BioOSS: A Bio-Inspired Oscillatory State System with Spatio-Temporal Dynamics

Zhongju Yuan[1*] **Geraint Wiggins**[2,3] **Dick Botteldooren**[1]
[1]WAVES Research Group, Ghent University, Gent, Belgium
[2]AI Lab, Vrije Universiteit Brussel, Brussel, Belgium
[3]EECS, Queen Mary University of London, London, UK
zhongju.yuan@ugent.be, geraint.wiggins@vub.be, dick.botteldooren@ugent.be

Abstract

Today's deep learning architectures are primarily based on perceptron models, which do not capture the oscillatory dynamics characteristic of biological neurons. Although oscillatory systems have recently gained attention for their closer resemblance to neural behavior, they still fall short of modeling the intricate spatio-temporal interactions observed in natural neural circuits. In this paper, we propose a **bio**-inspired **o**scillatory **s**tate **s**ystem (BioOSS) designed to emulate the wave-like propagation dynamics critical to neural processing, particularly in the prefrontal cortex (PFC), where complex activity patterns emerge. BioOSS comprises two interacting populations of neurons: p neurons, which represent simplified membrane-potential-like units inspired by pyramidal cells in cortical columns, and o neurons, which govern propagation velocities and modulate the lateral spread of activity. Through local interactions, these neurons produce wave-like propagation patterns. The model incorporates trainable parameters for damping and propagation speed, enabling flexible adaptation to task-specific spatio-temporal structures. We evaluate BioOSS on both synthetic and real-world tasks, demonstrating superior performance and enhanced interpretability compared to alternative architectures.

1 Introduction

The advent of deep learning models, such as transformers [Vaswani et al., 2017, Devlin et al., 2019], has significantly advanced performance across a variety of tasks and domains. However, these models are still fundamentally based on perceptrons, which differ from the oscillatory behavior of biological neurons. In contrast, oscillatory systems [Rusch and Mishra, 2021, Lanthaler et al., 2023, Rusch and Rus, 2025] have recently garnered considerable attention due to their closer alignment with the dynamic properties of biological neurons. While these models capture certain temporal aspects of neural oscillations, they fail to replicate the complex spatio-temporal interplay characteristic of biological neural circuits. This limitation arises because biological oscillations are typically driven by coupled neurons with both temporal and spatial interactions. Consequently, existing oscillatory models primarily capture temporal dependencies; even when spatial couplings are included, they do not account for the structured spatial organization and physical distances inherent in real neural circuits.

We propose a **bio**-inspired **o**scillatory **s**tate **s**ystem (BioOSS) to model the spatio-temporal wave dynamics observed in neural circuits. In the brain, neural states evolve not only over time but also through spatial interactions with neighboring neurons, forming the basis for neural dynamic computation. Such dynamics are particularly salient in the prefrontal cortex (PFC), where complex

[*] Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

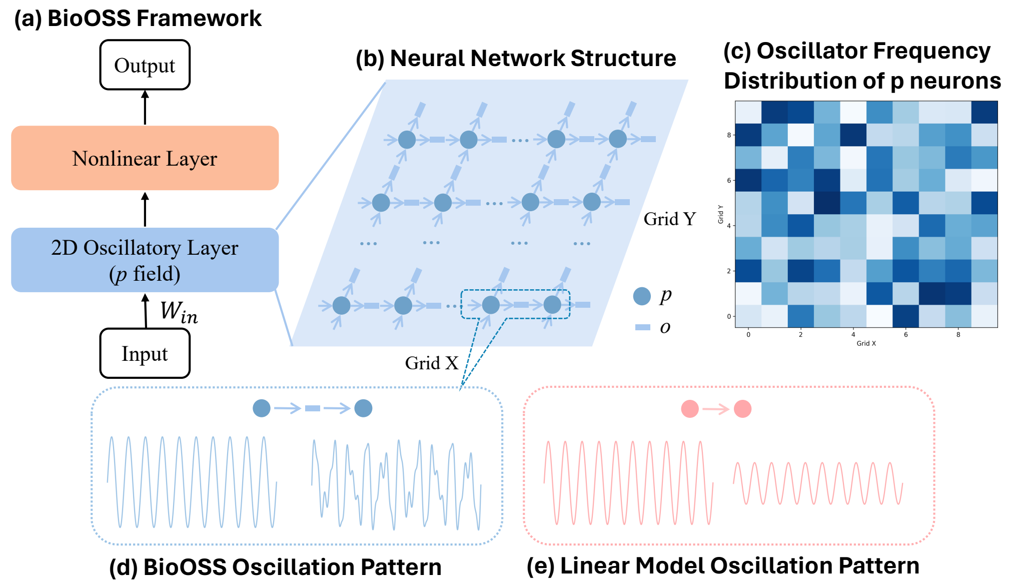

(a) BioOSS Framework
Output
Nonlinear Layer
2D Oscillatory Layer
(p field)
W_{in}
Input
(b) Neural Network Structure
Grid Y
Grid X
p
o
(c) Oscillator Frequency Distribution of p neurons
Grid Y
Grid X
0
2
4
6
8
(d) BioOSS Oscillation Pattern
(e) Linear Model Oscillation Pattern

Figure 1: **(a)** The Overview of the proposed BioOSS framework. **(b)** Structure of the 2D neural network composed of interacting p (pressure-like) and o (oscillation-like) neurons arranged on a spatial grid. **(c)** Distribution of the natural frequencies across the grid, illustrating local frequency heterogeneity. **(d)** Dynamic behavior of p neurons in the proposed model with local interactions. The left panel shows the input signal to a left p neuron; the right panel shows the enriched oscillatory pattern of a neighboring right p neuron. **(e)** Behavior in the absence of local spatial interactions, where the right p neuron exhibits only amplitude modulation without generating richer oscillatory dynamics.

and coordinated activity patterns support higher-order cognitive functions [Dale and Halgren, 2001, Buonomano and Maass, 2009]. Temporal recurrence and delay mechanisms enable memory and integration [Fuster and Bressler, 2012, Richards and Frankland, 2017], while spatially structured cortical columns facilitate localized wave propagation [Watakabe et al., 2023]. BioOSS captures these properties in a unified framework (Fig. 1a), offering a biologically grounded approach to modeling neural dynamics and enabling the development of more generalizable and interpretable artificial systems.

As observed in [Watakabe et al., 2023], neural circuits tend to organize primary signal-carrying units into structured rows rather than random patterns. To capture this property, we propose a 2D grid-based framework composed of two interacting neuronal populations: p and o neurons (Fig. 1b). p neurons serve as the main signal carriers in our model, functionally inspired by pyramidal neurons in cortical columns as described in [Watakabe et al., 2023], and form a distributed oscillatory pattern with diverse frequencies (Fig. 1c). In parallel, o neurons represent propagation velocities, mimicking diffuse projections that modulate activity across cortical tissue. The local interactions between neurons further enrich the oscillatory dynamics, giving rise to complex patterns beyond simple harmonic components (Fig. 1d). In contrast, purely linear transformations, such as fully connected layers without non-linearity, only reweight and shift existing components without introducing new frequency content (Fig. 1e).

The p and o neurons are locally interconnected, enabling wave-like lateral signal propagation. To introduce biologically plausible flexibility, we incorporate trainable parameters, including damping k and propagation speed c, which reflect region-specific attenuation and projection density [Alcamí and Pereda, 2019, Campagnola et al., 2022]. By learning these parameters from data, the model can adaptively generate task-relevant spatio-temporal patterns. Furthermore, the spatio-temporal couplings allow us to derive connectome eigenmodes for each grid [Xia et al., 2024], enriching the oscillatory dynamics and improving performance on time series tasks.

In summary, our contributions are:

- We introduce BioOSS, a biologically inspired oscillatory system that models simplified spatio-temporal wave propagation, drawing analogies to cortical column dynamics without aiming for full biological realism.

- We design an efficient trainable framework with flexible damping and propagation speed parameters, enabling adaptive and stable spatio-temporal pattern generation across tasks.

- We validate BioOSS through extensive experiments on synthetic and real-world datasets, demonstrating both competitive performance and enhanced interpretability compared to baseline architectures.

2 Related Work

Neuroscience Inspiration. Our model is inspired by neuroscience findings that cognitive functions and information processing emerge from synchronized oscillations across coupled neuronal populations. These neural oscillations, arising from the interaction between excitatory and inhibitory processes, exhibit rhythmic, coordinated activity patterns [Doelling and Assaneo, 2021, Esghaei et al., 2022]. Recent studies [Watakabe et al., 2023] show that neurons in the prefrontal cortex (PFC) are organized in a discrete, mosaic-like structure, where signal-carrying units communicate diffusely via intermediate neurons. This spatial organization enables circuits to generate complex oscillatory dynamics from multi-dimensional interactions. Building on this biological insight, we design a simplified 2D neural network composed of a grid of primary neurons coupled with intermediate neurons. By training the damping coefficients and propagation speeds, our model simulates wave-like dynamics [Foster and Scheinost, 2024] and resonance phenomena [Rigotti et al., 2013] observed in biological circuits.

Comparison to Existing Sequence Modeling Approaches. Sequence modeling is central to capturing spatio-temporal dependencies in both biological and artificial systems. Transformers [Wen et al., 2023] capture temporal relationships through attention, RNNs [Torres et al., 2021] maintain hidden states across time, and State Space Models (SSMs) [Auger-Méthé et al., 2021] describe latent state evolution.

Oscillatory extensions move beyond perceptron-like updates. LinOSS [Rusch and Rus, 2025] exemplifies this by introducing second-order ODE dynamics to model rhythmic activity within scan-based architectures, while providing stability guarantees. BioOSS builds on this line of work but diverges in key ways: it employs a 2D neural grid with local couplings instead of 1D hidden states, grounds its recurrence in first-order PDEs inspired by wave propagation rather than ODEs, and incorporates distinct signal-bearing and modulatory units to yield spatially structured, biologically interpretable dynamics. These design choices result in different numerical properties and spatio-temporal representations.

Beyond LinOSS, recent works have explored oscillatory and wave-like computation using varied abstractions. Neural Wave Machines [Keller and Welling, 2023] employ convolutional couplings on a topographic grid, spectral approaches [Balkenhol et al., 2024] use graph filters, and analog substrates such as Wave-RNNs [Hughes et al., 2019] or water-wave reservoirs [Maksymov, 2023] approximate physical propagation. These methods highlight the growing interest in wave-based computation, but approach it through different mechanisms, such as matrix couplings, graph filters, analog substrates, or through PDE formulations with different goals (e.g., spatial convolutional operators), rather than a discretized cortical-inspired PDE system as in BioOSS.

BioOSS extends this landscape by introducing a wave-PDE-driven, discretized dynamical system compatible with scan operations. Its analytical eigendecomposition yields efficient Fourier-domain recurrence, enabling closed-form characterization of oscillatory modes and frequency responses. This combination of spatial structure, biological grounding, and computational tractability distinguishes BioOSS from both LinOSS and other oscillatory models, while maintaining interoperability with structured SSM frameworks.

3 Methods

3.1 Overview of the proposed model

We propose BioOSS, illustrated in Fig. 1a, whose core component is a two-dimensional oscillatory layer (Fig. 1b). This biologically motivated model consists of two neuron types: p neurons, which serve as the primary computational units and carry the signal, and o neurons, which modulate the velocity of information propagation. The spatio-temporal dynamics of the system are described by the following equations:

$$\frac{\partial p}{\partial t} + k^p p + c^2 \nabla \cdot \mathbf{o} = \mathbf{B}\mathbf{u}(t), \tag{1}$$

$$\frac{\partial \mathbf{o}}{\partial t} + \mathbf{k}^o \mathbf{o} + \nabla p = 0, \tag{2}$$

where p denotes the p neuron value, $\mathbf{o} = (o_x, o_y)$ represents the o neuron values in the x and y directions, k^p and $\mathbf{k}^o$ are the damping coefficients, c is the wave speed, and $\mathbf{u}(t)$ is the external input at time step t, while $\mathbf{B}$ is the input weight matrix. For convenience, the system can be expressed in matrix form as:

$$\mathbf{x}'(t) = \begin{bmatrix} -k^p & -c^2\nabla\cdot \\ -\nabla & -\mathbf{k}^o \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \mathbf{u}(t) = \mathbf{A}\mathbf{x}(t) + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} \mathbf{u}(t), \tag{3}$$

where $\mathbf{x}(t) = [p, \mathbf{o}]^T$ and $\mathbf{A}$ is the coupling matrix.

The output equation is given by:

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \tag{4}$$

where $\mathbf{C}$ and $\mathbf{D}$ are the linear weight matrices.

3.2 Discretization Scheme

To mitigate the significant computational cost associated with solving the PDE at each time step, we propose an explicit discretization scheme that generates an explicit state transition matrix. This approach avoids the need for a solver, enabling more efficient computation.

We begin by updating the wave propagation with a fixed timestep Δt as follows:

$$\mathbf{o}_n^* = \mathbf{o}_{n-1} - \Delta t \nabla p_{n-1}, \tag{5}$$

$$p_n^* = p_{n-1} - c^2 \Delta t \nabla \cdot \mathbf{o}_n^* + \Delta t \mathbf{B}\mathbf{u}_n. \tag{6}$$

Next, we apply a damping correction as follows:

$$(\mathbf{I} + \Delta t \mathbf{k}^o)\mathbf{o}_n = \mathbf{o}_n^*, \tag{7}$$

$$(1 + \Delta t k^p) p_n = p_n^*. \tag{8}$$

The full update equation is then given by:

$$\mathbf{x}_n = \mathbf{M}^{\text{Damp}^{-1}} \mathbf{M}^{\text{Velocity}} \mathbf{x}_{n-1} + \mathbf{M}^{\text{Damp}^{-1}} \mathbf{F}_n^{\text{Velocity}} = \mathbf{A}\mathbf{x}_{n-1} + \mathbf{M}^{\text{Damp}^{-1}} \mathbf{F}_n^{\text{Velocity}}, \tag{9}$$

where $\mathbf{M}^{\text{Velocity}} = \begin{bmatrix} 1 & -c^2\Delta t\nabla\cdot \\ -\Delta t \nabla & \mathbf{I} \end{bmatrix}$, $\mathbf{M}^{\text{Damp}} = \begin{bmatrix} 1 + \Delta t k^p & 0 \\ 0 & \mathbf{I} + \Delta t \mathbf{k}^o \end{bmatrix}$, and $\mathbf{F}_n^{\text{Velocity}} = \begin{bmatrix} \Delta t \mathbf{B}\mathbf{u}_n \\ 0 \end{bmatrix}$.

Although $\mathbf{A} = \mathbf{M}^{\text{Damp}^{-1}} \mathbf{M}^{\text{Velocity}}$ serves as a coupling matrix to capture internal propagation.

Since $\mathbf{M}^{\text{Damp}}$ is diagonal by construction, its inverse is obtained by taking the reciprocal of its diagonal entries:

$$\mathbf{M}^{\text{Damp}^{-1}} = \text{diag}\left(\frac{1}{1 + \Delta t\, k_{i,j}^p}, \frac{1}{1 + \Delta t\, k_{i,j}^o} \right), \tag{10}$$

where each diagonal entry is simply the reciprocal of the corresponding factor $(1 + \Delta t\, k^p_{i,j})$ or $(1 + \Delta t\, k^o_{i,j})$ at grid point (i, j).

Therefore, the coupling matrix becomes the following form:

$$\mathbf{A} = \mathbf{M}^{\text{Damp}^{-1}}\mathbf{M}^{\text{Velocity}} = \begin{pmatrix} (\mathbf{I} + \Delta t\mathbf{k}^p)^{-1} & -\mathbf{c}^2\Delta t(\mathbf{I} + \Delta t\mathbf{k}^o)^{-1}\nabla\cdot \\ -\Delta t(\mathbf{I} + \Delta t\mathbf{k}^p)^{-1}\nabla & (\mathbf{I} + \Delta t\mathbf{k}^o)^{-1} \end{pmatrix} \tag{11}$$

Since both $\mathbf{c}$, $\mathbf{k}^p$ and $\mathbf{k}^o$ are matrices with potentially different values at each grid point, we need to perform a local analysis. For each grid point (i, j), we examine the local evolution of the system.

3.3 Efficient Recurrent Scan Operator

The computational efficiency of BioOSS is limited because the recurrence is inherently sequential and cannot be parallelized, while the spatial operators (gradient and divergence) must be applied at every time step. To alleviate this cost, we adopt the scan operator together with an eigendecomposition-based approximation of the system matrix, which reduces the burden of repeatedly applying the spatial operators.

The main challenge is that the recurrence induced by these operators breaks the associativity property $(x\bullet y)\bullet z = x\bullet(y\bullet z)$ [Rusch and Rus, 2025]. Here, x, y, z denote intermediate system states, typically represented as tuples $(\mathbf{A}, \mathbf{M}^{\text{Damp}^{-1}}\mathbf{F}^{\text{Velocity}})$ combining the system matrix and the input vector at each step, and $\bullet$ is a binary operation such as $(\mathbf{a}_1, \mathbf{a}_2) \bullet (\mathbf{b}_1, \mathbf{b}_2) = (\mathbf{b}_1\mathbf{a}_1, \mathbf{b}_1\mathbf{a}_2 + \mathbf{b}_2)$ [Rusch and Rus, 2025]. As emphasized in recent work on oscillatory state space models [Rusch and Rus, 2025], associativity is essential because it enables parallel scan operations, reducing the computational complexity of a serial recurrence from $\mathcal{O}(N)$ to $\mathcal{O}(\log N)$. Following SSM literature [Orvieto et al., 2023, Rusch and Rus, 2025], we therefore apply eigendecomposition $\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$, where $\mathbf{\Lambda}$ is diagonal. This diagonalization restores associativity and allows efficient scan operations across time steps, significantly lowering the computational cost of sequential updates.

To find the eigenvalues, we transform to the Fourier domain where gradient, and divergence operators become algebraic in Fourier space, as detailed in the proof provided in the Appendix Section A. For spatial frequency components (ξ_x, ξ_y) and for each grid point (i, j) with local values of $c_{i,j}$, $k^p_{i,j}$, and $k^o_{i,j}$:

$$\nabla p \to (i\xi_x p, i\xi_y p) \tag{12}$$

$$\nabla \cdot \mathbf{o} \to i\xi_x o_x + i\xi_y o_y \tag{13}$$

For each Fourier mode (ξ_x, ξ_y) and each local grid point, the system matrix becomes a 3×3 matrix:

$$\begin{pmatrix} \frac{1}{1+\Delta t k^p_{i,j}} & -\frac{c^2\Delta t i\xi_x}{1+\Delta t k^p_{i,j}} & -\frac{c^2\Delta t i\xi_y}{1+\Delta t k^p_{i,j}} \\ -\frac{\Delta t i\xi_x}{1+\Delta t k^o_{i,j}} & \frac{1}{1+\Delta t k^o_{i,j}} & 0 \\ -\frac{\Delta t i\xi_y}{1+\Delta t k^o_{i,j}} & 0 & \frac{1}{1+\Delta t k^o_{i,j}} \end{pmatrix}. \tag{14}$$

This structure can be expanded into a coupling matrix of size 3 × grid_size × 3 × grid_size, capturing the spatio-temporal coupling weights among all neurons. Then, we solve the characteristic equation $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$ to find the eigenvalues λ. By applying a cubic equation, we can get the following simplified equation:

$$\left(\frac{1}{1 + \Delta t k^o_{i,j}} - \lambda\right)\left[\left(\frac{1}{1 + \Delta t k^p_{i,j}} - \lambda\right)\left(\frac{1}{1 + \Delta t k^o_{i,j}} - \lambda\right) + \frac{c^2_{i,j}\Delta t^2(\xi_x^2 + \xi_y^2)}{(1 + \Delta t k^p_{i,j})(1 + \Delta t k^o_{i,j})}\right] = 0 \tag{15}$$

From the factored characteristic equation, we can identify two distinct eigenvalue solutions: a real eigenvalue corresponding to one component of the velocity field, $\lambda_1 = \frac{1}{1+\Delta t k^o_{i,j}}$, and the other two eigenvalues are the solutions to the right part quadratic term.

Applying the quadratic formula under the assumption that the damping coefficients are approximately equal ($k^p_{i,j} \approx k^o_{i,j}$) and the spatial frequencies are sufficiently large, the discriminant becomes negative. Consequently, the eigenvalues form a pair of complex conjugates:

$$\lambda_{2,3} \approx \frac{1}{2}\left(\frac{1}{1 + \Delta t k^p_{i,j}} + \frac{1}{1 + \Delta t k^o_{i,j}}\right) \pm i\frac{c_{i,j}\Delta t\sqrt{\xi_x^2 + \xi_y^2}}{\sqrt{(1 + \Delta t k^p_{i,j})(1 + \Delta t k^o_{i,j})}}. \tag{16}$$

Given the availability of local eigenvalues in BioOSS, the corresponding eigenvectors can also be computed. The complete eigenvector matrix is then assembled as follows (details of the derivation are provided in the Appendix Section B):

$$\mathbf{P} = \begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & \dfrac{\Delta t \cdot i\xi_x}{(1+\Delta t\, k^o)(\lambda_2 - \lambda_1)} & \dfrac{-\Delta t \cdot i\xi_x}{(1+\Delta t\, k^o)(\lambda_3 - \lambda_1)} \\ 0 & \dfrac{\Delta t \cdot i\xi_y}{(1+\Delta t\, k^o)(\lambda_2 - \lambda_1)} & \dfrac{-\Delta t \cdot i\xi_y}{(1+\Delta t\, k^o)(\lambda_3 - \lambda_1)} \end{bmatrix} \in \mathbb{C}^{3\times 3}.$$

Leveraging the scan property and the eigendecomposition of the system matrix, we express the update operator as $\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$, where $\mathbf{P}$ contains the eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues. Notably, the inverse $\mathbf{P}^{-1}$ is computed only once per sequence and backpropagation iteration during training, thereby reducing the computational overhead in iterative evaluations.

3.4 Stability Analysis

The system exhibits a recurrent dynamic structure similar to that in linear state-space models. Specifically, given the eigendecomposition of the state transition matrix $\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$, the recurrence takes the form

$$\mathbf{x}_n = \mathbf{A}\mathbf{x}_{n-1} + \mathbf{B}\mathbf{u}_n = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}\mathbf{x}_{n-1} + \mathbf{B}\mathbf{u}_n, \tag{17}$$

where $\mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$ denotes the eigendecomposition. Unrolling this recursion yields the explicit solution

$$\mathbf{x}_n = \sum_{k=0}^{n-1} \mathbf{A}^k \mathbf{B}\mathbf{u}_{n-k} = \sum_{k=0}^{n-1} \mathbf{P}\mathbf{\Lambda}^k \mathbf{P}^{-1}\mathbf{B}\mathbf{u}_{n-k}. \tag{18}$$

This standard eigenspace expansion of linear recurrences has also been used in oscillatory state-space models, such as LinOSS [Rusch and Rus, 2025], which focus on second-order ODE dynamics. Here, we adopt a similar formulation but extend it to PDE-driven spatio-temporal dynamics in BioOSS.

To ensure stability, all eigenvalues must satisfy $|\lambda| \leq 1$. We begin by observing that the real eigenvalue $\lambda_1 = \frac{1}{1+\Delta t k^o_{i,j}} < 1$ is strictly stable for any positive damping coefficient $k^o_{i,j} > 0$.

For the complex conjugate eigenvalue pair $\lambda_{2,3}$, we compute their squared magnitude:

$$|\lambda_{2,3}|^2 = \left(\frac{1}{2}\left(\frac{1}{1+\Delta t k^p_{i,j}} + \frac{1}{1+\Delta t k^o_{i,j}}\right)\right)^2 + \frac{c^2_{i,j}\Delta t^2(\xi_x^2 + \xi_y^2)}{(1+\Delta t k^p_{i,j})(1+\Delta t k^o_{i,j})}. \tag{19}$$

When both damping coefficients are positive ($k^p_{i,j} > 0$ and $k^o_{i,j} > 0$), the first term remains less than 1, while the second term reflects the contribution of wave propagation.

To ensure unconditional stability over all spatial frequencies (ξ_x, ξ_y), the wave speed must satisfy the following constraint, which generalizes the classic CFL (Courant–Friedrichs–Lewy) condition by incorporating the effects of damping parameters into the stability criterion:

$$c_{i,j} \leq \frac{\Delta x}{\Delta t} \cdot \frac{\sqrt{(1+\Delta t k^p_{i,j})(1+\Delta t k^o_{i,j})}}{\sqrt{2}}, \tag{20}$$

where Δx denotes fixed spatial step.

3.5 Eigenfrequency Structure and Emergent Selectivity

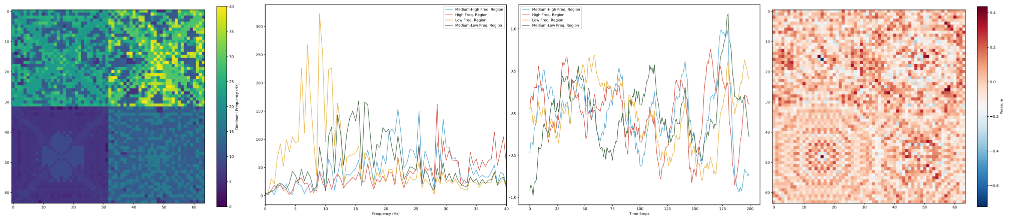
The eigenvalue structure of the BioOSS system provides insight into its emergent oscillatory behavior across the spatial grid. Each pair of complex conjugate eigenvalues corresponds to a spatio-temporal mode governed by the local wave parameters and spatial frequency components. While these modes are not tied to individual neurons, their frequency profiles shape the collective dynamics of the p-field. Although eigenvalues are not explicitly computed or used during training or inference, analyzing them post hoc reveals the types of oscillations the model can support after learning.

In discrete-time systems, complex eigenvalues $\lambda = a + ib$ are typically expressed in polar form as $\lambda = re^{i\theta}$, where $r = \sqrt{a^2 + b^2}$ denotes the magnitude and $\theta = \tan^{-1}(b/a)$ is the phase angle. This angle is directly related to the angular frequency of oscillation. As complex eigenvalues arise in conjugate pairs, a half revolution (π) around the unit circle corresponds to the system's sampling frequency $\frac{1}{\Delta t}$, allowing us to map the eigenvalue phase to a temporal frequency as $f = \frac{\theta}{\pi \Delta t}$.

For the dominant eigenvalues $\lambda_{2,3}$ derived from Eq. (16), this yields the following closed-form expression for the local oscillation frequency at each grid point (i, j):

$$f_{i,j} = \frac{1}{\pi \Delta t} \tan^{-1} \left(\frac{2 c_{i,j} \Delta t \sqrt{\xi_x^2 + \xi_y^2}}{\sqrt{(1 + \Delta t k^p_{i,j})(1 + \Delta t k^o_{i,j})} \left(\frac{1}{1+\Delta t k^p_{i,j}} + \frac{1}{1+\Delta t k^o_{i,j}} \right)} \right). \tag{21}$$

This frequency mapping links each spatio-temporal mode's oscillatory profile to the local parameters c, k^p, k^o and the wave vector (ξ_x, ξ_y). To empirically validate this interpretation, we assess whether neurons initialized with distinct eigenmodes exhibit the expected frequency selectivity.



(a) (b) (c) (d)

Figure 2: (a) Dominant-frequency map; (b) Spectra at region centers; (c) Time traces (last 200 steps); (d) Final p-field. Quadrants are initialized for 0–10, 10–20, 20–30, and 30–40 Hz and each resonates in its assigned band despite identical white-noise input, supporting the eigenvalue–frequency mapping. Full spectra and time traces are provided in Appendix E.

We partition a 64×64 grid into four quadrants and initialize each with eigenmodes targeting a different frequency band (0–10, 10–20, 20–30, 30–40 Hz), using Eq. (21). While Eq. (21) defines a mapping from eigenvalue phase to frequency using Δt, this interpretation is only meaningful when Δt corresponds to a physical time unit. In our synthetic experiments (e.g., white-noise input with frequency-limited bands), we set $\Delta t = 10\,\text{ms}$ to reflect biologically plausible temporal resolution, allowing the resulting frequencies to be interpreted in Hz. In contrast, for downstream tasks such as those in Section 4.2, Δt is treated as a unitless index (e.g., one step = one hour in the Traffic dataset), and frequency should be understood as a relative, dimensionless measure rather than a physical quantity. All regions receive the same white-noise input. Despite identical stimulation, each quadrant exhibits a dominant-frequency profile matching its assigned band (Fig. 2). This supports the hypothesis that the eigenvalue phase determines local spectral responses, and demonstrates BioOSS's capacity for encoding frequency preferences through spatially structured initialization.

These results offer a mechanistic account of how spatio-temporal selectivity arises in BioOSS, demonstrating that distinct frequency profiles can emerge even without anatomically imposed groupings. The resulting frequency-selective regions resemble functional "islands" with coherent internal dynamics and spectral specialization. While BioOSS does not currently enforce explicit connectivity boundaries, similar modular topologies—defined by dense intra-cluster and sparse inter-cluster connections—have been shown to support self-sustained, wiring-efficient dynamics in both biological and artificial systems Meunier et al. [2010], Liang et al. [2022].

3.6 Computational Complexity

The computational complexity of BioOSS scales linearly with the sequence length T, spatial size HW, and number of layers L. Specifically, the inference cost is $\mathcal{O}(LTHWd)$, and the training cost is $\mathcal{O}(LTHW(d + d^2))$, where $d = HW$ denotes the dimension of the hidden state. A detailed derivation of these results is provided in Appendix C.

4 Results

4.1 Experimental Setting

Baseline This paper primarily focuses on the exploration of oscillatory-based spatio-temporal model and sequential tasks. Therefore, we do not include Transformer models, as they are end-to-end frameworks. BioOSS is evaluated against several representative sequential models with recurrent mechanisms, including the Linear Recurrent Unit (LRU) [Orvieto et al., 2023], S5 [Smith et al., 2022], and LinOSS [Rusch and Rus, 2025]. While alternative sequential modeling approaches, such as the Neural Refined Differential Equation (NRDE) [Irie et al., 2022] and the Neural Controlled Differential Equation (NCDE) [Kidger et al., 2020], are available, these require solving differential equations at each time step, leading to significant computational and time overhead. As such, we exclude these models from our comparison to ensure a fair and efficient benchmarking protocol.

Nonlinear Components The core BioOSS system is linear, but practical implementations include nonlinear components such as multilayer readouts and self-gating blocks in the model head [Rusch and Rus, 2025]. These additions are independent of the oscillatory dynamics and are described in Appendix D.

Hyperparameters and Environment All BioOSS models employed in the experiments adopt a 2D architecture, with the model size varying across tasks. For each task, the learning rate was selected through a grid search to ensure optimal performance. A comprehensive summary of all hyperparameter configurations is provided in the Appendix Section D Table 4. All experiments were conducted using JAX for classification tasks and PyTorch for prediction tasks to ensure consistency with established baselines, datasets, and evaluation protocols. Models were trained using the Adam optimizer and executed on a single NVIDIA GeForce RTX 4090 GPU (24 GB memory). Details on model parameters, GPU usage, and memory consumption are provided in Appendix D, Tables 5 and 6.

4.2 Evaluation on Time Series Tasks

The benchmark datasets considered in this section use time steps that do not always correspond to fixed physical durations. For example, in the Traffic dataset, each step represents one hour. As a result, while BioOSS captures oscillatory dynamics across time steps, no mapping to physical frequency units (e.g., Hz) is applied in these experiments. We evaluate the proposed BioOSS model on two fundamental time series tasks: classification and prediction. Experiments are conducted on recent benchmark datasets to demonstrate the effectiveness of the bio-inspired architecture.

All models were trained under identical settings, with results averaged over five seeds and a 70:15:15 train/validation/test split. For BioOSS, we tuned the learning rate, while for LinOSS and other baselines we relied on the official implementations [Rusch and Rus, 2025] without grid search. We note that our reproduced LinOSS results differ slightly from those reported in the LinOSS paper, which we attribute to environment dependencies, lack of hyperparameter retuning, and the strong sensitivity of UEA and PPG datasets to the exact random split across Python/JAX environments. Our aim was not to re-evaluate LinOSS, but to benchmark all models under a unified protocol, presenting BioOSS as a competitive and interpretable alternative rather than a new state of the art.

4.2.1 Classification

We evaluate the proposed BioOSS model on a recently introduced sequential benchmark presented in [Walker et al., 2024], using the same benchmark protocol as in the LinOSS paper [Rusch and Rus, 2025] to ensure fair comparison and evaluation. This benchmark includes six multivariate time series datasets from the University of East Anglia (UEA) Multivariate Time Series Classification Archive (UEA-MTSCA) [Bagnall et al., 2018]: EigenWorms (Worms), SelfRegulationSCP1 (SCP1), SelfRegulationSCP2 (SCP2), EthanolConcentration (Ethanol), Heartbeat, and MotorImagery (Motor). These datasets exhibit a wide range of sequence lengths and classification complexities, providing a comprehensive evaluation of model performance under varying temporal and structural characteristics. The results are summarized in Table 1.

	Worms	SCP1	SCP2	Ethanol	Heartbeat	Motor
Seq. length	17,984	896	1,152	1,751	405	3,000
#Classes	5	2	2	4	2	2
LRU	89.4 ± 5.4	86.8 ± 3.1	54.7 ± 10.1	28.4 ± 5.0	70.0 ± 3.6	55.8 ± 6.2
S5	83.9 ± 4.1	**86.8 ± 4.5**	51.9 ± 4.5	24.3 ± 4.8	73.9 ± 3.1	50.5 ± 4.2
LinOSS	92.2 ± 7.7	85.9 ± 2.9	**60.4 ± 5.7**	29.9 ± 0.6	71.9 ± 2.6	**60.4 ± 7.2**
BioOSS	**92.8 ± 5.2**	85.6 ± 3.9	55.1 ± 1.8	**33.4 ± 10.7**	**74.8 ± 2.0**	55.8 ± 5.8

Table 1: **Multivariate long-term time series classification results.** Accuracy (%) and standard deviation for each model on six UEA datasets. Best results are shown in **red and bold**; second-best results are in blue and underlined. Higher is better.

We report the multivariate time series classification performance of all models across six benchmark UEA datasets in Table 1. On average, BioOSS achieves the highest accuracy at 66.25%, outperforming all baselines, followed by LinOSS at 65.12%. BioOSS achieves state-of-the-art results on three datasets (Worms, EthanolConcentration, and Heartbeat) and secures the second-best performance on two others (SCP2 and MotorImagery).

Notably, on the challenging Heartbeat dataset, BioOSS attains the best accuracy of 74.8%, improving upon LinOSS and S5 by 2.9% and 0.9%, respectively. On the long-sequence EthanolConcentration dataset (1,751 steps), BioOSS also outperforms all baselines, achieving 33.4% accuracy compared to 29.9% for LinOSS and 28.4% for LRU.

In contrast, S5 shows competitive performance only on SCP1 and Heartbeat, while LRU performs best only on SCP1 and exhibits high variance across datasets. LinOSS demonstrates consistent robustness, achieving the highest accuracy on SCP2 (60.4%) and the second-best result on Worms (92.2%). These results highlight the strong generalization ability of BioOSS, particularly for long sequences and heterogeneous biological and environmental signals.

4.2.2 Prediction

We evaluate the proposed models on the PPG-DaLiA dataset, a benchmark for heart rate regression using wearable sensor data. The dataset contains multi-channel physiological and motion signals collected from fifteen subjects engaged in daily activities. Each subject's recordings span approximately 2.5 hours at 128 Hz, covering six input channels: blood volume pulse, electrodermal activity, body temperature, and triaxial acceleration. To handle the long-range dependencies, we segment the data using a sliding window of length 49,920 and a step size of 4,992. Following the same evaluation protocol as in the LinOSS paper [Rusch and Rus, 2025], all models are trained with identical hyperparameter tuning procedures to ensure fair comparison and evaluation. Table 2 summarizes the average test mean-squared error (MSE) over five independent runs. LinOSS achieves the lowest MSE, while BioOSS secures the second-best performance. Compared to traditional recurrent and state-space baselines such as LRU and S5, both LinOSS and BioOSS demonstrate clear advantages in modeling long-duration dependencies, highlighting the effectiveness of the proposed structured designs for physiological signals.

Model	MSE $\times 10^{-2}$
LRU	15.64 ± 1.67
S5	12.79 ± 0.72
LinOSS	**6.4 ± 0.23**
BioOSS	7.7 ± 0.2

Table 2: **PPG-based regression results on the PPG-DaLiA dataset.** Mean-squared error ($\times 10^{-2}$) and standard deviation over 5 training runs for each model. All models are trained using the same hyper-parameter tuning protocol to ensure fair comparison. Best results are shown in **red and bold** and second-best results are in blue and underlined. Lower is better.

To further evaluate predictive capabilities, we assess the models on four additional benchmark datasets characterized by clear periodic patterns: Electricity, Solar-Energy, Traffic, and Weather [Lai et al., 2018]. Dataset splits, look-back sequence lengths, and prediction horizons are aligned with the cyclic properties of each dataset. The results, summarized in Table 3, further validate the strong performance of LinOSS and BioOSS across diverse long-term forecasting tasks.

	Electricity	Solar-Energy	Traffic	Weather
Timesteps	26,304	52,560	17,544	52,696
Channels	321	137	862	21
Frequency	1 hour	10 mins	1 hour	10 mins
Cyclic Patterns	Daily & Weekly	Daily	Daily & Weekly	Daily
Cycle Length	168	144	168	144
LinOSS	21.875 ± 0.000	22.363 ± 0.000	68.828 ± 0.002	22.288 ± 0.000
BioOSS	**18.699 ± 0.000**	**21.873 ± 0.003**	**62.284 ± 0.007**	**22.232 ± 0.000**

Table 3: **Multivariate time-series forecasting results across four benchmark datasets with intrinsic patterns.** Mean-squared error (MSE $\times 10^{-2}$) and standard deviation over 5 training runs are reported for each model. All models are trained using the same hyper-parameter tuning protocol to ensure a fair comparison. Best results for each dataset are highlighted in **red and bold**. Lower is better.

Table 3 reports the average test mean-squared error (MSE) across five independent runs for LinOSS and BioOSS on four benchmark multivariate time series datasets with inherent temporal patterns. BioOSS consistently outperforms LinOSS, achieving the lowest MSEs of 18.699, 21.873, 62.284, and 22.232 on the Electricity, Solar-Energy, Traffic, and Weather datasets, respectively. In contrast, LinOSS yields MSEs of 21.875, 22.363, 68.828, and 22.288 on the same datasets. These results highlight the superior ability of BioOSS to capture temporal dependencies, particularly in datasets exhibiting daily and weekly cyclic patterns. The oscillatory structure of BioOSS enables more effective modeling of periodic dynamics, even with a comparable number of neurons to LinOSS.

4.3 Wave Dynamics Visualization

To qualitatively examine the internal dynamics of BioOSS beyond benchmark accuracy, we generated visualizations of emergent wave behavior in the 2D oscillatory layer. Specifically, normalized logits were extracted from an audio tagging backbone processing a 1-minute audio segment using a 4-second sliding window with a 1-second stride. These logits were then used as point-wise inputs to BioOSS, with each logit dimension mapped to a local patch of 7–8 grid cells. The system subsequently evolved under BioOSS dynamics, and the pressure-field (p) values were recorded over time. The resulting sequences reveal traveling-wave propagation and coupling effects: activity initiated in one region not only spreads across its assigned area but also induces oscillatory responses in neighboring regions through local interactions. This dynamic behavior complements the static examples shown in Fig. 1(d–e), providing direct evidence of emergent spatiotemporal coordination. Animated GIFs of representative sequences are provided in the Supplementary Material, which contains three audio samples illustrating the evolution of the p-field under realistic inputs.

5 Conclusion

We proposed BioOSS, a biologically inspired oscillatory state system that captures spatio-temporal dynamics through wave-like propagation between pressure and velocity-like neurons on a 2D grid. By introducing trainable damping and propagation parameters, BioOSS adapts to task-specific patterns while preserving numerical stability and interpretability. It achieves strong performance across classification and forecasting benchmarks, particularly excelling in frequency-selective and long-range temporal tasks. Unlike traditional models, BioOSS offers insight into its internal dynamics via eigenvalue-based frequency decomposition. While promising, future work can explore scaling, multimodal integration, and continuous learning. BioOSS offers a step toward interpretable, brain-inspired computation in AI. A detailed discussion of limitations is included in Appendix F.

Acknowledgment

This work was supported by the Special Research Fund (BOF) of Ghent University under Grant BOF/24J/2021/246, and by the Flemish Government through the Onderzoeksprogramma AI Vlaanderen programme. We gratefully acknowledge the valuable effort of Prof. Eamonn Keogh and his team in creating, curating, and sharing the datasets used in this study.

References

Pepe Alcamí and Alberto E Pereda. Beyond plasticity: the dynamic impact of electrical synapses on neural circuits. *Nature Reviews Neuroscience*, 20(5):253–271, 2019.

Marie Auger-Méthé, Ken Newman, Diana Cole, Fanny Empacher, Rowenna Gryba, Aaron A King, Vianey Leos-Barajas, Joanna Mills Flemming, Anders Nielsen, Giovanni Petris, et al. A guide to state–space modeling of ecological time series. *Ecological Monographs*, 91(4):e01470, 2021.

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. *arXiv preprint arXiv:1811.00075*, 2018.

Johannes Balkenhol, Barbara Händel, Sounak Biswas, Johannes Grohmann, Jóakim v Kistowski, Juan Prada, Conrado A Bosman, Hannelore Ehrenreich, Sonja M Wojcik, Samuel Kounev, et al. Beyond-local neural information processing in neuronal networks. *Computational and Structural Biotechnology Journal*, 23:4288–4305, 2024.

Dean V Buonomano and Wolfgang Maass. State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, 10(2):113–125, 2009.

Luke Campagnola, Stephanie C Seeman, Thomas Chartrand, Lisa Kim, Alex Hoggarth, Clare Gamlin, Shinya Ito, Jessica Trinh, Pasha Davoudian, Cristina Radaelli, et al. Local connectivity and synaptic dynamics in mouse and human neocortex. *Science*, 375(6585):eabj5861, 2022.

Anders M Dale and Eric Halgren. Spatiotemporal mapping of brain activity by integration of multiple imaging modalities. *Current opinion in neurobiology*, 11(2):202–208, 2001.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Keith B Doelling and M Florencia Assaneo. Neural oscillations are a start toward understanding brain activity rather than the end. *PLoS biology*, 19(5):e3001234, 2021.

Moein Esghaei, Stefan Treue, and Trichur R Vidyasagar. Dynamic coupling of oscillatory neural activity and its roles in visual attention. *Trends in Neurosciences*, 45(4):323–335, 2022.

Maya Foster and Dustin Scheinost. Brain states as wave-like motifs. *Trends in Cognitive Sciences*, 2024.

Joaquín M Fuster and Steven L Bressler. Cognit activation: a mechanism enabling temporal integration in working memory. *Trends in cognitive sciences*, 16(4):207–218, 2012.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Tyler W Hughes, Ian AD Williamson, Momchil Minkov, and Shanhui Fan. Wave physics as an analog recurrent neural network. *Science advances*, 5(12):eaay6946, 2019.

Kazuki Irie, Francesco Faccio, and Jürgen Schmidhuber. Neural differential equations for learning to program neural nets through continuous learning rules. *Advances in Neural Information Processing Systems*, 35:38614–38628, 2022.

T Anderson Keller and Max Welling. Neural wave machines: learning spatiotemporally structured representations with locally coupled oscillatory recurrent neural networks. In *International Conference on Machine Learning*, pages 16168–16189. PMLR, 2023.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33:6696–6707, 2020.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

Samuel Lanthaler, T Konstantin Rusch, and Siddhartha Mishra. Neural oscillators are universal. *Advances in Neural Information Processing Systems*, 36:46786–46806, 2023.

Junhao Liang, Sheng-Jun Wang, and Changsong Zhou. Less is more: wiring-economical modular networks support self-sustained firing-economical neural avalanches for efficient processing. *National Science Review*, 9(3):nwab102, 2022.

Ivan S Maksymov. Analogue and physical reservoir computing using water waves: Applications in power engineering and beyond. *Energies*, 16(14):5366, 2023.

David Meunier, Renaud Lambiotte, and Edward T Bullmore. Modular and hierarchically modular organization of brain networks. *Frontiers in neuroscience*, 4:200, 2010.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pages 26670–26698. PMLR, 2023.

Blake A Richards and Paul W Frankland. The persistence and transience of memory. *Neuron*, 94(6): 1071–1084, 2017.

Mattia Rigotti, Omri Barak, Melissa R Warden, Xiao-Jing Wang, Nathaniel D Daw, Earl K Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497 (7451):585–590, 2013.

T Konstantin Rusch and Siddhartha Mishra. Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies. In *International Conference on Learning Representations (ICLR 2021)*. OpenReview, 2021.

T. Konstantin Rusch and Daniela Rus. Oscillatory state-space models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=GRMfXcAAFh`.

Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022.

José F Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: a survey. *Big data*, 9(1):3–21, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Benjamin Walker, Andrew Donald Mcleod, Tiexin Qin, Yichuan Cheng, Haoliang Li, and Terry Lyons. Log neural controlled differential equations: The lie brackets make a difference. In *International Conference on Machine Learning*, pages 49822–49844. PMLR, 2024.

Akiya Watakabe, Henrik Skibbe, Ken Nakae, Hiroshi Abe, Noritaka Ichinohe, Muhammad Febrian Rachmadi, Jian Wang, Masafumi Takaji, Hiroaki Mizukami, Alexander Woodward, et al. Local and long-distance organization of prefrontal cortex circuits in the marmoset brain. *Neuron*, 111 (14):2258–2273, 2023.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: a survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 6778–6786, 2023.

Jie Xia, Cirong Liu, Jiao Li, Yao Meng, Siqi Yang, Huafu Chen, and Wei Liao. Decomposing cortical activity through neuronal tracing connectome-eigenmodes in marmosets. *Nature Communications*, 15(1):2289, 2024.

A Fourier Transformation of Differential Operators

In this section, we derive the fundamental relationship between differential operators in physical space and their counterparts in Fourier space.

A.1 Definitions and Notation

We begin with the definition of the Fourier transform for a function $p(\mathbf{r})$ in $\mathbb{R}^n$:

$$\tilde{p}(\boldsymbol{\xi}) = \mathcal{F}\{p(\mathbf{r})\} = \int_{\mathbb{R}^n} p(\mathbf{r}) e^{-i\boldsymbol{\xi}\cdot\mathbf{r}}\, d\mathbf{r}, \tag{22}$$

where $\boldsymbol{\xi} = [\xi_x, \xi_y, \ldots]^\top$ is the frequency domain vector.

For simplicity, we focus on the two-dimensional case where $\mathbf{r} = [x, y]^\top$ and $\boldsymbol{\xi} = [\xi_x, \xi_y]^\top$.

A.2 Transformation of the Gradient Operator

We aim to derive the Fourier transform of the gradient of p, denoted by ∇p:

$$\mathcal{F}\{\nabla p\} = \mathcal{F}\left\{ \begin{bmatrix} \frac{\partial p}{\partial x} \\ \frac{\partial p}{\partial y} \end{bmatrix} \right\}. \tag{23}$$

We first evaluate the Fourier transform of the spatial derivative of p, given by $\mathcal{F}\left\{\frac{\partial p}{\partial x}\right\}$:

$$\mathcal{F}\left\{\frac{\partial p}{\partial x}\right\} = \int_{\mathbb{R}^2} \frac{\partial p(x,y)}{\partial x} e^{-i(\xi_x x + \xi_y y)}\, dx\, dy. \tag{24}$$

Focusing on the integration with respect to x, we apply integration by parts:

$$\begin{aligned} \int_{-\infty}^{\infty} \frac{\partial p(x,y)}{\partial x} e^{-i\xi_x x}\, dx &= \left[p(x,y)e^{-i\xi_x x}\right]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} p(x,y) \frac{\partial}{\partial x}\left(e^{-i\xi_x x}\right)\, dx \\ &= \left[p(x,y)e^{-i\xi_x x}\right]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} p(x,y)\left(-i\xi_x e^{-i\xi_x x}\right)\, dx \\ &= \left[p(x,y)e^{-i\xi_x x}\right]_{-\infty}^{\infty} + i\xi_x \int_{-\infty}^{\infty} p(x,y) e^{-i\xi_x x}\, dx. \end{aligned} \tag{25}$$

For functions that are sufficiently smooth and decay rapidly as $|x| \to \infty$, the boundary term vanishes:

$$\left[p(x,y)e^{-i\xi_x x}\right]_{-\infty}^{\infty} = 0. \tag{26}$$

Therefore:

$$\int_{-\infty}^{\infty} \frac{\partial p(x,y)}{\partial x} e^{-i\xi_x x}\, dx = i\xi_x \int_{-\infty}^{\infty} p(x,y) e^{-i\xi_x x}\, dx. \tag{27}$$

Reintroducing the integration with respect to y:

$$\begin{aligned} \mathcal{F}\left\{\frac{\partial p}{\partial x}\right\} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\partial p(x,y)}{\partial x} e^{-i(\xi_x x + \xi_y y)}\, dx\, dy \\ &= \int_{-\infty}^{\infty} \left[i\xi_x \int_{-\infty}^{\infty} p(x,y) e^{-i\xi_x x}\, dx \right] e^{-i\xi_y y}\, dy \\ &= i\xi_x \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y) e^{-i(\xi_x x + \xi_y y)}\, dx\, dy \\ &= i\xi_x \tilde{p}(\xi_x, \xi_y). \end{aligned} \tag{28}$$

By symmetry, we can similarly derive:

$$\mathcal{F}\left\{\frac{\partial p}{\partial y}\right\} = i\xi_y \tilde{p}(\xi_x, \xi_y). \tag{29}$$

Thus, for the gradient operator, we have:

$$\mathcal{F}\{\nabla p\} = \mathcal{F}\left\{\begin{bmatrix}\frac{\partial p}{\partial x}\\ \frac{\partial p}{\partial y}\end{bmatrix}\right\} = \begin{bmatrix} i\xi_x \tilde{p} \\ i\xi_y \tilde{p}\end{bmatrix} = i\boldsymbol{\xi}\tilde{p}. \tag{30}$$

This gives the important relationship:

$$\nabla p \xrightarrow{\mathcal{F}} i\boldsymbol{\xi}\tilde{p}. \tag{31}$$

A.3 Transformation of the Divergence Operator

For a vector field $\mathbf{o} = [o_x, o_y]^\top$, the divergence operator is defined as follows:

$$\nabla \cdot \mathbf{o} = \frac{\partial o_x}{\partial x} + \frac{\partial o_y}{\partial y}. \tag{32}$$

Applying the Fourier transform and using the result from the previous section:

$$\begin{aligned}\mathcal{F}\{\nabla \cdot \mathbf{o}\} &= \mathcal{F}\left\{\frac{\partial o_x}{\partial x} + \frac{\partial o_y}{\partial y}\right\} \\ &= \mathcal{F}\left\{\frac{\partial o_x}{\partial x}\right\} + \mathcal{F}\left\{\frac{\partial o_y}{\partial y}\right\} \\ &= i\xi_x \tilde{o}_x + i\xi_y \tilde{o}_y \\ &= i\boldsymbol{\xi}\cdot\tilde{\mathbf{o}}.\end{aligned} \tag{33}$$

This gives the important relationship:

$$\nabla \cdot \mathbf{o} \xrightarrow{\mathcal{F}} i\boldsymbol{\xi}\cdot\tilde{\mathbf{o}}. \tag{34}$$

Thus, the Fourier transform converts differential operators into algebraic operations:

$$\frac{\partial}{\partial t} \xrightarrow{\mathcal{F}} -i\omega, \tag{35}$$

$$\nabla p \xrightarrow{\mathcal{F}} i\boldsymbol{\xi}\tilde{p}, \tag{36}$$

$$\nabla \cdot \mathbf{o} \xrightarrow{\mathcal{F}} i\boldsymbol{\xi}\cdot\tilde{\mathbf{o}}. \tag{37}$$

B Analytical Form of Eigenvectors P

We consider the linearized local system matrix at a grid point (i, j) in Fourier space:

$$A = \begin{bmatrix} \alpha & -\alpha c^2 \Delta t\, i\xi_x & -\alpha c^2 \Delta t\, i\xi_y \\ -\beta\Delta t\, i\xi_x & \beta & 0 \\ -\beta\Delta t\, i\xi_y & 0 & \beta \end{bmatrix}, \quad \alpha = \frac{1}{1+\Delta t\, k^p}, \quad \beta = \frac{1}{1+\Delta t\, k^o}.$$

Let λ be an eigenvalue of A, and $\mathbf{v} = [v_1, v_2, v_3]^T$ its associated eigenvector. We assume without loss of generality $v_1 = 1$ and solve $(A - \lambda I)\mathbf{v} = 0$. From the second and third rows, we obtain

$$v_2 = \frac{\beta\Delta t\, i\xi_x}{\lambda - \beta}, \quad v_3 = \frac{\beta\Delta t\, i\xi_y}{\lambda - \beta}.$$

For eigenvalues λ_2 and λ_3 corresponding to oscillatory modes, the associated eigenvectors are

$$\mathbf{v}_2 = \begin{bmatrix} 1 \\ \dfrac{\Delta t \cdot i\xi_x}{(1+\Delta t\, k^o)(\lambda_2 - \lambda_1)} \\ \dfrac{\Delta t \cdot i\xi_y}{(1+\Delta t\, k^o)(\lambda_2 - \lambda_1)} \end{bmatrix}, \quad \mathbf{v}_3 = \begin{bmatrix} 1 \\ \dfrac{-\Delta t \cdot i\xi_x}{(1+\Delta t\, k^o)(\lambda_3 - \lambda_1)} \\ \dfrac{-\Delta t \cdot i\xi_y}{(1+\Delta t\, k^o)(\lambda_3 - \lambda_1)} \end{bmatrix}.$$

In contrast, for the purely dissipative eigenvalue $\lambda_1 = \beta$, the eigenvectors correspond to vectors in the nullspace of the first row. A basis can be chosen, for example, as

$$\mathbf{v}_1 = \begin{bmatrix}0\\1\\0\end{bmatrix}, \quad \text{or} \quad \begin{bmatrix}0\\0\\1\end{bmatrix}.$$

C Computational Complexity Analysis

The computational cost of BioOSS primarily arises from its spatio-temporal update in the hidden layer, which is based on finite-difference approximations of a coupled PDE system. We characterize training and inference complexity as follows.

Let $p \in \mathbb{R}^{H \times W}$ and $o = (o^x, o^y) \in \mathbb{R}^{2 \times H \times W}$ denote the pressure- and velocity-like states defined on a 2D grid. At each time step, a BioOSS layer performs:

- Gradient update:
$$\nabla p_{i,j} = \left(\frac{p_{i,j} - p_{i-1,j}}{\Delta x}, \frac{p_{i,j} - p_{i,j-1}}{\Delta x} \right)$$
- Divergence update:
$$\nabla \cdot o_{i,j} = \frac{o^x_{i+1,j} - o^x_{i,j}}{\Delta x} + \frac{o^y_{i,j+1} - o^y_{i,j}}{\Delta x}$$
- Damped update:
$$o^{(l)} = \frac{o^*}{1 + \Delta t \cdot k^o}, \quad p^{(l)} = \frac{p^*}{1 + \Delta t \cdot k^p}$$

These operations require $\mathcal{O}(HW)$ computations per layer per time step, where $H \times W$ is the spatial resolution of the 2D state. Each layer also performs linear projections, gating, and readout on a hidden state of dimension $d = HW$, yielding a cost of $\mathcal{O}(HW \cdot d)$.

Thus, the total inference cost per layer over T time steps is:

$$\mathcal{O}(LTHWd).$$

During training, gradients must also be computed through the differentiable spatial operators and projection layers, with weight updates costing $\mathcal{O}(HW \cdot d^2)$ per layer. Therefore, the total training complexity is:

$$\mathcal{O}(LTHW(d + d^2)).$$

In practice, the hidden layer dominates both inference and training cost, and the model's complexity scales linearly with sequence length T, number of layers L, and spatial size HW.

D Training Details

Algorithm 1 illustrates the full architecture of the multi-layer BioOSS model. While the main text focuses on a single-layer formulation, the full model stacks L such layers sequentially. Each layer integrates two key components: a biologically inspired spatio-temporal propagation mechanism and a scan-compatible gated update.

Given an input sequence u, the hidden state $x^{(l)}$ is first updated by solving a PDE-based wave propagation equation over a 2D neural grid, as defined in Eq. (3). This involves local interactions between pressure-like p neurons and velocity-like o neurons through spatial gradients and divergence operators. These updates are governed by trainable wave speed c and damping coefficients k^p, k^o.

The resulting hidden state is then projected through a linear layer and passed to a gating mechanism, inspired by LinOSS Rusch and Rus [2025], but applied on a 2D spatio-temporal field. The gating update follows:

$$x^{(l)}_{\text{out}} = \sigma(W_g x^{(l)}) \odot \tanh(W_z x^{(l)}) + (1 - \sigma(W_g x^{(l)})) \odot x^{(l-1)},$$

where W_g and W_z are learnable matrices shared across time. This residual formulation promotes stable deep learning and supports efficient scan operations.

Next, the gated output is mapped via a linear readout: $y^{(l)} = Cx^{(l)}_{\text{out}} + Du^{(l-1)}$, followed by a GELU activation Hendrycks and Gimpel [2016]. A Gated Linear Unit (GLU) Dauphin et al. [2017] then modulates the activations via $\text{GLU}(x) = \sigma(W_1 x) \circ (W_2 x)$. The result is added to the previous layer input to produce $u^{(l)}$, which is fed into the next block.

Algorithm 1 Training Procedure for Full BioOSS Model with Spatio-Temporal Layer

```
Input: Input sequence $u$
Output: BioOSS output sequence $o$ after $L$ layers
for $l = 1$ to $L$ do
    // Step 1: Spatio-temporal PDE-based propagation over 2D grid
    Compute gradient: $\nabla p^{(l-1)} \leftarrow \text{SpatialGradient}(p^{(l-1)})$
    Compute divergence: $\nabla \cdot o^{(l-1)} \leftarrow \text{SpatialDivergence}(o^{(l-1)})$
    Update velocity-like neurons: $o^* \leftarrow o^{(l-1)} - \Delta t \cdot \nabla p^{(l-1)}$
    Update pressure-like neurons: $p^* \leftarrow p^{(l-1)} - \Delta t \cdot (c^2 \odot \nabla \cdot o^{(l-1)}) + \Delta t \cdot Bu^{(l-1)}$
    Apply damping: $o^{(l)} \leftarrow o^*/(1 + \Delta t \cdot k^o), \quad p^{(l)} \leftarrow p^*/(1 + \Delta t \cdot k^p)$
    Hidden state: $x^{(l)} \leftarrow \text{concat}(p^{(l)}, o^{(l)})$
    // Step 2: Linear projection and gated residual update (SSM-style)
    Linear projection: $z^{(l)} \leftarrow W_z x^{(l)}$
    Gating: $\text{gate}^{(l)} \leftarrow \sigma(W_g x^{(l)})$
    Residual gated update: $x_{\text{out}}^{(l)} \leftarrow \text{gate}^{(l)} \odot \tanh(z^{(l)}) + (1 - \text{gate}^{(l)}) \odot x^{(l-1)}$
    // Step 3: Readout and layer update
    Linear readout: $y^{(l)} \leftarrow C x_{\text{out}}^{(l)} + D u^{(l-1)}$
    Apply nonlinearity: $y^{(l)} \leftarrow \text{GELU}(y^{(l)})$
    Update input for next layer: $u^{(l)} \leftarrow \text{GLU}(y^{(l)}) + u^{(l-1)}$
end for
Return final output: $\mathbf{o} \leftarrow \mathbf{W}_{\text{out}} \mathbf{y}^{(L)} + \mathbf{b}_{\text{out}}$
```

After L layers, the final output is computed as $o = W_{\text{out}} y^{(L)} + b_{\text{out}}$. All operations are parallelized across time steps, and time indices are omitted from the notation for clarity.

The best-performing hyperparameter configurations for the BioOSS model across all datasets are summarized in Table 4. These configurations were obtained using the same search procedure described in the main paper (Section 4.2). Due to the high nonlinearity of the BioOSS model, its performance is highly sensitive to both the learning rate and the spatial grid resolution. Therefore, for each dataset, we conducted a systematic grid search, exploring learning rates from 0.000001 to 0.005, hidden dimensions of 16, 32, 64, 128, and 256, and spatial grid sizes of 10×10, 15×15, and 20×20. For each dataset, we report the selected learning rate, hidden dimension size, spatial grid resolution, and number of blocks. Notably, a single block was sufficient for all cases, while the optimal learning rate, hidden size, and grid size varied depending on the dataset characteristics.

Model	**lr**	**Hidden Dim**	**Grid Size**	**#layers**
Worms	0.0001	128	20×20	1
SCP1	0.0002	128	20×20	1
SCP2	0.00001	128	15×15	1
Ethanol	0.00025	16	20×20	1
Heartbeat	0.000005	16	20×20	1
Motor	0.000004	128	20×20	1
PPG	0.0002	128	20×20	1
Electricity	0.01	16	10×10	1
Solar-Energy	0.01	16	10×10	1
Traffic	0.01	16	10×10	1
Weather	0.01	16	10×10	1

Table 4: Best hyperparameter configurations for BioOSS.

We report the number of parameters, GPU memory usage (in MB), and run time (in seconds) for all evaluated models across the datasets introduced in Section 4.2. The experimental settings for the baseline models strictly follow those established in [Rusch and Rus, 2025], ensuring consistency in both codebase and hardware environment. All experiments were conducted on the same GPU device to guarantee fair and direct comparability. Table 5 summarizes the model configurations, memory consumption, and training run time for the time-series classification (TSC) and PPG forecasting tasks, where all experiments were implemented using JAX. In contrast, Table 6 presents the corresponding results for the long-term time-series forecasting (TSP) benchmarks, with experiments conducted in PyTorch. We note that differences in parallel computation mechanisms between JAX and PyTorch may account for minor variations in GPU memory usage. Overall, LinOSS and BioOSS exhibit competitive memory efficiency and favorable run time performance relative to established baselines such as LRU and S5.

Dataset	**Config**	**LRU**	**S5**	**LinOSS**	**BioOSS**
Worms	#parameters	101129	22007	134279	153549
	GPU memory (MB)	18761	18877	18747	18727
	run time (s)	138	78	11	15
SCP1	#parameters	25892	226328	991240	153162
	GPU memory (MB)	18759	18879	18893	18739
	run time (s)	15	4	34	6
SCP2	#parameters	26020	5652	448072	107002
	GPU memory (MB)	18781	18881	18853	18741
	run time (s)	10	3	46	4
Ethanol	#parameters	76522	76214	6728	16332
	GPU memory (MB)	18737	18915	18813	18733
	run time (s)	4	5	4	9
Heartbeat	#parameters	338820	158310	10936	16300
	GPU memory (MB)	18503	18885	18835	18741
	run time (s)	7	3	5	2
Motor	#parameters	107544	17496	91844	153162
	GPU memory (MB)	18753	18857	18769	18727
	run time (s)	47	26	8	17
PPG	#parameters	107544	17496	91844	169416
	GPU memory (MB)	18773	18915	18719	18729
	run time (s)	107	57	21	17

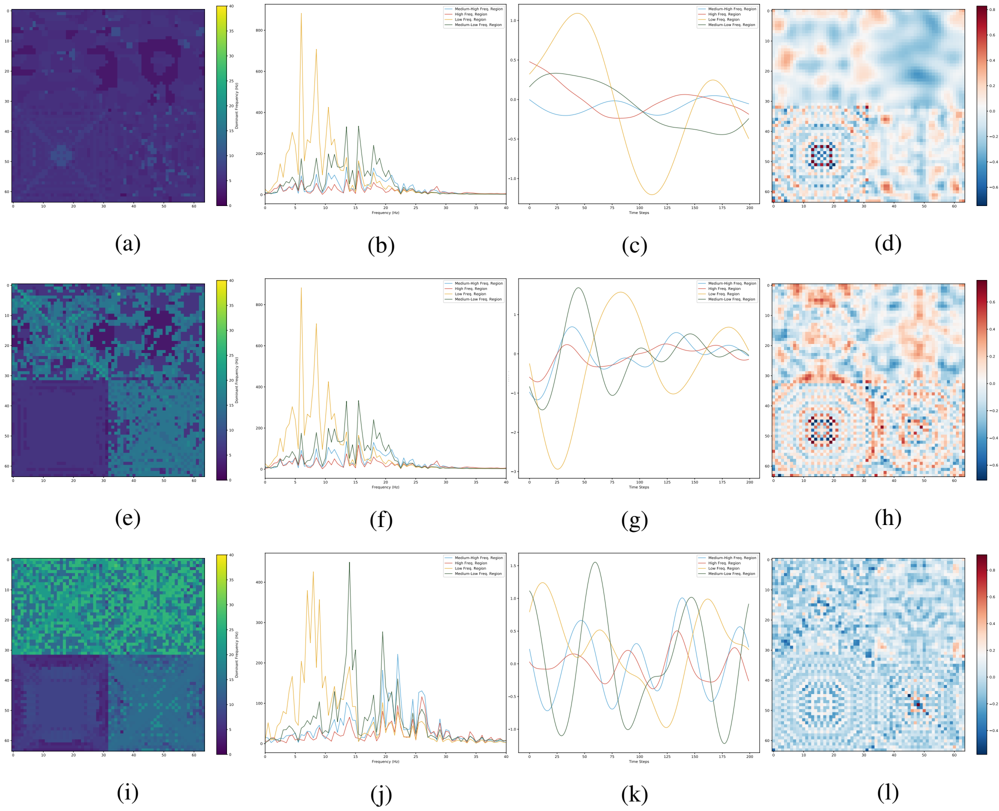
Table 5: Number of parameters, GPU memory usage (in MB) and run time (in seconds) for selected models on all datasets from Table 1 and Table 2 in Section 4.2.

Model	**Config**	**Electricity**	**Solar-Energy**	**Traffic**	**Weather**
LinOSS	#parameters	91521	67785	161310	52821
	GPU memory (MB)	1871	1717	2079	1725
	run time (s)	89	70	105	71
BioOSS	#parameters	101129	22007	26119	134279
	GPU memory (MB)	1765	1659	1967	1653
	run time (s)	170	170	196	162

Table 6: Number of parameters, GPU memory usage (in MB) and run time (in seconds) for selected models on all datasets from Table 3 in Section 4.2.

E Additional Spatio-Temporal Dynamics Related Results

In this experiment, a spatially partitioned 64×64 BioOSS structure was stimulated with white noise inputs of varying frequency bands (0–10 Hz, 0–20 Hz, and 0–30 Hz) to investigate its frequency-selective response properties, as shown in Fig. 2. Each quadrant of the grid was designed to exhibit a distinct natural frequency preference. The dominant frequency response maps (Figs. 3a, 3e, and 3i) reveal that each spatial region selectively responds to components of the input signal that match its predefined natural frequency. When the input white noise was limited to the 0–10 Hz band (first row), all regions exhibited oscillations concentrated around 10 Hz, as no higher frequency components were available to activate region-specific resonances. As the input bandwidth increased to 0–20 Hz (second row), the bottom quadrants began to resonate at their natural frequencies within the lower band, while the top quadrants, unable to find matching excitations at their designated higher natural frequencies, adapted by oscillating around the upper limit of the input band near 20 Hz. This behavior is clearly observable in the frequency spectra and time-domain responses at selected points (Figs. 3f and 3g). Finally, with a 0–30Hz input (third row), the top left quadrant was also able to resonate at its natural frequency within the newly available band (Figs. 3i–3k), whereas the top right quadrant continued to oscillate predominantly around 30 Hz, indicating that its natural resonance lies beyond 30 Hz and thus remained driven by the highest available frequencies. The final pressure field snapshots (Figs. 3d, 3h, and 3l) illustrate the emergent spatial patterns after 2,000 time steps for each input condition. These observations demonstrate the spatial-frequency selectivity of the BioOSS and its capacity to filter and respond dynamically to broadband excitations in a region-specific manner.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 3: **Spatially partitioned frequency-selective responses of the 64 $\times$ 64 BioOSS under varying white noise input bands.** The three rows correspond to different input frequency bands: (a)–(d) 0–10 Hz, (e)–(h) 0–20 Hz, and (i)–(l) 0–30 Hz. Within each row: (a, e, i) depict the spatial distribution of dominant frequencies across the four quadrants of the grid, illustrating distinct regional frequency responses despite identical white noise excitation; (b, f, j) show the frequency spectra at the center of each region, highlighting region-specific resonant behavior; (c, g, k) present the time-domain responses at selected points during the final 200 time steps, reflecting the differentiated temporal dynamics induced by the frequency-selective properties of each partition; (d, h, l) display the final pressure field P after 2,000 time steps, revealing the emergent spatial patterns shaped by the respective input conditions.

F Limitation

The model's non-linear dynamics introduce mild training complexity and hyperparameter sensitivity. Moreover, the linear encoder limits the expression of intrinsic oscillatory behavior, making the model a partial abstraction of biological signals.

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 clearly state the paper's main contributions, including the development of a bio-inspired model to capture the sptio-temporal dependency, its integration with oscillatory dynamics, and its application to a range of temporal tasks. These claims are consistently supported by both theoretical insights and experimental results throughout the paper. The scope of the work are also appropriately framed, ensuring that the reader has a realistic understanding of the model's capabilities and generalization potential. The narrative in the introduction aligns with the broader motivation while accurately delineating the specific contributions made.

   Guidelines:

   - The answer NA 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 NA 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 includes a dedicated "Limitations" section that discusses the main assumptions underlying our approach and how they might affect the generalizability of the results. Additionally, we point out potential sensitivity to hyperparameter settings. By providing a separate section for limitations, we aim to ensure transparency and help readers understand the scope and boundaries of our claims.

   Guidelines:

   - The answer NA 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: The paper includes several theoretical results, each accompanied by a clearly stated set of assumptions and formal statements. All theorems, lemmas, and key formulas are properly numbered and cross-referenced throughout the manuscript. Complete and correct proofs are provided in the appendix, and for results whose proofs appear in the supplemental material, intuitive proof sketches are included in the main text to aid understanding. Furthermore, all supporting theorems and lemmas used in the proofs are appropriately cited and referenced, ensuring clarity and correctness in the theoretical development.

   Guidelines:

   - The answer NA 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 paper provides all necessary details to reproduce the main experimental results, including descriptions of the datasets, model architecture, training procedures, and evaluation protocols. All hyperparameters are specified, the experiments use public benchmark datasets.

   Guidelines:

   - The answer NA 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: All datasets used in the experiments are publicly available, and appropriate citations are provided. The complete implementation code is provided in `https://github.com/zjyuan1208/BioOSS`.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all necessary training and testing details, including data splits, hyperparameter settings, optimizer types. These details are clearly described in the main text and further elaborated in the appendix, ensuring that the experimental results are fully understandable and reproducible.

   Guidelines:

- The answer NA 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: All experiments are conducted with 5 different random seeds. The mean and variance of the results are calculated and reported in the tables to reflect variability. This ensures a clear and appropriate assessment of the statistical significance of the experimental outcomes.

   Guidelines:

   - The answer NA 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 paper specifies the types of GPUs used for the experiments. Memory usage and computation time for each experimental run are detailed in the appendix, providing sufficient information to estimate the resources required to reproduce the experiments.

   Guidelines:

   - The answer NA 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 fully complies with the NeurIPS Code of Ethics. It respects principles related to fairness, transparency, reproducibility, and responsible use of resources. No ethical concerns or deviations from the guidelines are associated with the research conducted in the paper.

Guidelines:

- The answer NA 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: The paper discusses the broader societal impacts and concludes that the research primarily has positive contributions, such as advancing scientific knowledge and potential applications for beneficial technologies. No direct negative societal impacts or risks of misuse have been identified.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper does not involve the release of models or datasets with a high risk of misuse. Therefore, no specific safeguards are necessary.

Guidelines:

- The answer NA 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 makes use of publicly available datasets and baseline models under their respective licenses for non-commercial academic research. All assets are properly cited, and their licenses and terms of use are respected as per the original creators' conditions.

    Guidelines:

    - The answer NA 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: [Yes]

    Justification: The paper introduces a new model and provides detailed documentation within the main text, including the full model architecture, algorithmic flow, and all hyperparameters. While the implementation code is not included in the supplementary material, it will be released publicly upon acceptance, along with usage instructions.

    Guidelines:

    - The answer NA 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: [NA]

Justification: The paper does not involve any crowdsourcing experiments, research with human subjects, or the use of human-related data.

Guidelines:

- The answer NA 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: [NA]

    Justification: The paper does not involve any research with human subjects and thus does not require IRB approval or equivalent ethical review.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The research does not involve the use of large language models (LLMs) as any important, original, or non-standard component of the core methodology.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.