WAVEOPERATOR RNN: LEARNING STRUCTURED WORLD DYNAMICS WITH TRAINABLE LOCAL WAVE PROPAGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Many sequential data are not only temporally ordered but also organized by an underlying topology in which neighboring variables interact locally and information propagates across structured states. Existing sequence models capture temporal or global interactions effectively but rarely encode externally defined local topology explicitly, while neural operators remain largely field-to-field models. We introduce WaveOperator RNN (WO-RNN), which uses local wave propagation as a recurrent operator to combine structured spatial interaction with temporal persistence. Instead of the staggered-grid discretization commonly used for wave equations, WO-RNN adopts a backward-gradient/forward-divergence scheme that preserves local wave coupling in an aligned tensor form. Spatially varying propagation and damping parameters enable heterogeneous, location-dependent dynamics and task-adaptive state evolution. This inductive bias is well suited to sequential and spatiotemporal data with meaningful internal topology. We demonstrate the same operator on tonotopically organized spiking audio and PDE forecasting, illustrating its applicability to sequence modeling and structured world-state evolution. Here, we use world dynamics in the broad sense of predicting how structured latent or physical states evolve through local interactions over time.

1 INTRODUCTION

Many sequential data are not only ordered in time but also exhibit an underlying topology, where neighboring variables interact locally and information propagates across structured states. Examples include spatiotemporal physical systems, event-based sensory signals, and other structured sequences. Designing a general model that can exploit such topology while supporting effective information propagation remains challenging.

Existing sequence models address only part of this problem. Transformers provide powerful global interaction through self-attention, but pairwise computation can be costly for long or high-dimensional sequences, and local topology is not naturally encoded (Vaswani et al., 2017). State-space models enable efficient temporal modeling (Gu et al., 2022; Orvieto et al., 2023; Gu & Dao, 2023), yet standard SSMs mainly model dynamics along the sequence dimension and do not explicitly represent interactions defined by an external spatial or feature topology. Neural operators such as FNO learn mappings between continuous fields and perform strongly on PDEs (Li et al., 2021), but their spectral operators rely on global mode mixing and are primarily designed for function-to-function prediction. More recent operator architectures extend this setting through attention or structured state-space dynamics (Cao, 2021; Hao et al., 2023; Li et al., 2023; Koren & Lanthaler, 2026), but a gap remains between generic sequence modeling and structured predictive dynamics in which state evolution is constrained by meaningful local topology.

We therefore seek an architecture that incorporates local topology directly into its recurrent dynamics, providing a structured transition model for systems whose state evolves through local interaction, propagation, and persistence. Our key insight is that structured information exchange can emerge through repeated local propagation. A sparse geometry allows neighboring states to exchange information iteratively, progressively coupling different parts of the system. This inductive bias is especially useful when the model geometry aligns with meaningful structure in the data, such as

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

tonotopic organization or spatially local physical dynamics. However, the appropriate propagation range, attenuation, and persistence time scales vary across tasks and should be adaptive.

Wave dynamics provide a natural realization of this idea because they combine local transport and temporal persistence within a single state transition. A discretized wave system couples neighboring states, while propagation speed and damping explicitly control how information spreads and decays over time. These properties make wave dynamics a structured recurrent primitive with task-adaptive spatiotemporal behavior. Prior work has explored oscillatory and wave-like recurrent models for sequence processing (Rusch & Mishra, 2021; Keller & Welling, 2023; Keller et al., 2024), but existing approaches do not jointly combine an explicit spatial gradient–divergence structure with trainable propagation and attenuation scales.

We introduce **WaveOperator RNN (WO-RNN)**, a recurrent architecture based on trainable local wave dynamics. WO-RNN organizes its hidden state on a two-dimensional grid and updates neighboring states through a discretized damped wave transition. Its propagation speed and damping coefficients are learned end-to-end, allowing spatial communication and temporal persistence to adapt to the task. The same recurrent mechanism is used for both structured sequence processing and spatiotemporal forecasting.

We evaluate WO-RNN across sequence classification and PDE forecasting to test the scope of this inductive bias. The model is competitive but not dominant on serialized visual sequences, obtains a favorable result on SHD with its 700 tonotopically ordered cochlear channels, and performs well on reaction–diffusion and shallow-water forecasting under our PDEBench-style protocol. Its weaker results on compressible CFD help identify an applicability boundary, while ablations show that spatial coupling and structured recurrent topology materially affect performance.

In summary, our contributions are:

- We introduce trainable local wave dynamics as a recurrent operator (WO-RNN), where spatially varying propagation and damping fields enable location-dependent transport, attenuation, and persistence.
- We impose an explicit nearest-neighbor topology through sparse backward-gradient and forward-divergence operators, while separating fixed spatial coupling from trainable local dynamical parameters.
- We validate WO-RNN across sequence classification and PDE forecasting, using cross-domain evaluations and ablations to identify the regimes in which meaningful local topology and spatial propagation are beneficial.

2 Related Work

2.1 Physics-Structured Recurrent and Wave Models

Reservoir computing processes temporal signals with a fixed recurrent system and trains only its readout (Maass et al., 2002; Jaeger & Haas, 2004), making optimization inexpensive but leaving connectivity without physical geometry. FDTDESN instead derives sparse, linearly scaling local coupling from an FDTD wave discretization, but fixes wave speed and damping for stability and thus cannot adapt its propagation time scale (Yuan et al., 2025). Wave-like alternatives include coRNN's stably discretized second-order system (Rusch & Mishra, 2021), spatially organized Neural Wave Machines (Keller & Welling, 2023), and traveling-wave wRNNs (Keller et al., 2024); none uses the finite-difference gradient–divergence operators of a spatial PDE. BioOSS formulates related two-field dynamics as a state-space system and uses an eigendecomposition-based diagonal approximation to enable associative scans (Yuan et al., 2026).

2.2 State-Space Models and Structured Sequence Learning

LSTMs and GRUs learn general nonlinear state updates (Hochreiter & Schmidhuber, 1997; Cho et al., 2014), whereas structured state-space models constrain latent dynamics for stable, efficient long-range processing: S4 has convolutional and recurrent forms (Gu et al., 2022), LRU uses a stable diagonal complex recurrence (Orvieto et al., 2023), Mamba adds input-dependent transitions

with linear sequence scaling (Gu & Dao, 2023), and LinOSS combines second-order oscillatory dynamics with parallel scans (Rusch & Rus, 2025). Although suitable encodings let these models learn spatial dependencies, their standard transitions neither impose a 2D nearest-neighbor stencil nor assign physical meanings to state components.

2.3 Neural Operators and PDE Forecasting

Neural operators learn function-space mappings, often transferable across inputs or discretizations. FNO uses global Fourier kernels (Li et al., 2021), Galerkin Transformer gives attention a projection interpretation (Cao, 2021), and OFormer and GNOT target spatiotemporal fields and heterogeneous geometries (Li et al., 2023; Hao et al., 2023); PDEBench standardizes their comparison with grid-based baselines on time-dependent PDEs (Takamoto et al., 2022). State Space Neural Operator links structured sequence and operator learning via joint spatiotemporal dynamics, adaptive damping, and spectral selection (Koren & Lanthaler, 2026). These approaches span global spectral kernels, attention mechanisms, and structured state-space dynamics; their goals range from fixed-grid forecasting to discretization-transferable operator approximation.

2.4 Spatial and Tonotopic Inductive Biases

Topographic networks encourage nearby units to learn related representations (Keller & Welling, 2021; Lee et al., 2020; Kubilius et al., 2019), consistent with observations linking local connectivity to traveling population activity (Watakabe et al., 2023; Bhattacharya et al., 2022; Sato, 2022) and frequency-dependent visual and auditory responses (Haas & White, 2002; Da Costa et al., 2013; Goldman-Rakic, 1988). Concretely, SHD's artificial cochlea yields 700 frequency-ordered channels (Cramer et al., 2020), giving adjacent channels meaningful neighborhoods and making SHD a natural benchmark for spatial or tonotopic inductive biases.

3 Methods

WO-RNN, illustrated in Fig. 1, consists of a core wave-coupled recurrent state on a $G \times G$ grid and an optional input-modulation module.

Let $\ell \in \{1, \dots, T\}$ denote the recurrent step and let $N = G^2$ be the number of cells in the spatial grid. The complete hidden state contains one primary field and two auxiliary velocity components,

$$\mathbf{h}^\ell = [p^\ell, \mathbf{o}^\ell], \qquad p^\ell \in \mathbb{R}^{G\times G}, \quad \mathbf{o}^\ell = (o_x^\ell, o_y^\ell) \in \mathbb{R}^{G\times G\times 2}.$$

Equivalently, its vectorized representation belongs to $\mathbb{R}^{3N}$. The wave operator $\mathcal{S}$ acts on the complete previous state but returns its propagated primary component and updated auxiliary field:

$$\begin{aligned} (\bar{p}^\ell, \mathbf{o}^\ell) &= \mathcal{S}(\mathbf{h}^{\ell-1}), \\ \tilde{\mathbf{u}}_\ell &= \mathbf{u}_\ell \odot \mathbf{m}_\ell, \\ p^\ell &= f\left(\bar{p}^\ell + \mathrm{grid}(\mathbf{W}_{\mathrm{in}}\tilde{\mathbf{u}}_\ell)\right), \\ \mathbf{h}^\ell &= [p^\ell, \mathbf{o}^\ell], \\ \mathbf{y}_\ell &= \mathbf{W}_{\mathrm{out}}\,\mathrm{vec}(p^\ell). \end{aligned} \tag{1}$$

Here $\mathbf{u}_\ell, \mathbf{m}_\ell \in \mathbb{R}^d$ are the input and modulation vectors, respectively, and $\odot$ denotes elementwise multiplication. The matrices $\mathbf{W}_{\mathrm{in}} \in \mathbb{R}^{N\times d}$ and $\mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{n_{\mathrm{out}}\times N}$ define the input projection and primary-field readout. $\mathrm{grid} : \mathbb{R}^N \to \mathbb{R}^{G\times G}$ reshapes the projected input, and f is a pointwise nonlinearity applied to the primary field. Thus, the input is injected only into p, whereas $\mathbf{o}$ is updated internally by $\mathcal{S}$. Section 3.3 specifies the single FDTD step performed by $\mathcal{S}$.

3.1 Input Modulation $\mathcal{M}$

We optionally apply a learnable sinusoidal transformation before the input projection to introduce structured variation into the driving signal. This module changes the representation supplied to the wave layer and is optimized jointly with the recurrent model. It is not required to define the physical transition $\mathcal{S}$, and we evaluate its contribution separately by reporting results with and without modulation.

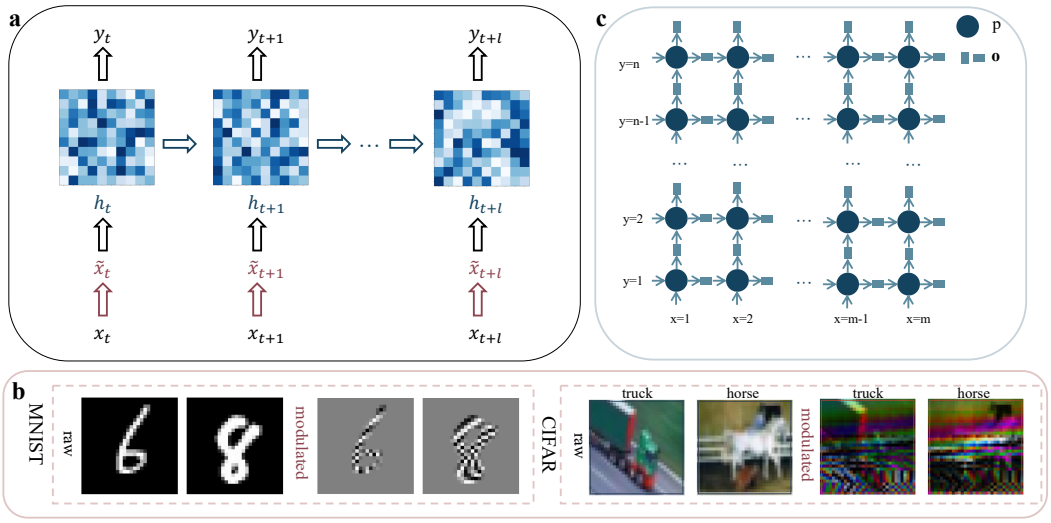


Figure 1: WO-RNN processes sequential data through input modulation and wave propagation. (a) At recurrent step ℓ, the modulation module $\mathcal{M}$ produces $\mathbf{m}_\ell$ and forms $\tilde{\mathbf{u}}_\ell = \mathbf{u}_\ell \odot \mathbf{m}_\ell$. The wave operator $\mathcal{S}$ advances the complete previous state and returns $(\bar{p}^\ell, \mathbf{o}^\ell)$; the projected input is combined only with $\bar{p}^\ell$ to produce p^ℓ. The readout uses the primary field. (b) Example of the modulation applied to an MNIST and CIFAR-10 input. (c) The hidden state $\mathbf{h}^\ell = [p^\ell, \mathbf{o}^\ell]$ is arranged on a 2D grid, where $\mathbf{o}^\ell = (o_x^\ell, o_y^\ell)$ mediates directional nearest-neighbor propagation.

The modulation uses learnable sinusoidal parameters indexed according to the task representation. These parameters are optimized jointly with the model, allowing $\mathbf{m}_\ell$ to adapt the transformed input representation. Section 4.1 gives the task-specific definitions.

3.2 2D Wave-Coupled Hidden State $\mathcal{S}$

The recurrent state contains a primary scalar field p and an auxiliary directional velocity $\mathbf{o} = (o_x, o_y)$. Their local coupling implements wave-like propagation on the 2D grid.

The hidden dynamics follow a damped first-order wave system (Yuan et al., 2025) with an externally driven input term, inspired by propagation in lossy media. We use τ for continuous physical time, reserving ℓ for the recurrent index. The continuous system is

$$\begin{aligned} \frac{\partial p}{\partial \tau} + k^p(\mathbf{x})p + c^2(\mathbf{x})\nabla \cdot \mathbf{o} &= I(\mathbf{x}, \tau), \\ \frac{\partial \mathbf{o}}{\partial \tau} + k^o(\mathbf{x})\mathbf{o} + \nabla p &= 0, \end{aligned} \tag{2}$$

where $\mathbf{x} = (x, y)$, $p(\mathbf{x}, \tau)$ is the primary scalar field, $\mathbf{o}(\mathbf{x}, \tau) = (o_x, o_y)$ is the directional velocity, and $I(\mathbf{x}, \tau)$ is the external input field. We use $c^2(\mathbf{x})\nabla \cdot \mathbf{o}$ as the spatially heterogeneous propagation parameterization of the recurrent operator. The propagation field $c(\mathbf{x})$ controls the strength of local transport, while the nonnegative fields $k^p(\mathbf{x})$ and $k^o(\mathbf{x})$ regulate attenuation of the primary and auxiliary states, respectively. In the recurrent implementation, we use an operator-split recurrent update: $\mathcal{S}$ denotes the autonomous propagation–damping step ($I = 0$), after which the learned input projection is injected into its primary output as specified in Eq. (1).

3.3 Discretization

Let Δt denote the physical time step, let $\Delta x, \Delta y$ denote the grid spacings, and let $c_{m,n} = c(\mathbf{x}_{m,n})$ be the propagation speed at grid cell (m, n). Unlike a homogeneous wave layer with a single scalar speed, WO-RNN permits a spatially heterogeneous speed field $\{c_{m,n}\}_{m,n=1}^{G}$. We define the local dimensionless directional Courant numbers

$$\nu_{x,m,n} = \frac{c_{m,n}\Delta t}{\Delta x}, \qquad \nu_{y,m,n} = \frac{c_{m,n}\Delta t}{\Delta y}. \tag{3}$$

These quantities measure the local propagation distance relative to the grid spacing during one recurrent update. For reference, the corresponding local CFL-like quantity is $\nu^2_{x,m,n} + \nu^2_{y,m,n}$, while the classical homogeneous explicit two-dimensional wave scheme satisfies $\nu_x^2 + \nu_y^2 \leq 1$. For the heterogeneous WO-RNN, we use $\max_{m,n}(\nu^2_{x,m,n} + \nu^2_{y,m,n})$ only as a diagnostic of the propagation scale, not as a stability guarantee for the complete nonlinear recurrent model. The local values of $c_{m,n}$ and the shared step Δt jointly control the spatially varying propagation time scale.

3.3.1 Mixed Finite Difference Discretization

We combine an explicit propagation step with an implicit damping step. Spatial indices are $(m, n) \in \{1, \ldots, G\}^2$, and superscript ℓ denotes the state after processing input $\mathbf{u}_\ell$. The equations use periodic index extension; other boundary conditions require the corresponding boundary stencil. We use a uniform square grid in the experiments but retain Δx and Δy to distinguish physical coefficients from their dimensionless discrete counterparts. Because the continuous system is written without the homogeneous-speed rescaling of $\mathbf{o}$, the auxiliary update contains the gradient of p alone, whereas the primary-field update contains the local factor $c^2_{m,n}$.

Spatial Operators Starting from $\mathbf{h}^{\ell-1}$, the auxiliary field is updated using backward differences:

$$\begin{aligned} o^*_{x,m,n} &= o^{\ell-1}_{x,m,n} - \frac{\Delta t}{\Delta x}(p^{\ell-1}_{m,n} - p^{\ell-1}_{m-1,n}), \\ o^*_{y,m,n} &= o^{\ell-1}_{y,m,n} - \frac{\Delta t}{\Delta y}(p^{\ell-1}_{m,n} - p^{\ell-1}_{m,n-1}), \\ (\nabla_h \cdot \mathbf{o}^*)_{m,n} &= \frac{o^*_{x,m+1,n} - o^*_{x,m,n}}{\Delta x} + \frac{o^*_{y,m,n+1} - o^*_{y,m,n}}{\Delta y}, \\ p^*_{m,n} &= p^{\ell-1}_{m,n} - c^2_{m,n}\Delta t\,(\nabla_h \cdot \mathbf{o}^*)_{m,n}. \end{aligned} \tag{4}$$

Here $\nabla_h \cdot$ denotes the discrete forward-divergence operator, and the superscript $*$ identifies the intermediate pre-damping fields. Under periodic boundary conditions, the backward-gradient and forward-divergence operators form a negative-adjoint pair, preserving the discrete gradient–divergence structure.

The implicit damping step produces the output of the wave operator $\mathcal{S}$:

$$\mathbf{o}^\ell_{m,n} = \frac{\mathbf{o}^*_{m,n}}{1 + \Delta t\, k^o_{m,n}}, \qquad \bar{p}^\ell_{m,n} = \frac{p^*_{m,n}}{1 + \Delta t\, k^p_{m,n}}. \tag{5}$$

The quantities $k^p_{m,n}$ and $k^o_{m,n}$ retain the units of inverse time; their dimensionless decay parameters are $\gamma^p_{m,n} = \Delta t\, k^p_{m,n}$ and $\gamma^o_{m,n} = \Delta t\, k^o_{m,n}$. We do not rename these products as k^p or k^o. Finally, the input is injected into the primary field as

$$I^\ell_{m,n} = [\mathrm{grid}(\mathbf{W}_{\mathrm{in}}(\mathbf{u}_\ell \odot \mathbf{m}_\ell))]_{m,n}\,, \qquad p^\ell_{m,n} = f(\bar{p}^\ell_{m,n} + I^\ell_{m,n}). \tag{6}$$

Thus, $\mathcal{S}(\mathbf{h}^{\ell-1}) = (\bar{p}^\ell, \mathbf{o}^\ell)$ performs exactly one propagation–damping step, and only its primary component is combined with the projected input. Locally, the characteristic propagation times between adjacent cells are $\Delta x/c_{m,n}$ and $\Delta y/c_{m,n}$ (Iacob & Dambre, 2024).

Appendix A further characterizes the WO-RNN transition at the operator level. Fig. A.1 relates the aligned backward-gradient/forward-divergence implementation to classical staggered FDTD, while Fig. A.2 shows that the resulting block operator separates a fixed sparse neighbor topology from the trainable diagonal fields $\mathbf{C}^2$, $\mathbf{K}_p$, and $\mathbf{K}_o$. Figs. A.3–A.5 then isolate the dynamical roles of these fields: spatial variation in c changes propagation, k_p changes primary-state attenuation, and k_o changes auxiliary-velocity persistence, producing distinct impulse responses and pole distributions without modifying the topology. All nonnegative configurations analyzed there remain inside the linear unit-circle stability boundary; Fig. A.6 additionally uses a deliberately signed k_p field to illustrate how local anti-damping can produce expansive linear modes. These poles characterize only the autonomous linear propagation–damping operator, since the complete recurrence additionally applies a nonlinear activation to the primary state. This dynamics-side heterogeneity is distinct from input-side heterogeneity introduced by modulation and the learned input projection.

Algorithm 1 summarizes the complete forward pass of WO-RNN, making the two-phase structure explicit.

Algorithm 1 WO-RNN Forward Pass

Require: Sequence $\{\mathbf{u}_1, \ldots, \mathbf{u}_T\}$, initial state $\mathbf{h}^0 = [p^0, \mathbf{o}^0]$, $\{c_{m,n}\}$, k^p, k^o, Δt, Δx, Δy
Ensure: Outputs $\{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$

for $\ell = 1$ **to** T **do**
  $\tilde{\mathbf{u}}_\ell \leftarrow \mathbf{u}_\ell \odot \mathbf{m}_\ell$ ▷ Input modulation
  Phase 1: Wave propagation
  $o^*_{x,m,n} \leftarrow o^{\ell-1}_{x,m,n} - (\Delta t/\Delta x)(p^{\ell-1}_{m,n} - p^{\ell-1}_{m-1,n})$ ▷ Backward gradient
  $o^*_{y,m,n} \leftarrow o^{\ell-1}_{y,m,n} - (\Delta t/\Delta y)(p^{\ell-1}_{m,n} - p^{\ell-1}_{m,n-1})$
  $(\nabla_h \cdot \mathbf{o}^*)_{m,n} \leftarrow (o^*_{x,m+1,n} - o^*_{x,m,n})/\Delta x + (o^*_{y,m,n+1} - o^*_{y,m,n})/\Delta y$
  $p^*_{m,n} \leftarrow p^{\ell-1}_{m,n} - c^2_{m,n}\Delta t(\nabla_h \cdot \mathbf{o}^*)_{m,n}$
  Phase 2: Damping
  $\mathbf{o}^\ell_{m,n} \leftarrow \mathbf{o}^*_{m,n}/(1 + \Delta t\, k^o_{m,n})$
  $\bar{p}^\ell_{m,n} \leftarrow p^*_{m,n}/(1 + \Delta t\, k^p_{m,n})$
  $I^\ell \leftarrow \mathrm{grid}(\mathbf{W}_{\mathrm{in}}\tilde{\mathbf{u}}_\ell)$
  $p^\ell_{m,n} \leftarrow f(\bar{p}^\ell_{m,n} + I^\ell_{m,n})$
  $\mathbf{h}^\ell \leftarrow [p^\ell, \mathbf{o}^\ell]$
  $\mathbf{y}_\ell \leftarrow \mathbf{W}_{\mathrm{out}}\,\mathrm{vec}(p^\ell)$
end for

4 Experiments

Our experiments test whether the proposed local wave operator remains useful across distinct data structures and identify the regimes in which its spatial inductive bias is beneficial. Sequence classification and PDE forecasting provide complementary cross-domain scope tests: the sequence tasks cover visual and spiking-audio modalities, including SHD with its tonotopically ordered cochlear channels, while the forecasting tasks cover three PDE systems. All hyperparameters and environment settings are detailed in Appendix B.

4.1 Tasks and Evaluation Protocols

We consider two sequence-structure regimes. The *serialized visual sequences*, sMNIST (28 steps, 28 features) and nsCIFAR-10 (64 steps, 48 features), have no explicit domain-aligned correspondence between their input features and the WO-RNN recurrent grid. The *structured auditory sequence*, SHD, contains 250 time bins and 700 tonotopically ordered cochlear channels, so neighboring input features have a domain-specific relationship.

Input modulation is treated as an optional preprocessing module and evaluated separately with and without modulation. For an input $\mathbf{u}_\ell \in \mathbb{R}^d$, feature index $q \in \{0, \ldots, d-1\}$, and recurrent step ℓ, modulation is applied elementwise as $\tilde{u}_{\ell,q} = u_{\ell,q} \odot m_{\ell,q}$. The optional module $\mathcal{M}$ generates the task-specific vector $\mathbf{m}_\ell = (m_{\ell,0}, \ldots, m_{\ell,d-1})$ using learnable frequencies f_ℓ. The unmodulated setting uses $\mathbf{m}_\ell = \mathbf{1}$.

Sequential MNIST (sMNIST). Each 28×28 image is represented as 28 row vectors $\mathbf{u}_\ell \in \mathbb{R}^{28}$. We use $m_{\ell,q} = \sin(2\pi f_\ell q/d)$ with $d = 28$.

Noisy CIFAR-10 (nsCIFAR-10). Each 32×32 RGB image is divided into 64 flattened 4×4 patches, giving $\mathbf{u}_\ell \in \mathbb{R}^{48}$. We use $m_{\ell,q} = 0.5 + 0.5\sin(2\pi f_\ell q/d)$ with $d = 48$.

SHD (Spiking Heidelberg Digits). SHD represents each sample as an event stream of spike times and cochlear units. We bin each 1.4-s recording into 250 intervals (5.6 ms each), yielding $\mathbf{u}_\ell \in \mathbb{R}^{700}_{\geq 0}$ over tonotopically ordered channels. We use $m_{\ell,q} = (1 - \alpha_q) + \alpha_q[\sin(2\pi f_q \ell/250 + \phi_q) + 1]/2$, where f_q, ϕ_q, and α_q are learned per channel; setting $\alpha_q = 0$ recovers identity modulation.

For spatial-field forecasting, we evaluate reaction–diffusion (RD2D), shallow water (SW2D), and compressible fluid dynamics (2D CFD) from PDEBench. Section 4.4 describes the comparison protocol, and Appendix D provides its full configuration.

4.2 Sequence Modeling: When Does Structured Propagation Help?

We ask whether the local wave inductive bias behaves differently across sequence structures. Table 1 serves as a scope test against LSTM (Hochreiter & Schmidhuber, 1997), GRU (Cho et al., 2014), modern SSMs, biologically inspired recurrent models, ESN, and FDTDESN (Yuan et al., 2025), the closest fixed-wave reference. WO-RNN achieves 98.10% on sMNIST, 54.20% on nsCIFAR-10, and 90.59% on SHD under the evaluated setting. The contrast across tasks is the main result.

Table 1: Cross-domain scope test on sequence classification. Accuracy (%) is reported for sMNIST, nsCIFAR-10, and SHD; each cell shows w/o modulation $\rightarrow$ w/ modulation, with the change in parentheses.

Model	**sMNIST**	**nsCIFAR-10**	**SHD**
ESN	91.50 $\rightarrow$ 86.10 (-5.40)	38.40 $\rightarrow$ 34.90 (-3.50)	5.74 $\rightarrow$ 6.63 (+0.89)
FDTDESN (full $p + \mathbf{o}$)	97.79 $\rightarrow$ 97.70 (-0.09)	49.85 $\rightarrow$ 50.18 (+0.33)	77.16 $\rightarrow$ 81.32 (+4.16)
LSTM	98.80 $\rightarrow$ 97.40 (-1.40)	58.30 $\rightarrow$ 57.40 (-0.90)	89.40 $\rightarrow$ 89.89 (+0.49)
GRU	99.10 $\rightarrow$ 97.50 (-1.60)	60.80 $\rightarrow$ 61.20 (+0.40)	87.59 $\rightarrow$ 86.53 (-1.06)
Mamba	99.20 $\rightarrow$ 98.71 (-0.49)	64.18 $\rightarrow$ 63.92 (-0.26)	80.48 $\rightarrow$ 82.42 (+1.94)
LRU	98.90 $\rightarrow$ 98.45 (-0.45)	65.30 $\rightarrow$ 65.94 (+0.64)	82.73 $\rightarrow$ 87.15 (+4.42)
S4D	98.57 $\rightarrow$ 97.86 (-0.71)	70.07 $\rightarrow$ 64.54 (-5.53)	86.84 $\rightarrow$ 82.42 (-4.42)
NWM	98.60 $\rightarrow$ 94.90 (-3.70)	56.20 $\rightarrow$ 54.10 (-2.10)	68.29 $\rightarrow$ 68.51 (+0.22)
coRNN	99.30 $\rightarrow$ 97.00 (-2.30)	59.00 $\rightarrow$ 37.30 (-21.70)	69.13 $\rightarrow$ 69.92 (+0.79)
wRNN	97.60 $\rightarrow$ 95.90 (-1.70)	55.00 $\rightarrow$ 16.10 (-38.90)	74.56 $\rightarrow$ 76.72 (+2.16)
WO-RNN	97.00 $\rightarrow$ 98.10 (+1.10)	53.40 $\rightarrow$ 54.20 (+0.80)	81.32 $\rightarrow$ 90.59 (+9.27)

On the serialized visual tasks, the recurrent grid does not correspond to an explicit input-space topology. WO-RNN is competitive on sMNIST and less effective on nsCIFAR-10 relative to the generic sequence models in the table. This pattern is consistent with, but does not establish, the hypothesis that the local inductive bias is most useful when it can align with meaningful feature locality.

The most favorable relative result occurs on SHD, whose cochlear channels provide a meaningful local ordering. WO-RNN reaches 90.59% and performs better than the evaluated LSTM, GRU, S4D, and LRU configurations under this setting. This result is consistent with the hypothesis that WO-RNN benefits when neighboring latent interactions can align with meaningful feature locality.

Relative to FDTDESN (Yuan et al., 2025), the closest fixed-wave reference, WO-RNN obtains higher accuracies on all three tasks under the configurations in Table 1. This comparison does not isolate trainability from every architectural difference.

Finally, modulation is not uniformly beneficial across architectures or datasets. For WO-RNN, its largest gain occurs on SHD, whereas its gains on the two visual tasks are smaller. This suggests that structured input variation is particularly useful when feature dimensions possess a meaningful local ordering. We examine this effect together with the recurrent spatial components in the next section.

4.3 Ablation and Interpretation of Spatial Inductive Bias

We examine the roles of spatial propagation, auxiliary dynamics, topology, and input modulation in WO-RNN. In the reported sequence experiments, the propagation and damping fields are initialized homogeneously and then optimized end-to-end. Input modulation therefore provides an additional source of structured variation in how the recurrent field is excited. Because the Single Field + No Mod. variant changes two components jointly, it should only be interpreted as a combined ablation.

Full Model: Includes wave propagation ($c \neq 0$), coupled $p/\mathbf{o}$ fields, a structured 2D grid, and input modulation.

No Spatial Coupling ($c = 0$): Removes inter-cell propagation while retaining the remaining recurrent structure and modulation.

Single Field + No Mod.: Removes $\mathbf{o}$ and disables modulation, yielding a primary-field-only recurrence.

Random Topology: Retains propagation and modulation but replaces the regular grid neighborhood with random connections.

Table 2: WO-RNN ablations. Test accuracy (%) is reported, with the decrease from the full model in parentheses. Single Field + No Mod. is a joint ablation.

Model	sMNIST	nsCIFAR-10	SHD
Full WO-RNN	**98.1**	**54.2**	**90.59**
No Spatial Coupling	95.75 (-2.35)	44.60 (-9.60)	81.89 (-8.70)
Single Field + No Mod.	92.26 (-5.84)	37.91 (-16.29)	75.80 (-14.79)
Random Topology	72.85 (-25.25)	24.29 (-29.91)	81.05 (-9.54)

Table 2 shows that removing spatial coupling degrades all three tasks, while randomizing the topology causes even larger drops, especially on sMNIST and nsCIFAR-10. This indicates that both inter-cell propagation and the organization of local connections contribute to performance. The Single Field + No Mod. variant further degrades accuracy, but its joint removal of $\mathbf{o}$ and modulation prevents attribution to either component alone.

The effect of modulation is closely tied to the initialization of the recurrent dynamics. With homogeneous coefficient initialization, modulation introduces structured variation before the signal enters the recurrent field. Its largest gain occurs on SHD, where the cochlear channels have a meaningful tonotopic ordering, suggesting that differentiated input excitation is most useful when the input itself has local structure. In exploratory runs, spatially heterogeneous random initialization yielded comparable performance without modulation. Although these runs are not included in the main comparison, they suggest that modulation and heterogeneous initialization provide partially overlapping sources of variation: the former differentiates how the recurrent field is driven, whereas the latter differentiates how activity propagates and decays across space.

Consistent with this interpretation, the operator analysis in Appendix A shows that spatial variation in c, k_p, and k_o modifies propagation, primary-state attenuation, and auxiliary-velocity persistence, respectively, thereby reshaping the impulse responses and linear pole distributions while preserving the same sparse neighbor topology. The present experiments do not provide a controlled homogeneous-versus-heterogeneous coefficient ablation, so we do not attribute the reported accuracy gains specifically to coefficient heterogeneity. Rather, the ablations directly support the importance of spatial propagation and topology, while heterogeneity should be interpreted as a mechanism for diversifying local propagation, attenuation, and persistence time scales.

Appendix C further examines sensitivity to the integration step and propagation speed. The visual sequence tasks tolerate broader parameter ranges, whereas SHD benefits from coordinated tuning of Δt and c.

4.4 Cross-Domain Validation on PDE Forecasting

Having established the role of structured propagation and topology on sequence tasks, we next test whether the same recurrent operator transfers to predictive world-state dynamics, where the model must evolve a structured spatial state forward in time and locality is intrinsic to the data. We evaluate WO-RNN on PDEBench (Takamoto et al., 2022), a standardized suite of time-dependent PDE forecasting tasks. We use three two-dimensional benchmarks: reaction–diffusion (RD2D), shallow water (SW2D), and compressible fluid dynamics (2D CFD). Following the PDEBench evaluation convention, we report normalized root mean squared error (nRMSE), where lower is better. We separate the published PDEBench results for U-Net (Takamoto et al., 2022), FNO (Li et al., 2021), and PINN (Takamoto et al., 2022) from ResNet, GNOT (Hao et al., 2023), Galerkin Transformer (Cao, 2021), OFormer (Li et al., 2023), LRU (Orvieto et al., 2023), S4D, Mamba (Gu & Dao, 2023), and

Table 3: PDEBench forecasting results. Lower nRMSE is better. Superscripts indicate source: † denotes PDEBench-reported baselines, and * denotes models evaluated locally under our PDEBench-style protocol with autoregressive rollout.

Model	RD2D nRMSE	SW2D nRMSE	2D CFD nRMSE
U-Net†	0.8400	0.0830	5.1000
FNO†	0.1200	**0.0044**	0.3600
PINN†	1.6000	0.0170	–
ResNet*	1.4927	0.1089	**0.3450**
GNOT*	0.9880	0.0854	0.9859
Galerkin Transformer*	0.6647	0.1067	0.3863
OFormer*	0.9877	0.1006	2.1566
LRU*	0.9344	0.0134	0.4284
S4D*	0.9344	0.0128	0.4050
Mamba*	0.9343	0.0129	0.4020
WO-RNN*	**0.0522**	0.0045	0.3990

WO-RNN, which we evaluate locally under a common PDEBench-style protocol. The published numbers are contextual references rather than controlled head-to-head comparisons.

Under the common local protocol in Table 3, WO-RNN obtains nRMSE values of 0.0522 on RD2D and 0.0045 on SW2D, the lowest errors among the locally evaluated configurations. The separately published FNO value on SW2D is 0.0044 and is included only as context. On 2D CFD, WO-RNN obtains 0.3990, while the local ResNet configuration obtains 0.3450. These outcomes are consistent with the local wave transition being well matched to RD2D and SW2D under the evaluated setting, whereas compressible CFD marks a less favorable regime. More broadly, the same recurrent transition applies to both vector sequences and spatial-field forecasting.

Taken together, the results support viewing WO-RNN as a topology-aware inductive bias rather than a universally stronger recurrent architecture. Its strongest sequence result occurs on SHD, where local feature structure is explicit, whereas gains are smaller on serialized visual inputs without a direct correspondence to the recurrent grid. A similar pattern appears in PDE forecasting, where WO-RNN performs strongly on reaction–diffusion and shallow-water systems but is less competitive on compressible CFD. This suggests that the wave transition is most suitable when task dynamics can be represented through structured local propagation and persistent spatial interaction.

For the PDEBench experiments, detailed hyperparameters and trainable parameter counts are reported in Appendix D. PDEBench-reported U-Net, FNO, and PINN are excluded from that table because they were not retrained locally.

5 Conclusion

We introduced WaveOperator RNN (WO-RNN), a topology-aware recurrent architecture built from trainable local wave dynamics. By combining sparse backward-gradient and forward-divergence coupling with spatially varying propagation and damping fields, WO-RNN supports location-dependent propagation, attenuation, and recurrent time scales while retaining a fixed explicit local topology. The same operator can be applied to both sequential inputs and spatiotemporal fields, connecting conventional sequence modeling with PDE forecasting within a unified recurrent formulation. Our experiments suggest that this inductive bias is most useful when the data exhibit meaningful internal organization, as illustrated by tonotopically structured audio and spatial PDE dynamics, while offering less benefit on serialized inputs without an explicit topology. These results position trainable wave operators as a structured transition mechanism for recurrent representation learning and predictive world dynamics, particularly in systems where future states emerge from local propagation, attenuation, and persistent spatial interaction.

AI USE STATEMENT

Large language models (LLMs) were used solely to assist with language editing, grammar refinement, and typo checking during manuscript preparation. The scientific ideas, methodology, experimental design, implementation, analysis, and conclusions were developed and verified by the authors.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, the main paper provides a complete description of the proposed WO-RNN architecture, recurrent update equations, and Algorithm 1 for the forward computation. The appendix further gives the explicit sparse block form of the heterogeneous wave transition and reports dataset splits, training environment, implementation details, hyperparameter settings, sensitivity analyses, ablation studies, and the complete PDEBench evaluation protocol. These materials provide sufficient information to reproduce the reported experiments. The source code will be publicly released upon acceptance.

REFERENCES

Sayak Bhattacharya, Scott L Brincat, Mikael Lundqvist, and Earl K Miller. Traveling waves in the prefrontal cortex during working memory. *PLoS computational biology*, 18(1):e1009827, 2022.

Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2744–2757, 2020.

Sandra Da Costa, Wietske van der Zwaag, Lee M Miller, Stephanie Clarke, and Melissa Saenz. Tuning in to sound: frequency-selective attentional filter in human primary auditory cortex. *Journal of Neuroscience*, 33(5):1858–1863, 2013.

Patricia S Goldman-Rakic. Topography of cognition: parallel distributed networks in primate association cortex. *Annual review of neuroscience*, 1988.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=uYLFoz1vlAC`.

Julie S Haas and John A White. Frequency selectivity of layer ii stellate cells in the medial entorhinal cortex. *Journal of neurophysiology*, 88(5):2422–2429, 2002.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International conference on machine learning*, pp. 12556–12569. PMLR, 2023.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Stefan Iacob and Joni Dambre. Exploiting signal propagation delays to match task memory requirements in reservoir computing. *Biomimetics*, 9(6):355, 2024.

Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science*, 304(5667):78–80, 2004.

T Anderson Keller and Max Welling. Topographic vaes learn equivariant capsules. *Advances in Neural Information Processing Systems*, 34:28585–28597, 2021.

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*, pp. 16168–16189. PMLR, 2023.

T Anderson Keller, Lyle Muller, Terrence Sejnowski, and Max Welling. Traveling waves encode the recent past and enhance sequence learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Nodens Koren and Samuel Lanthaler. Merging memory and space: A state space neural operator. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=SwLxxz0x58.

Jonas Kubilius, Martin Schrimpf, Kohitij Kar, Rishi Rajalingham, Ha Hong, Najib Majaj, Elias Issa, Pouya Bashivan, Jonathan Prescott-Roy, Kailyn Schmidt, et al. Brain-like object recognition with high-performing shallow recurrent anns. *Advances in neural information processing systems*, 32, 2019.

Hyodong Lee, Eshed Margalit, Kamila M Jozwik, Michael A Cohen, Nancy Kanwisher, Daniel LK Yamins, and James J DiCarlo. Topographic deep artificial neural networks reproduce the hallmarks of the primate inferior temporal cortex face processing network. *BioRxiv*, pp. 2020–07, 2020.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=EPPqt3uERT.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14 (11):2531–2560, 2002.

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*, pp. 26670–26698. PMLR, 2023.

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 *International Conference on Learning Representations*, volume 2025, pp. 22303–22322, 2025.

Naoyuki Sato. Cortical traveling waves reflect state-dependent hierarchical sequencing of local regions in the human connectome network. *Scientific Reports*, 12(1):334, 2022.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, 2022.

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.

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.

Zhongju Yuan, Wannes Van Ransbeeck, Geraint A. Wiggins, and Dick Botteldooren. A dynamic systems approach to modeling humanmachine rhythm interaction. *IEEE Transactions on Cybernetics*, 55(5):2052–2064, 2025. doi: 10.1109/TCYB.2025.3547216.

Zhongju Yuan, Geraint Wiggins, and Dick Botteldooren. Biooss: A bio-inspired oscillatory state system with spatio-temporal dynamics. *Advances in Neural Information Processing Systems*, 38: 157137–157161, 2026.

A EXPLICIT STRUCTURE OF THE WO-RNN WAVE TRANSITION

Following the explicit FDTD coupling-matrix construction developed for wave-based reservoirs (Yuan et al., 2025), we express the linear propagation–damping component of WO-RNN as a sparse block operator. The present formulation differs in three respects: its propagation and damping coefficients are trainable spatial fields, it uses the backward-gradient/forward-divergence update in Section 3.3 with c^2 appearing only in the primary-field branch, and the resulting operator is used directly inside the recurrent transition rather than being converted into a reservoir weight matrix.

A.1 SPARSE DIFFERENCE OPERATORS AND HETEROGENEOUS FIELDS

For compact notation, define the coordinate-wise backward-difference operators D_x^-, D_y^- and forward-difference operators D_x^+, D_y^+ by

$$\begin{aligned} (D_x^- p)_{m,n} &\equiv \frac{p_{m,n} - p_{m-1,n}}{\Delta x}, & (D_y^- p)_{m,n} &\equiv \frac{p_{m,n} - p_{m,n-1}}{\Delta y}, \\ (D_x^+ o_x)_{m,n} &\equiv \frac{o_{x,m+1,n} - o_{x,m,n}}{\Delta x}, & (D_y^+ o_y)_{m,n} &\equiv \frac{o_{y,m,n+1} - o_{y,m,n}}{\Delta y}. \end{aligned} \tag{A.1}$$

Thus, D^- denotes the backward differences used for the discrete gradient, whereas D^+ denotes the forward differences used for the discrete divergence. These are precisely the one-sided differences written explicitly in Section 3.3; boundary indices follow the same implementation convention.

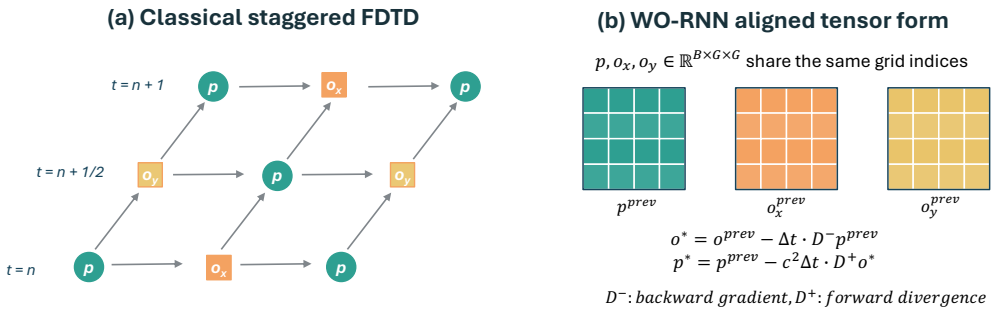


Figure A.1: Comparison of state representations. **(a)** Classical staggered FDTD places pressure at integer spatial and temporal positions and directional velocity at half-grid and half-time positions. **(b)** WO-RNN stores p, o_x, and o_y on aligned tensor indices and realizes the backward-gradient and forward-divergence updates through shifted tensor operations. This aligned representation is convenient for batched automatic differentiation and accelerator execution; it is not a separate guarantee of backpropagation stability.

Let $N = G^2$ and vectorize $p, o_x, o_y \in \mathbb{R}^{G\times G}$ as vectors in $\mathbb{R}^N$. We denote the corresponding matrix representations by $\mathbf{S}_x, \mathbf{S}_y, \mathbf{R}_x, \mathbf{R}_y \in \mathbb{R}^{N\times N}$, defined through $\mathbf{S}_x \operatorname{vec}(p) = \operatorname{vec}(D_x^- p)$, $\mathbf{S}_y \operatorname{vec}(p) = \operatorname{vec}(D_y^- p)$, $\mathbf{R}_x \operatorname{vec}(o_x) = \operatorname{vec}(D_x^+ o_x)$, and $\mathbf{R}_y \operatorname{vec}(o_y) = \operatorname{vec}(D_y^+ o_y)$. Below, we reuse the field symbols for their vectorized forms. Here, $\mathbf{S}_x$ and $\mathbf{S}_y$ are the backward-gradient matrices, while $\mathbf{R}_x$ and $\mathbf{R}_y$ are the forward-divergence matrices. These matrices encode the fixed nearest-neighbor topology. The trainable heterogeneous coefficients are represented separately by the diagonal matrices

$$\begin{aligned} \mathbf{C}^2 &= \operatorname{diag}(c_1^2, \ldots, c_N^2), \\ \mathbf{K}_p &= \operatorname{diag}(k_1^p, \ldots, k_N^p), \\ \mathbf{K}_o &= \operatorname{diag}(k_1^o, \ldots, k_N^o), \end{aligned} \tag{A.2}$$

where each flattened index corresponds to one grid cell (m, n). The implicit damping operators are

$$\mathbf{D}_p = (\mathbf{I}_N + \Delta t\, \mathbf{K}_p)^{-1}, \qquad \mathbf{D}_o = (\mathbf{I}_N + \Delta t\, \mathbf{K}_o)^{-1}. \tag{A.3}$$

Fig. A.1 contrasts classical space–time staggering with the aligned tensor representation used to apply these operators in WO-RNN.

A.2 Explicit Block Transition

In vectorized form, the propagation phase in Eq. (4) is

$$\begin{aligned} o_x^* &= o_x^{\ell-1} - \Delta t\, \mathbf{S}_x p^{\ell-1}, \\ o_y^* &= o_y^{\ell-1} - \Delta t\, \mathbf{S}_y p^{\ell-1}, \\ p^* &= p^{\ell-1} - \Delta t\, \mathbf{C}^2 (\mathbf{R}_x o_x^* + \mathbf{R}_y o_y^*). \end{aligned} \tag{A.4}$$

Substituting the auxiliary updates into the primary-field update gives

$$\begin{aligned} p^* = \; & \left[\mathbf{I}_N + \Delta t^2 \mathbf{C}^2 (\mathbf{R}_x \mathbf{S}_x + \mathbf{R}_y \mathbf{S}_y)\right] p^{\ell-1} \\ & - \Delta t\, \mathbf{C}^2 \mathbf{R}_x o_x^{\ell-1} - \Delta t\, \mathbf{C}^2 \mathbf{R}_y o_y^{\ell-1}. \end{aligned} \tag{A.5}$$

Applying the implicit damping operators yields

$$\begin{bmatrix} \bar{p}^\ell \\ o_x^\ell \\ o_y^\ell \end{bmatrix} = \mathbf{A}_{\mathrm{WO}} \begin{bmatrix} p^{\ell-1} \\ o_x^{\ell-1} \\ o_y^{\ell-1} \end{bmatrix}, \tag{A.6}$$

with

$$\mathbf{A}_{\mathrm{WO}} = \begin{bmatrix} \mathbf{D}_p \left[\mathbf{I}_N + \Delta t^2 \mathbf{C}^2 (\mathbf{R}_x \mathbf{S}_x + \mathbf{R}_y \mathbf{S}_y)\right] & -\Delta t\, \mathbf{D}_p \mathbf{C}^2 \mathbf{R}_x & -\Delta t\, \mathbf{D}_p \mathbf{C}^2 \mathbf{R}_y \\ -\Delta t\, \mathbf{D}_o \mathbf{S}_x & \mathbf{D}_o & \mathbf{0} \\ -\Delta t\, \mathbf{D}_o \mathbf{S}_y & \mathbf{0} & \mathbf{D}_o \end{bmatrix}. \tag{A.7}$$

Eq. (A.7) separates the fixed sparse topology $(\mathbf{R}_x, \mathbf{R}_y, \mathbf{S}_x, \mathbf{S}_y)$ from the trainable heterogeneous diagonal fields $(\mathbf{C}^2, \mathbf{D}_p, \mathbf{D}_o)$. This factorization is the explicit matrix form of the wave operator $\mathcal{S}$ used in the implementation, and Fig. A.2 visualizes its block sparsity and diagonal factors.

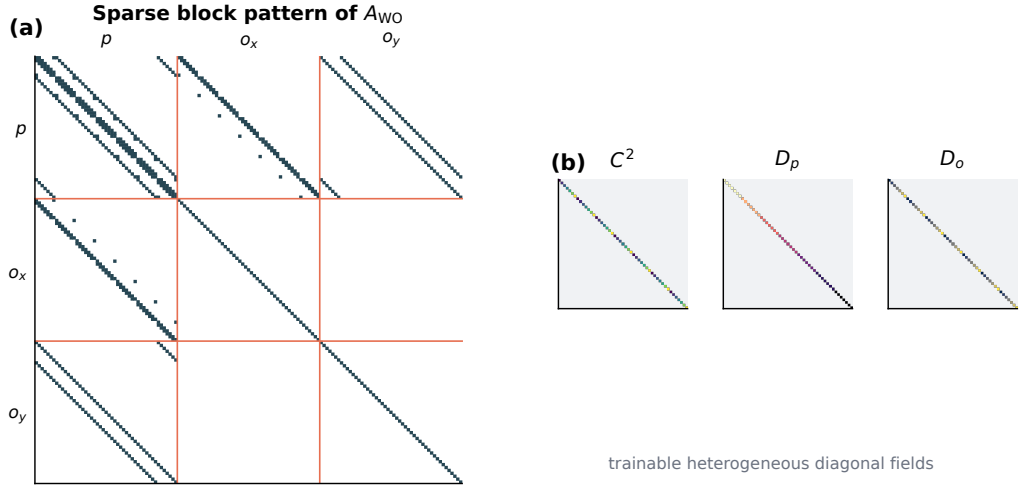


Figure A.2: Structure of the explicit WO-RNN wave operator for an illustrative $G = 7$ periodic grid. **(a)** The nonzero pattern of $\mathbf{A}_{\mathrm{WO}}$ is determined by the local backward-gradient/forward-divergence topology; red lines mark the p, o_x, and o_y block boundaries. **(b)** The propagation and damping factors enter as the diagonal fields $\mathbf{C}^2$, $\mathbf{D}_p$, and $\mathbf{D}_o$. The displayed values are illustrative, whereas the separation between fixed sparse coupling and trainable diagonal fields applies generally.

A.3 Heterogeneous Fields, Linear Poles, and Nonlinear Saturation

For homogeneous coefficients, the diagonal fields reduce to scalar multiples of the identity:

$$\mathbf{C}^2 = c_0^2 \mathbf{I}_N, \qquad \mathbf{K}_p = k_0^p \mathbf{I}_N, \qquad \mathbf{K}_o = k_0^o \mathbf{I}_N. \tag{A.8}$$

Every grid cell then shares the same propagation and attenuation scales. WO-RNN instead permits spatially varying coefficient fields,

$$c(x, y), \qquad k_p(x, y), \qquad k_o(x, y), \tag{A.9}$$

where c controls the local propagation scale, k_p attenuates the primary field, and k_o controls attenuation and persistence of the auxiliary velocity. Because these coefficients enter through diagonal matrices, heterogeneity modifies local dynamics without changing the fixed backward-gradient/forward-divergence topology.

Figs. A.3–A.5 isolate these effects in controlled autonomous rollouts. For each coefficient, we compare a homogeneous field, a spatial gradient, and a smoothly random heterogeneous field, with the heterogeneous fields restricted to the same prescribed range. Only the indicated coefficient varies within each comparison; the remaining coefficients, periodic topology, time step, initial impulse, and rollout procedure are fixed. We examine both the primary-field response and the eigenspectrum of $\mathbf{A}_{\mathrm{WO}}$.

For the autonomous linear recurrence

$$\mathbf{h}^{\ell} = \mathbf{A}_{\mathrm{WO}}\mathbf{h}^{\ell-1}, \tag{A.10}$$

the eigenvalues of $\mathbf{A}_{\mathrm{WO}}$ are its discrete-time modes, or poles in a recurrent-system interpretation. Their modulus determines the asymptotic behavior of each linear mode:

$$|\lambda_i| < 1 \Rightarrow \text{decay}, \qquad |\lambda_i| = 1 \Rightarrow \text{marginal persistence}, \qquad |\lambda_i| > 1 \Rightarrow \text{growth}. \tag{A.11}$$

Accordingly, the spectral radius $\rho(\mathbf{A}_{\mathrm{WO}}) = \max_i |\lambda_i|$ determines asymptotic stability of this finite-dimensional autonomous linear system. The spectra below are computed from the same 12×12 periodic-grid operator used for the impulse-response illustrations.

The three coefficient families therefore affect the recurrent dynamics through distinct mechanisms. Spatial variation in c primarily changes propagation and oscillatory structure; k_p directly changes primary-state attenuation through $\mathbf{D}_p = (\mathbf{I}_N + \Delta t\, \mathbf{K}_p)^{-1}$; and k_o changes the persistence of spatial transport through $\mathbf{D}_o = (\mathbf{I}_N + \Delta t\, \mathbf{K}_o)^{-1}$. Despite these differences in impulse responses and pole distributions, all nine homogeneous, gradient, and smoothly heterogeneous configurations in Figs. A.3–A.5 satisfy $\rho(\mathbf{A}_{\mathrm{WO}}) < 1$. Thus, spatial heterogeneity alone does not imply an expansive linear recurrence.

To make the unit-circle boundary explicit, we additionally construct a mechanism-only case with a signed heterogeneous $k_p(x, y)$ field. Negative values of k_p lie outside the nonnegative attenuation parameterization used by WO-RNN and act locally as anti-damping. This intentionally expansive construction retains the same sparse spatial topology while allowing a small number of linear modes to cross the unit circle.

Relationship to the complete recurrent update. In the earlier reservoir construction, an FDTD transition matrix was subsequently mapped to a leaky-reservoir weight matrix through a relation of the form $\mathbf{W} = [\mathbf{A} - (1-\alpha)\mathbf{I}]/\alpha$ (Yuan et al., 2025). WO-RNN does not use this additional mapping: $\mathbf{A}_{\mathrm{WO}}$ is itself the linear propagation–damping operator inside the recurrence. It is nevertheless not the complete nonlinear WO-RNN transition. After applying Eq. (A.6), the projected input and pointwise nonlinearity are applied to the primary field:

$$p^{\ell} = f\big(\bar{p}^{\ell} + \mathrm{grid}(\mathbf{W}_{\mathrm{in}}\tilde{\mathbf{u}}_{\ell})\big), \qquad \mathbf{h}^{\ell} = [p^{\ell}, o_x^{\ell}, o_y^{\ell}]. \tag{A.12}$$

For $f(z) = \tanh(z)$,

$$|f(z)| \le 1, \qquad f'(z) = 1 - \tanh^2(z) \in (0, 1], \qquad f'(z) \to 0 \quad \text{as} \quad |z| \to \infty. \tag{A.13}$$

Thus, the primary field is pointwise bounded after activation, while saturation reduces its effective local gain at large amplitudes. The Jacobian of the complete recurrent map is therefore state dependent through f' rather than being the fixed matrix $\mathbf{A}_{\mathrm{WO}}$. Since the activation acts directly on p but not on the auxiliary velocity components, the pole distribution of $\mathbf{A}_{\mathrm{WO}}$ characterizes only the autonomous linear propagation–damping component and does not by itself establish global stability of the complete coupled nonlinear recurrence.

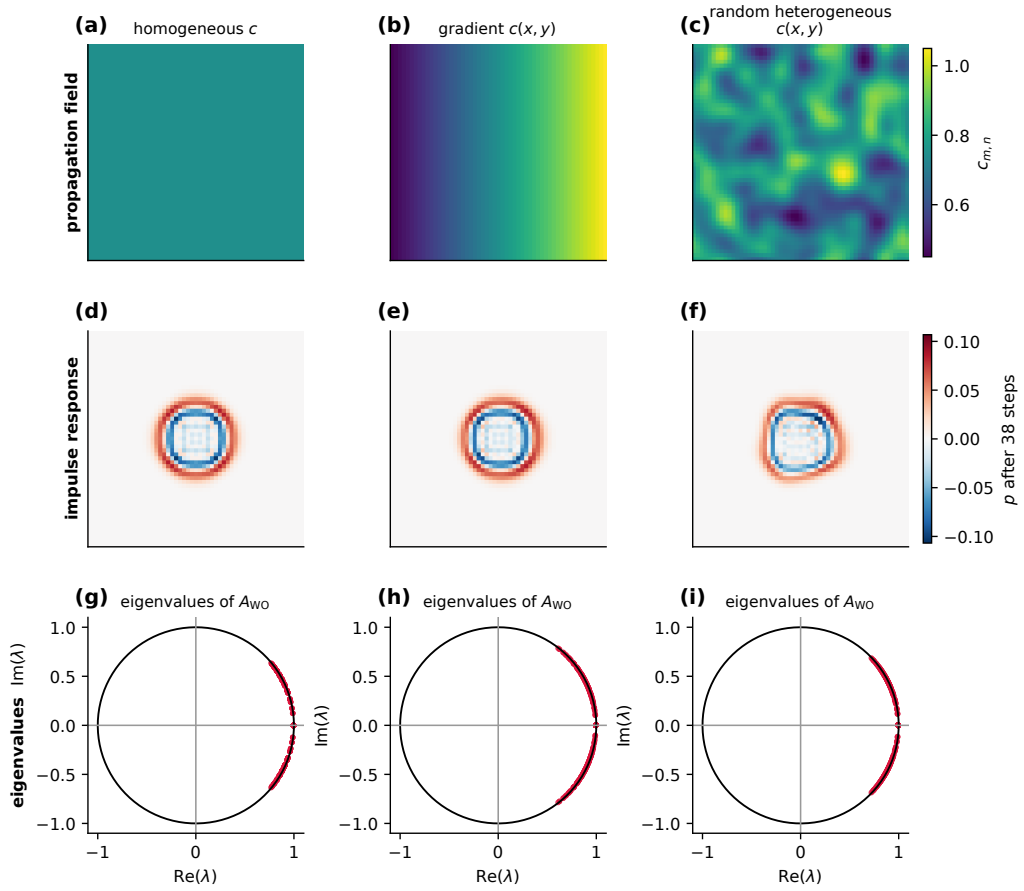


Figure A.3: Effect of spatially heterogeneous propagation coefficients in the autonomous linear WO-RNN operator. **(a–c)** Homogeneous, spatial-gradient, and smoothly random propagation fields $c(x, y)$. **(d–f)** Primary-field responses after 38 updates from the same centered impulse, with damping, time step, topology, and initial condition fixed. Spatial variation in c deforms the otherwise approximately symmetric wave front through location-dependent propagation scales. **(g–i)** Poles of $\mathbf{A}_{\mathrm{WO}}$; the black curve denotes the unit circle. The spectral radii are 0.995223, 0.995223, and 0.995223, respectively, with all poles inside the unit circle. Although the maximum pole modulus is unchanged in this example, heterogeneity redistributes the remaining oscillatory modes and changes the spatial impulse response.

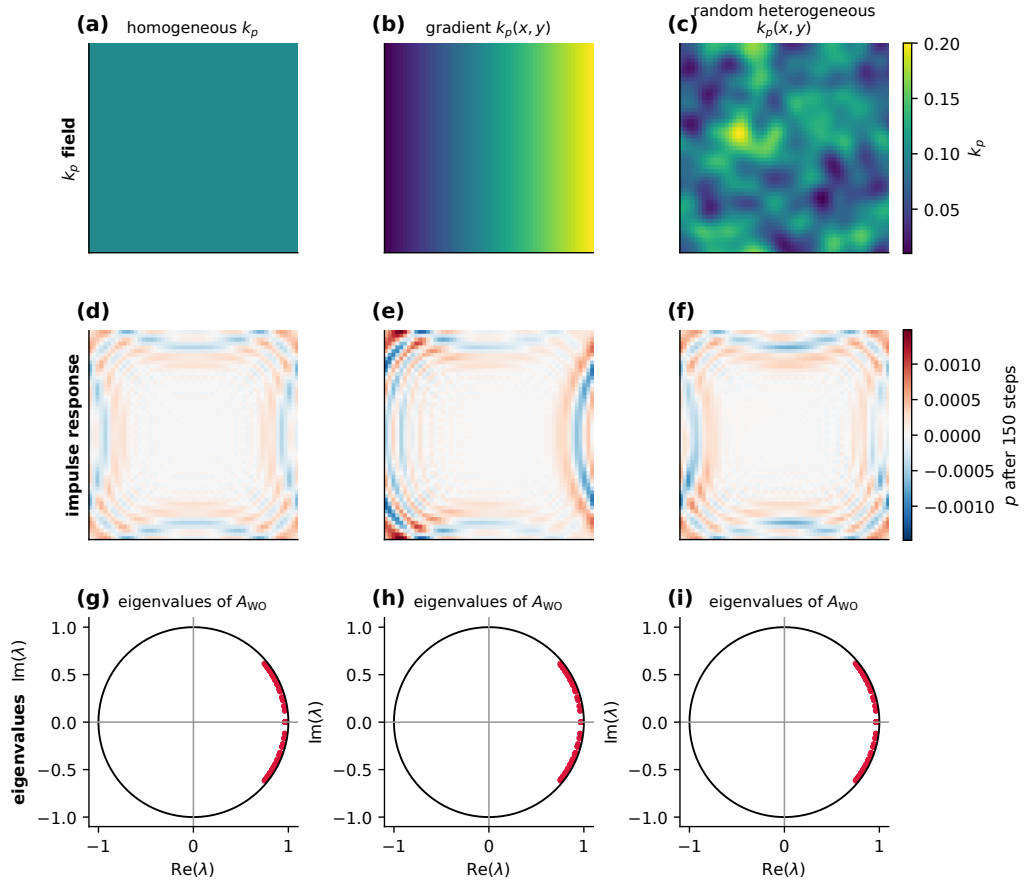


Figure A.4: Effect of spatially heterogeneous primary-field damping in the autonomous linear WO-RNN operator. **(a–c)** Homogeneous, spatial-gradient, and smoothly random $k_p(x, y)$ fields. **(d–f)** Primary-field responses after 150 updates from the same impulse, with $c(x, y)$, $k_o(x, y)$, time step, and topology fixed. **(g–i)** Poles of $\mathbf{A}_{\mathrm{WO}}$ and the unit circle. The spectral radii are 0.968992, 0.973195, and 0.973649, respectively, with all poles inside the unit circle. Spatial variation in k_p introduces location-dependent primary-state attenuation and modifies the associated persistence and decay modes without changing the coupling topology.

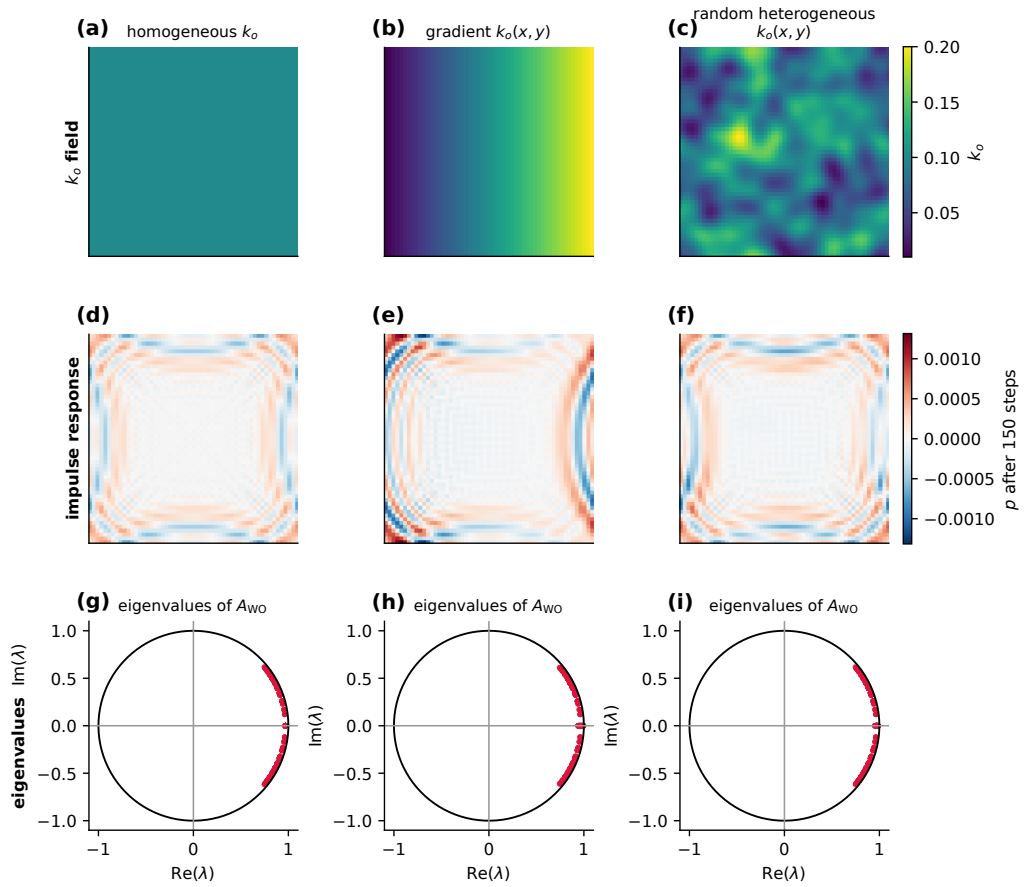


Figure A.5: Effect of spatially heterogeneous auxiliary-field damping in the autonomous linear WO-RNN operator. **(a–c)** Homogeneous, spatial-gradient, and smoothly random $k_o(x, y)$ fields. **(d–f)** Primary-field responses after 150 updates from the same impulse, with $c(x, y)$, $k_p(x, y)$, time step, and topology fixed. **(g–i)** Poles of $\mathbf{A}_{\mathrm{WO}}$ and the unit circle. The spectral radii are 0.968992, 0.996810, and 0.989266, respectively, with all poles inside the unit circle. Spatial variation in k_o changes auxiliary-velocity persistence and thereby alters the subsequent primary-field evolution.

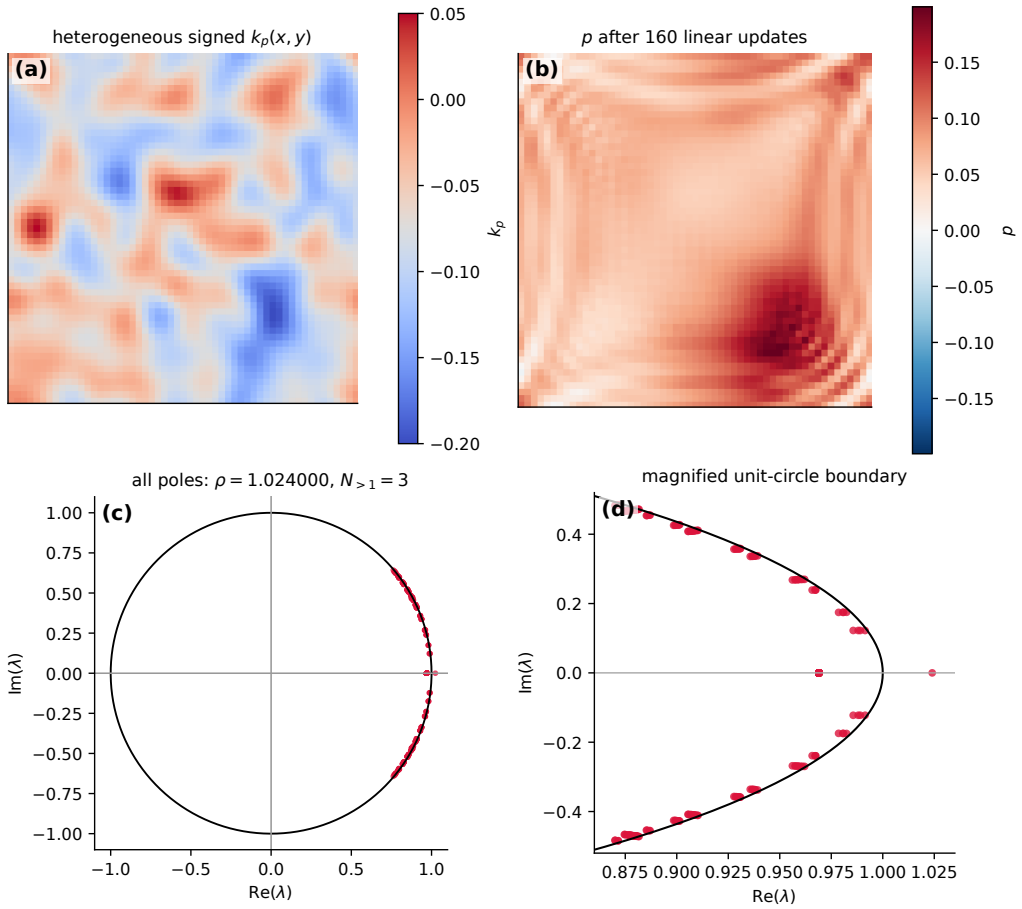


Figure A.6: Deliberately expansive linear regime induced by a signed heterogeneous primary-field damping coefficient. **(a)** A smoothly varying signed field $k_p(x, y) \in [-0.20, 0.05]$, containing both positive damping and negative local anti-damping. **(b)** Primary-field response after 160 autonomous linear updates from a centered impulse, with $c = 0.75$, $k_o = 0.10$, $\Delta t = 0.32$, and the periodic topology fixed. **(c)** Poles of the corresponding 12×12-grid operator $\mathbf{A}_{\mathrm{WO}}$. The spectral radius is $\rho(\mathbf{A}_{\mathrm{WO}}) = 1.024$, with three poles satisfying $|\lambda| > 1$. **(d)** Magnified view near the unit-circle boundary, exposing the poles that cross into the expansive region. This signed construction is a spectral stress test used to illustrate the linear stability boundary and is not part of the standard nonnegative WO-RNN parameterization.

In Figs. A.3–A.5, the linear operator is already asymptotically stable, so the bounded responses do not rely on nonlinear saturation compensating for expansive poles. Fig. A.6 instead deliberately introduces local anti-damping to expose the opposite regime: mildly expansive linear modes can amplify the state, while a bounded nonlinear activation makes the effective recurrent gain state dependent as the primary field approaches saturation. This illustrates a mechanism by which near-marginal or locally expansive linear modes can interact with nonlinear saturation, without implying that saturation alone guarantees stability of the full coupled system.

Taken together, the fixed sparse topology and spatially heterogeneous c, k_p, and k_o fields provide location-dependent propagation, multiple attenuation time scales, and distinct distributions of oscillatory, persistent, and decaying linear modes. Their composition with a bounded nonlinear activation further introduces state-dependent recurrent responses. The heterogeneous parameterization therefore enlarges the family of dynamics available to WO-RNN without changing its underlying neighbor graph; the analysis characterizes this dynamical capacity rather than proving that every learned heterogeneous configuration is more expressive or globally stable.

B Hyperparameters and Environment

This appendix reports dataset splits, training environment, and resource statistics for both sequence-classification and PDEBench forecasting experiments. Dataset splits for the sequence-classification tasks are summarized in Table B.1. Final sequence-classification results reported in the main text were obtained using the optimal WO-RNN hyperparameters identified in Table C.1.

All models were implemented in PyTorch. Sequence-classification experiments were trained using the Adam optimizer for 120 epochs and evaluated by accuracy (percent). All reported experiments, including task comparisons, grid search, ablation studies, and PDEBench runs, use the fixed random seed `torch.manual_seed(42)`. Accordingly, the numerical comparisons describe the evaluated setting and are not claims of statistical significance across repeated runs. All experiments were performed on a single NVIDIA GeForce RTX 4090 GPU with 24 GB of memory.

Table B.1: Training, validation, and test splits for the sequence classification datasets used in our experiments.

Dataset	Training Set	Validation Set	Test Set
sMNIST	57,000	3,000	10,000
nsCIFAR10	47,000	3,000	10,000
SHD	8,156	–	2,264

For the image domain, we employ the *noisy CIFAR-10* (nsCIFAR10) benchmark, which is derived from the standard CIFAR-10 dataset containing 60,000 color images across 10 classes. To simulate label imperfections commonly encountered in real-world scenarios, a fraction of the training labels are intentionally corrupted according to predefined noise schemes. In our experiments, we adopt the commonly used symmetric noise setting, where labels are flipped uniformly at random to other classes, while maintaining clean labels in the validation and test sets.

Table B.2 reports trainable parameter counts, peak GPU memory usage, and run time for the locally implemented models. Resource measurements not recorded with the reported configuration are marked as unavailable. For PDEBench tasks, these resource statistics correspond to the same PDEBench-style protocol and hyperparameter settings summarized in Appendix D. Published PDEBench-only baselines that were not instantiated locally are omitted from this resource table.

C Detailed ANOVA Results

C.1 Sensitivity to Integration and Propagation Scales

We analyze sensitivity to the time step $\Delta t \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$, propagation speed $c \in \{1, 40, 80\}$, damping coefficient $k \in \{0.01, 0.05, 0.1\}$ with $k^p = k^o$, and grid size $G \in \{16, 32, 64\}$ on sMNIST, nsCIFAR-10, and SHD. The spatial step is fixed at $\Delta x = 1$. The analysis characterizes

Table B.2: Number of parameters, GPU memory usage (MB), and run time (s) for selected models on all benchmark datasets.

Dataset	Model	#Params	GPU Memory (MB)	Run Time (s)
sMNIST	ESN	40,970	77.0	0.0038
	FDTDESN	–	–	–
	LSTM	295,434	49.9	0.0216
	GRU	222,218	49.7	0.0009
	LRU	1,198,858	148.6	0.0204
	S4D	706,570	757.6	0.0148
	Mamba	979,594	143.1	0.0028
	WO-RNN	172,070	375.4	0.0184
nsCIFAR-10	ESN	41,034	81.2	0.0091
	FDTDESN	–	–	–
	LSTM	670,538	149.4	0.0057
	GRU	503,882	145.9	0.0025
	LRU	1,206,602	335.4	0.0498
	S4D	714,314	1,643.6	0.0348
	Mamba	987,338	308.7	0.0038
	WO-RNN	254,026	854.1	0.0540
SHD	ESN	6,540	72.4	0.0131
	FDTDESN	744,896	–	–
	LSTM	1,679,296	270.8	0.0077
	GRU	742,284	198.2	0.0062
	LRU	1,464,256	854.6	0.1696
	S4D	971,968	1,819.8	0.0495
	Mamba	1,244,992	405.8	0.0043
	coRNN	128,576	159.8	0.0524
	NWM	188,116	170.0	0.1660
	wRNN	1,480,328	350.0	0.0617
	WO-RNN	743,496	378.6	0.1371
RD2D	ResNet	90,729,472	1,660.3	1.4736
	GNOT	7,319,298	11,346.8	1.8855
	Galerkin Transformer	796,994	5,075.7	0.9111
	OFormer	3,639,074	3,935.3	0.7375
	LRU	9,729,024	1,180.3	0.1035
	S4D	6,376,704	5,394.4	0.1958
	Mamba	6,922,752	871.2	0.0750
	WO-RNN	18,123,676	7,087.9	0.4440
SW2D	ResNet	57,127,296	1,141.1	0.1207
	GNOT	7,313,921	11,315.0	1.8849
	Galerkin Transformer	795,777	5,044.5	0.9117
	OFormer	3,637,921	3,919.5	0.7359
	LRU	5,061,696	599.0	0.0230
	S4D	5,460,288	9,856.0	0.3336
	Mamba	6,552,384	1,246.2	0.0866
	WO-RNN	18,040,495	8,577.2	0.4487
2D CFD	ResNet	57,221,376	888.3	0.0993
	GNOT	7,330,052	2,882.6	0.2174
	Galerkin Transformer	799,428	2,553.5	0.1978
	OFormer	3,641,380	1,987.3	0.1677
	LRU	78,078,592	624.8	0.0651
	S4D	78,477,184	628.2	0.0086
	Mamba	79,569,280	622.0	0.0111
	WO-RNN	18,044,278	2,691.3	0.0946

Table C.1: Comparison of Parameter Importance and Optimal Values Across sMNIST, noisy CIFAR-10, and SHD. For each task, the table lists model parameters in descending order of importance (ranked by their marginal effect on performance), along with the percentage of explained variance (marginal effect) attributed to each parameter and its corresponding optimal value. The ranking highlights which parameters most strongly influence performance within each task.

Task	Param	Effect (%)	Opt. Val.	Rank
sMNIST	Δt	**95.72**	0.1	1
	c	15.14	80	2
	grid	10.72	64	3
	k	0.74	0.05	4
Noisy CIFAR	Δt	**95.55**	0.1	1
	c	0.44	80	2
	grid	0.42	64	3
	k	0.02	0.1	4
SHD	Δt	**57.42**	0.005	1
	c	8.37	40	2
	grid	0.72	64	3
	k	0.06	0.05	4

the task-dependent response to these initialization parameters. As quantified below, visual sequence tasks are primarily sensitive to Δt, while SHD exhibits a stronger interaction between Δt and c.

C.2 Parameter Importance Analysis

To better understand how each parameter contributes to model performance across tasks, we performed both marginal (one-way) and multi-way ANOVA analyses. For SHD, the analysis is computed from 135 grid-search runs over Δt, c, grid size, and k. The results are summarized in Tables C.1 and C.2, and discussed in the following paragraphs.

We first present a one-way ANOVA where each parameter was varied independently to evaluate its marginal influence on the model's accuracy. As shown in Table C.1, the time step size Δt consistently dominates across all tasks, with marginal effects exceeding 90% in sMNIST and Noisy CIFAR and remaining the largest factor on SHD. The relative impact of c, grid size, and k is smaller in the marginal analysis, although c becomes important through interactions on SHD. These results suggest that Δt is the most sensitive individual parameter across regimes, while other parameters exert task-dependent secondary effects.

For SHD, Δt remains the dominant marginal factor, but its effect is weaker than in sMNIST and nsCIFAR10. The reduced marginal dominance is accompanied by a strong $\Delta t \times c$ interaction, indicating that temporal discretization and wave propagation speed must be jointly tuned for spike-based auditory inputs. The best single SHD grid-search configuration uses $\Delta t = 0.005$, $c = 80$, $k = 0.05$, and grid size 32, reaching 87.94% accuracy.

To further account for interactions between parameters, we then performed a multi-way ANOVA. Table C.2 reports the explained variance (η^2) for each main effect and pairwise interaction term. While Δt remains the dominant main effect, explaining 90.87% of variance in sMNIST and 95.48% in Noisy CIFAR, the SHD results show a different pattern.

Although Δt is still the largest SHD main effect, the $\Delta t \times c$ interaction explains 24.77% of the variance. For spike-based auditory sequences, the effective recurrent time scale is therefore governed by the joint setting of temporal integration and wave propagation.

The small values of η^2 for k in all tasks, as well as the weak interactions involving k, indicate that this parameter has limited influence on performance in the tested configurations. Grid size and c show increasing importance as task complexity and sequence length increase, reflecting their roles in controlling spatial resolution and wave propagation.

Table C.2: Explained Variance (η^2, in %) of Model Parameters and Their Interactions Across Tasks. Each row corresponds to a main parameter or a pairwise interaction (e.g., $\Delta t \times c$), and each column represents a specific task: sMNIST, noisy CIFAR-10 (nsCIFAR10), and SHD. The values indicate the proportion of total performance variance (in percentage) attributable to each parameter or interaction term, based on an ANOVA-style decomposition. Higher η^2 indicates a greater influence on model performance for that task.

Source	sMNIST	nsCIFAR10	SHD
Main Effects			
Δt	**90.87**	**95.48**	**57.42**
c	5.21	0.44	8.37
grid	0.78	0.43	0.72
k	0.45	0.02	0.06
Two-way			
$\Delta t \times c$	2.07	1.36	24.77
$\Delta t \times$ grid	1.18	0.52	0.77
$\Delta t \times k$	0.89	0.35	0.41
c $\times$ grid	—	—	0.08
c $\times$ k	—	—	0.04
grid $\times$ k	0.16	0.08	0.32

Although the marginal effect analysis and multi-way ANOVA largely agree on the dominant role of Δt, they provide complementary perspectives: the former highlights sensitivity when each parameter is varied in isolation, while the latter reflects statistical attribution after accounting for interactions.

D PDEBench Protocol and Benchmark Hyperparameters

This appendix documents the protocol alignment used for PDEBench-style comparisons and reports the training settings for the sequence-classification and PDE forecasting runs in Tables D.2 and D.3, respectively. Trainable parameter counts and resource usage are reported separately in Table B.2. PDEBench-reported U-Net, FNO, and PINN baselines are omitted from these local configuration and resource tables because their results are taken directly from the published benchmark.

Transformer baseline protocol. The official OFormer, Galerkin Transformer, and GNOT results use distinct operator-learning setups. OFormer evaluates Burgers, Darcy, and Navier–Stokes datasets inherited from the FNO setting,[1] with Navier–Stokes examples using encoder–decoder rollout, relative L_2 loss, curriculum training, and optional gradient loss. Galerkin Transformer uses FNO-style Burgers and Darcy data such as `burgers_data_R10.mat` and `piececonst_r421_N1024_smooth*.mat`,[2] and includes operator mappings from an initial condition to a terminal state or from a coefficient field to a solution field. GNOT targets irregular-mesh and multi-input parametric PDE datasets with the format `[X, Y, Theta, Inputs_funcs]`.[3] PDEBench provides a standardized forward-model pipeline for FNO, U-Net, and PINN, including shared metrics and configuration arguments such as `initial_step`.[4]

We therefore do not treat the official OFormer/Galerkin/GNOT numbers as PDEBench-comparable results. Instead, we adapt their architectures to the same PDEBench data split, normalization, optimizer, rollout procedure, and evaluation metrics used for WO-RNN. In our implementation, RD2D and SW2D use 10 input steps and 91 autoregressive prediction steps, while 2D CFD uses 5 input steps and 11 prediction steps.

Tables D.2 and D.3 summarize the hyperparameter configurations used for the sequence-classification and PDE forecasting experiments, respectively. In Table D.2, "n/a" denotes non-forecasting tasks. In Table D.3, "full" denotes training on the available forecasting trajectory under

[1] `https://github.com/BaratiLab/OFormer`
[2] `https://github.com/scaomath/galerkin-transformer#data`
[3] `https://github.com/thu-ml/GNOT#get-start`
[4] `https://github.com/pdebench/PDEBench#baseline-models`

Table D.1: Comparison of official operator-learning protocols with our common PDEBench-style evaluation.

Dimension	Official model protocols	Common PDEBench-style protocol
Data	Repository-specific; typically FNO-style benchmarks or custom GNOT datasets	PDEBench RD2D, SW2D, and 2D CFD
Split	Repository-specific; e.g., fixed first-N training samples and a held-out test set	Common train/validation/test split
I/O map	Task-specific operator maps, multi-input graphs, or sequence rollouts	Past frames $\rightarrow$ future frames
Rollout	One-shot prediction, decoder rollout, or graph/operator inference	Common autoregressive rollout
Loss	Relative L_2, gradient/H1 regularization, or GNOT-weighted losses	MSE with optional relative-L_2 terms
Metrics	Paper-specific relative errors and task metrics	PDEBench RMSE, nRMSE, CSV, max, boundary, and Fourier metrics
Pipeline	Repository-specific optimizer, scheduler, and normalization	Shared loader, normalization, optimizer, scheduler, and evaluator

the corresponding baseline implementation, and numeric train horizons denote the number of conditioning frames used before autoregressive rollout. Together with Table B.2, these tables separate computational cost from the settings under which the reported benchmark results were obtained.

Table D.2: Hyperparameter configurations for the sequence-classification benchmarks.

Dataset	Model	Epochs	Batch / test batch	LR	Train horizon	Pred. steps	Extra
sMNIST	ESN	120	120 / 1000	$1e{-4}$	n/a	n/a	H=256, modulated input
	FDTDESN	120	120 / 1000	$1e{-4}$	n/a	n/a	full $p + \mathbf{o}$, modulated
	LSTM	120	120 / 1000	$1e{-4}$	n/a	n/a	H=256, 1 layer
	GRU	120	120 / 1000	$1e{-4}$	n/a	n/a	H=256, 1 layer
	LRU	120	120 / 1000	$1e{-4}$	n/a	n/a	H=256, layers=1
	S4D	120	120 / 1000	$1e{-4}$	n/a	n/a	H=256, layers=1
	Mamba	120	120 / 1000	$1e{-4}$	n/a	n/a	H=256, layers=1
	WO-RNN	120	120 / 1000	$5e{-3}$	n/a	n/a	G=64×64, c=80, dt=0.1, k_p/k_o=0.05/0.05
nsCIFAR-10	ESN	120	120 / 1000	$1e{-3}$	n/a	n/a	patch=4, H=384, modulated
	FDTDESN	120	120 / 1000	$1e{-3}$	n/a	n/a	patch=4, full $p + \mathbf{o}$, modulated
	LSTM	120	120 / 1000	$1e{-3}$	n/a	n/a	patch=4, H=384, 1 layer
	GRU	120	120 / 1000	$1e{-3}$	n/a	n/a	patch=4, H=384, 1 layer
	LRU	120	120 / 1000	$1e{-3}$	n/a	n/a	patch=4, H=384, layers=1
	S4D	120	120 / 1000	$1e{-3}$	n/a	n/a	patch=4, H=384, layers=1
	Mamba	120	120 / 1000	$1e{-3}$	n/a	n/a	patch=4, H=384, layers=1
	WO-RNN	120	120 / 1000	$5e{-3}$	n/a	n/a	patch=4, G=64×64, c=80, dt=0.1, k_p/k_o=0.01/0.01
SHD	ESN	120	32 / 64	$5e{-3}$	n/a	n/a	H=256, no modulation
	FDTDESN	120	32 / 64	$5e{-3}$	n/a	n/a	full $p + \mathbf{o}$, modulated
	LSTM	120	32 / 64	$5e{-3}$	n/a	n/a	H=384, modulated
	GRU	120	32 / 64	$1e{-3}$	n/a	n/a	H=256, no modulation
	LRU	120	32 / 64	$5e{-3}$	n/a	n/a	H=384, layers=1, modulated
	S4D	120	32 / 64	$5e{-3}$	n/a	n/a	H=384, layers=1, modulated
	Mamba	120	32 / 64	$5e{-3}$	n/a	n/a	H=384, layers=1, modulated
	coRNN	120	32 / 64	$1e{-3}$	n/a	n/a	H=128, dt=0.01, γ/ϵ=2/2
	NWM	120	32 / 64	$1e{-3}$	n/a	n/a	H=256, dt=0.01, γ/ϵ=2/2
	wRNN	120	32 / 64	$1e{-3}$	n/a	n/a	H=256, ch=8, readout=max
	WO-RNN	120	32 / 64	$5e{-3}$	n/a	n/a	G=32×32, c=80, dt=0.005, k_p/k_o=0.05/0.05, modulated

Table D.3: Hyperparameter configurations for the PDE forecasting benchmarks.

Dataset	Model	Epochs	Batch / test batch	LR	Train horizon	Pred. steps	Extra
RD2D	ResNet	500	16 / 32	1e−3	full	91	wd=1e−4
	GNOT	100	4 / 8	1e−3	10	91	hidden=256, heads=8, depth=4
	Galerkin	100	4 / 8	1e−3	10	91	dim=256, heads=4, depth=4
	OFormer	100	2 / 4	1e−3	10	91	latent=384, heads=1, depth=5
	LRU	120	4 / 4	1e−3	full	91	patch=8, H=384, layers=4
	S4D	120	4 / 4	1e−3	full	91	patch=8, H=384, layers=2
	Mamba	120	4 / 4	1e−3	full	91	patch=8, H=384, layers=2
	WO-RNN	250	4 / 4	2e−4	rollout	91	scales=64,32,16; lc=4; c=1; dt=0.001
SW2D	ResNet	500	16 / 32	3e−4	10	91	relL2=0.05, clip=6.0
	GNOT	100	4 / 8	1e−3	10	91	hidden=256, heads=8, depth=4
	Galerkin	100	4 / 8	1e−3	10	91	dim=256, heads=4, depth=4
	OFormer	100	2 / 4	1e−3	10	91	latent=384, heads=1, depth=5
	LRU	120	4 / 4	1e−3	full	91	patch=8, H=384, layers=2
	S4D	120	4 / 4	1e−3	full	91	patch=8, H=384, layers=4
	Mamba	120	4 / 4	1e−3	full	91	patch=8, H=384, layers=4
	WO-RNN	500	5 / 5	1e−3	rollout	91	scales=64,32,16; lc=4; c=1; dt=0.001
2D CFD	ResNet	500	32 / 64	1e−4	5	11	cw=1,2,0.75,0.75; relL2=0.2; clip=6.0
	GNOT	100	4 / 8	1e−3	5	11	cw=1,2,0.75,0.75; relL2=0.2
	Galerkin	100	8 / 16	1e−3	5	11	cw=1,2,0.75,0.75; relL2=0.2
	OFormer	100	4 / 8	1e−3	5	11	cw=1,2,0.75,0.75; relL2=0.2
	LRU	120	4 / 4	1e−3	full	11	flat, H=384, layers=2
	S4D	120	4 / 4	1e−3	full	11	flat, H=384, layers=4
	Mamba	120	4 / 4	1e−3	full	11	flat, H=384, layers=4
	WO-RNN	500	32 / 32	1e−3	rollout	11	scales=64,32,16; lc=4; c=3; dt=0.005