signature tokens at the same depth and configuration (Table 11): test accuracy (%, mean over five seeds) and whole-dataset tokenisation time with `iisignature` (s, best of three).

	Accuracy (%)		Tokenisation (s)	
Dataset	Log-signature	Signature	Log-signature	Signature
EigenWorms	**95.3**	94.4	1.56	3.73
SCP1	**88.4**	85.7	0.85	1.34
SCP2	**58.4**	52.9	0.04	0.05
Average / total	**80.7**	77.6	2.45	5.12

D Truncation Depth: Matched-Budget Comparison

A level-one log-signature over a window is its increment and carries no cross-channel content, second-order terms are where cross-channel interactions (Lévy areas) enter the token. To decide the depth without per-dataset selection, we ran, for each dataset, a fresh 100-trial Optuna search over the tokeniser hyperparameters (window lengths, number of scales, stride) with the backbone fixed at the reported configuration (Table 11) and the depth forced to one or to two on every window, so that both arms receive a matched budget at their own best tokeniser configuration. Table 9 reports test accuracy over five seeds.

Table 9: Matched-budget comparison of truncation depth $m_k = 1$ against $m_k = 2$ on every window (UEA accuracy %; mean $\pm$ std over five seeds). Δ is the gain of depth two. The depth-two column is the configuration reported in Table 1.

Dataset	d_x	$m_k = 1$	$m_k = 2$	Δ
EigenWorms	6	$88.1_{\pm 6.4}$	$95.3_{\pm 3.6}$	$+7.2$
SCP1	6	$84.1_{\pm 1.4}$	$88.4_{\pm 1.9}$	$+4.3$
SCP2	7	$55.6_{\pm 4.9}$	$58.4_{\pm 2.9}$	$+2.8$
Ethanol	3	$51.0_{\pm 4.2}$	$51.4_{\pm 6.9}$	$+0.4$
Heartbeat	61	$78.3_{\pm 9.4}$	$76.9_{\pm 9.7}$	-1.4
Motor	64	$59.6_{\pm 7.5}$	$59.9_{\pm 3.4}$	$+0.3$
Average (UEA)		69.5	71.7	$+2.3$

The two directions do not cost the same. Forcing depth one costs 2.3 points on average, with losses of 7.2, 4.3 and 2.8 on the three datasets with few channels, where the second-order terms are load-bearing: on these three datasets the depth-one arm is level with the strongest alternative tokeniser (Wst-Mamba, Table 7; -1.0, -1.8, $+2.5$), while depth two leads it by 6.2, 2.5 and 5.3 points. Forcing depth two costs at most 1.4 points on any dataset and about 0.2 points on average relative to selecting the better depth per dataset, within seed noise everywhere. On Heartbeat and

MotorImagery the depth-two token per branch (1,891 and 2,080 coordinates) is $6.6\times$ and $7.9\times$ wider than the training set, so the second-order coordinates are under-determined and sit on a plateau that we expect to relax with more data. On Ethanol the discriminative structure is carried by the increments, and the extra depth neither helps nor hurts. Depth two is therefore a safe default, its cost being tokenisation time on wide inputs (Table 6), and we report it throughout.

E Signal Visualisation

We visualise representative signals from the six UEA classification tasks used in our evaluation (Figure 2). The goal is to highlight the qualitative differences in temporal structure across datasets, which helps explain the heterogeneous empirical performance observed in Table 1.

The datasets separate naturally into two regimes. **Five datasets** (SelfRegulationSCP1, SelfRegulationSCP2, MotorImagery, Heartbeat, EigenWorms) exhibit *high-frequency, oscillatory, or noise-dominated dynamics*, where discriminative information is often localised in short-time patterns or frequency content. In contrast, **EthanolConcentration** consists of *smooth absorption spectra* with a single broad peak, where class differences manifest as *global shifts in shape and location* over long time horizons.

This distinction is important for interpreting our results. Multi-scale log-signature tokens summarise the *geometry of the path* over extended intervals, capturing global shape characteristics and cross-channel relationships. This is well matched to signals such as EthanolConcentration, where discriminative structure is inherently low-frequency and global. This supports the interpretation that the gains of LogSig-SSM arise from its ability to capture *multi-scale path geometry*, rather than compression alone.

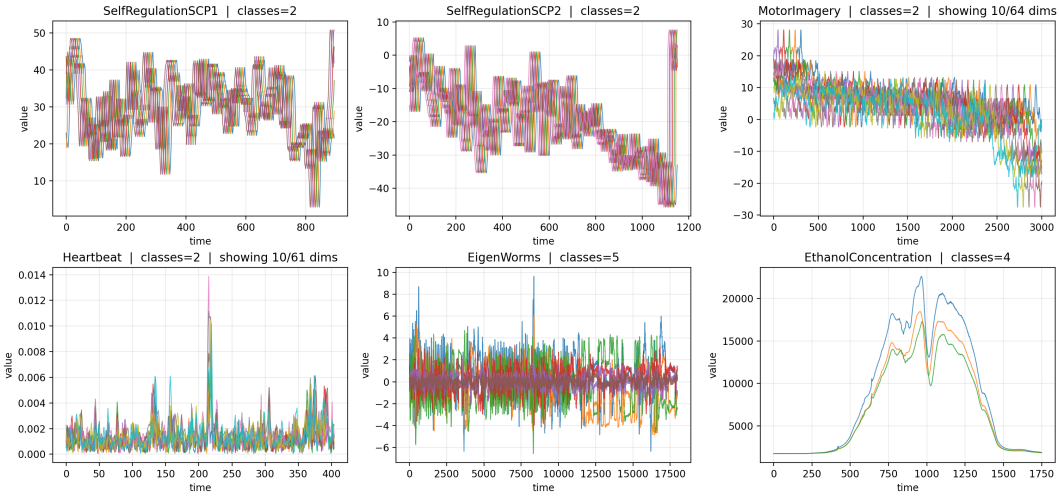


Figure 2: Representative time-series from the six UEA datasets. EthanolConcentration exhibits smooth, low-frequency structure with global shape variation, while the other datasets are dominated by high-frequency or noisy dynamics.

F Selected Hyperparameters

F.1 Hyperparameter optimisation protocol

We optimise log-signature tokenisation and the Mamba backbone using a two-stage Optuna [1] protocol with 100 trials per stage. Stage 1 searches the tokenisation space with the backbone fixed at a small reference Mamba (d=64, 2 layers, N=32, convolution width 3, expansion 2). Stage 2 searches the Mamba backbone jointly with a local neighbourhood of the stage-1 tokenisation. The same protocol and budget are used for the LogSig-S5 and LogSig-Transformer ablations, with backbone-appropriate hyperparameter ranges substituted in Stage 2.

Stage 1 tokenisation ranges. Stride $s \in \{2, \ldots, 199\}$; number of multi-scale branches $K \in \{1, \ldots, 5\}$, bounded by the number of distinct admissible window lengths; window lengths L_k drawn from a T-dependent geometric menu with values T/r for $r \in \{4, \ldots, 64\}$, each lower-bounded by a small constant; an optional global branch (window equal to the full sequence). On UEA the truncation depth is fixed at $m_k = 2$ on every window. On PPG-DaLiA, Weather and PhysioNet the per-branch depth was searched in $\{1, \ldots, m_{\max}\}$ with $m_{\max}$=4 if $d_x \leq 20$ and $m_{\max}$=2 otherwise, and every selected configuration contains a depth-two branch.

Stage 2 local tokenisation neighbourhood. Stride is scaled by $\{0.5, 0.75, 1.0, 1.5, 2.0\}$ around the stage-1 best; the number of branches $K^* \pm 1$; per-branch window lengths are scaled by $\{0.5, 0.75, 1.0, 1.5, 2.0\}$; where the depth is searched, depths within $m_k^* \pm 1$, clamped to $[1, m_{\max}]$.

Stage 2 backbone ranges (Mamba). Learning rate $\in \{10^{-3}, 10^{-4}, 10^{-5}\}$; model dimension $d \in \{16, 64, 128\}$; blocks $\in \{2, 4, 6\}$; state size $N \in \{16, 64, 256\}$; convolution width $\in \{2, 3, 4\}$; expansion $\in \{1, 2, 4\}$; per-feature normalisation $\in \{\text{True}, \text{False}\}$. For LogSig-S5, the layer/block/state-size ranges are searched in place of the Mamba-specific hyperparameters; for LogSig-Transformer, number of layers, attention heads, feedforward dimension, dropout, and learning rate are searched.

Comparison to prior protocols. For fairness with prior work, we adopt the same Mamba hyperparameter grid as the Log-NCDE protocol [38] and its models following this protocol [31, 4]: learning rate, time-channel inclusion, hidden dimension, number of layers, state dimension, convolution width, and expansion factor, with the same value ranges on each axis. Walker et al. [38] search this grid exhaustively, evaluating $3^6 \cdot 2 = 1458$ Mamba configurations (and a comparable 1296-point grid for Log-NCDE itself). We instead use Optuna to evaluate 100 Mamba configurations in Stage 2 over the same axes and ranges. Two consequences follow. First, our Mamba backbone is searched over the same space as the strongest published Mamba results on these benchmarks, so the comparison in Table 1 is structurally fair. Second, our budget is smaller in raw trial count but is allocated adaptively rather than uniformly. Stage 1 separately searches the tokenisation space, so the total budget per dataset is 200 trials, of which 100 involve backbone selection.

Search-free rule. The search is not required to beat raw SSMs. With the Mamba backbone fixed at the reported LogSig-SSM configuration of each dataset and every tokeniser hyperparameter set from the sequence length alone ($s = T/100$, lengths $[T/25, T/100]$, depths $[2, 1]$), applied unchanged to all six datasets, LogSig-SSM reaches 64.0% average UEA accuracy (Table 10), above S5 (61.8), LRU (61.7) and Mamba (58.6), and searching the tokeniser adds the remaining 7.7 points. The full per-axis study is provided with the code.

Table 10: Search-free rule on UEA (accuracy %, mean $\pm$ std over five seeds): tokeniser hyperparameters set from the sequence length alone, Mamba backbone as in Table 11, against the searched configuration and the raw SSM baselines of Table 1.

Model	EigenWorms	SCP1	SCP2	Ethanol	Heartbeat	Motor	Avg
LogSig-SSM (searched)	95.3±3.6	88.4±1.9	58.4±2.9	51.4±6.9	76.9±9.7	59.9±3.4	71.7
LogSig-SSM (rule, no search)	92.5±6.8	85.5±3.5	51.9±8.9	32.3±2.1	72.2±7.3	49.7±4.8	64.0
S5	81.1±3.7	89.9±4.6	50.5±2.6	24.1±4.3	77.7±5.5	47.7±5.5	61.8
LRU	87.8±2.8	82.6±3.4	51.2±3.6	21.5±2.1	78.4±6.7	48.4±5.0	61.7
Mamba	70.9±15.8	80.7±1.4	48.2±3.9	27.9±4.5	76.2±3.8	47.7±4.5	58.6

F.2 Log-signature tokeniser

Table 11: Selected hyperparameters for **LogSig-SSM**. LR: learning rate; Hidden: model dimension d; State: state size N; Blocks: number of Mamba blocks; Time: time channel appended to the input; Conv: convolution width; Exp: expansion factor; Norm: per-feature normalisation; Lengths/Depths: window lengths L_k and truncation depths m_k; Global: depth of the optional global branch (– = none).

Dataset	LR	Hidden	State	Blocks	Time	Conv	Exp	Norm	Lengths / Depths	Stride	Global
EigenWorms	1e-3	64	64	2	F	2	1	T	[281] / [2]	21	–
Ethanol	1e-3	16	256	6	T	2	2	T	[18, 132] / [2, 2]	66	–
Heartbeat	1e-3	128	64	2	F	4	1	T	[143] / [2]	26	–
Motor	1e-4	16	16	2	T	3	1	F	[51, 16] / [2, 2]	50	–
SCP1	1e-3	128	64	2	F	3	2	T	[56] / [2]	5	1
SCP2	1e-4	128	16	2	T	2	3	T	[36] / [2]	46	–
PPG-DaLiA	1e-3	16	128	4	T	3	4	T	[64, 8, 8] / [2, 2, 1]	64	–
Weather	1e-3	32	32	6	F	2	2	T	[20, 30, 90] / [1, 2, 1]	7	1
PhysioNet (OI)	1e-3	64	128	6	F	4	2	T	[6, 2, 2] / [1, 2, 1]	1	–
PhysioNet (no OI)	5e-4	32	16	4	F	4	2	T	[4, 6, 22] / [2, 2, 2]	1	–

F.3 Ablation variants

Tables 12–15 list the selected configurations of the LogSig-S5, LogSig-Transformer, Catch24-Mamba and Wst-Mamba variants.

Table 12: Selected hyperparameters for **LogSig-S5** (log-signature tokens + S5). Blocks are reported as layers×blocks.

Dataset	LR	Hidden	State	Blocks	Time	Norm	Lengths	Depths	Stride	Global
EigenWorms	1e-3	64	16	4×4	F	F	[562, 281]	[2, 2]	57	F
Ethanol	1e-3	16	64	2×2	F	T	[109, 43]	[2, 1]	70	F
Heartbeat	1e-5	64	16	6×2	T	T	[14]	[1]	3	F
Motor	1e-4	64	16	4×4	T	F	[93, 250, 125]	[2,1,2]	35	T
SCP1	1e-3	16	256	6×4	F	F	[112, 18]	[1, 1]	30	F
SCP2	1e-4	16	64	4×2	F	T	[18, 144, 192]	[4,2,3]	13	T

Table 13: Selected hyperparameters for **LogSig-Transformer** (log-signature tokens + Transformer). FFN: feed-forward dimension; Global: depth of the optional global branch (– = none).

Dataset	LR	Hidden	Heads	FFN	State	Blocks	Time	Norm	Lengths	Depths	Stride	Global
EigenWorms	1e-3	128	4	256	64	2	T	T	[374, 899]	[2, 1]	55	–
Ethanol	1e-3	16	8	128	64	6	T	F	[22, 48, 114]	[1, 1, 1]	74	–
Heartbeat	1e-5	16	8	256	64	4	T	T	[16, 25]	[1, 1]	14	–
Motor	1e-4	16	8	512	64	4	T	T	[250, 46]	[1, 1]	46	–
SCP1	1e-4	16	8	512	64	4	T	T	[74]	[1]	14	1
SCP2	1e-3	128	8	512	64	2	T	T	[68]	[2]	76	–

Table 14: Selected hyperparameters for **Catch24-Mamba** (Catch24 tokens).

Dataset	LR	Hidden	State	Blocks	Time	Conv	Exp	Norm	Lengths	Stride
EigenWorms	1e-3	64	256	6	T	2	4	T	[70, 749, 499, 70, 35]	135
Ethanol	1e-3	128	256	4	T	3	3	T	[109, 350, 13, 109, 350]	79
Heartbeat	1e-5	128	64	2	T	3	2	F	[11, 16, 50, 8, 3]	38
Motor	1e-3	64	16	6	F	3	3	T	[187, 11]	163
SCP1	1e-4	64	16	2	T	4	2	T	[179, 9, 179, 32, 224]	98
SCP2	1e-3	64	256	6	T	4	3	T	[192, 48, 115, 192]	67

Table 15: Selected hyperparameters for **Wst-Mamba** (wavelet-scattering tokens, Kymatio `Scattering1D` with Morlet wavelets).

Dataset	LR	Hidden	State	Blocks	Conv	Exp	WST Lengths	Stride
EigenWorms	1e-3	128	64	6	2	1	[749, 2997]	48
Ethanol	1e-4	128	64	2	3	4	[18, 291, 62, 48]	66
Heartbeat	1e-4	64	64	2	3	4	[67, 67, 40]	75
Motor	1e-5	16	16	2	2	1	[375, 125, 150, 187]	125
SCP1	1e-5	128	256	2	4	2	[179, 28]	122
SCP2	1e-3	16	16	2	3	1	[48, 4, 96, 48, 192]	141

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately outline the contributions of the LogSig-SSM model, its theoretical properties, and its empirical evaluations on UEA, PPG-DaLiA, weather forecasting, and PhysioNet Sepsis.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper states its limitations in the Conclusion: degenerate cross-channel structure on single-channel paths, token width growing as $O(d_x^m/m)$ on wide inputs, and window lengths and stride still selected per dataset.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical claims, such as the depth-truncation tradeoff and within-window time-reparameterisation invariance, are stated with clear assumptions and formal proofs provided either in the main text or the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The complete source code, the tokeniser, all training scripts and the per-run configuration files (including seeds) and outputs of every reported number are publicly available (URL in the abstract). Every benchmark follows a published protocol whose data, splits and metrics are specified in Section 5 and Appendix F.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code, data-preparation scripts and the configuration of every reported run are publicly available at the URL given in the abstract, together with the selected hyperparameters listed in Appendix F.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Comprehensive tables outlining all selected hyperparameters (learning rates, hidden dimensions, state sizes, sequence lengths, strides) and model configurations are provided in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main empirical results report the mean and standard deviation across multiple random splits or runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The Appendix details that all experiments were conducted on an NVIDIA RTX 4090 and provides comprehensive metrics on parameter counts, memory usage, and wall-clock training times per 1000 steps.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to standard ethical practices, evaluating foundational architectural improvements on established, publicly accessible datasets.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: LogSig-SSM is a general-purpose time-series architecture. As such, it inherits the dual-use considerations of efficient sequence models: applications to clinical and physiological data (e.g. PhysioNet Sepsis, PPG-DaLiA) raise standard concerns around dataset bias, calibration drift across populations, and downstream deployment in clinical

decision-making, none of which we evaluate. The efficiency gains may also enable training in resource-constrained settings, which we view as a positive.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper proposes a general-purpose time-series architecture and does not release pre-trained models or assets posing a high risk for public misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper properly cites all benchmark datasets used (e.g., UEA Archive, PPG-DaLiA, PhysioNet 2019) and software libraries utilised (e.g., `iisignature`).

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The work introduces a new modelling methodology rather than a new standalone dataset or foundational asset requiring separate structured documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The study relies exclusively on existing public datasets and conducts no new crowdsourcing or direct human subject research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: As the experiments only utilize retrospective, de-identified public datasets, new IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Large Language Models (LLMs) are not a component of the core methodology proposed in this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.