NEURO-SPECTRAL ARCHITECTURES WITH TIME-DOMAIN DECOMPOSITION

Vitor Balestro[1,2], **Márcio Marques**[1], **Leonardo Mendonça**[1], **Leonardo Moreira**[1,3], **Christian Oliveira**[1], **Francisco Ganacim**[1], **Tiago Novello**[1], **Pavel Petrov**[1], **Daniel Yukimura**[1], **Lucas Nissenbaum**[1]

[1]Instituto de Matemática Pura e Aplicada (IMPA)
[2]Universidade Federal Fluminense (UFF)
[3]Universidade do Estado do Rio de Janeiro (UERJ)

ABSTRACT

Physics-informed neural networks (PINNs) have emerged as a promising approach for solving partial differential equations (PDEs) within the framework of scientific machine learning. However, despite their flexibility, PINNs often struggle with certain issues such as spectral bias and difficulties in capturing temporal causality. Recently, neuro-spectral architectures (NeuSA) were proposed as an alternative strategy that successfully deals with some of these problems. In this study, a modified D-NeuSA architecture is introduced. By contrast to the original one, D-NeuSA performs a decomposition of the time domain in order to increase robustness of the resulting neuro-spectral model. D-NeuSA consistently outperforms NeuSA and other baseline models in terms of both predictive accuracy and training efficiency across a wide range of problems. Moreover, D-NeuSA is able to produce reliable solutions in scenarios where NeuSA and other established PINN methods fail to converge.

1 INTRODUCTION

Physics-informed neural networks (PINNs), introduced by (Raissi et al., 2019), leverage machine learning techniques to solve direct and inverse problems involving partial differential equations. Automatic differentiation algorithms (Baydin et al., 2018) allows us to optimize the neural network parameters to find a solution that most suits the differential equation. A remarkable advantage of this approach is the flexibility to choose between an unsupervised training scheme and a scheme that can incorporate real-world data (Kissas et al., 2020; Yazdani & Tahani, 2024). This leads to a wide range of applications for PINNs in different areas such as wave propagation (Ito et al., 2025; Marques et al., 2025; Schoder, 2025), biomedicine (Buoso et al., 2021; Herrero Martin et al., 2022; Lin et al., 2026) and material science (Zheng et al., 2022; Zhang et al., 2022; Rezaei et al., 2024).

Despite promising results, physics-informed neural networks are known to have some limitations, including issues with high frequency components of the solution (*spectral bias*) (Wang et al., 2022; Rahaman et al., 2019) and difficulties in capturing time causality in evolutionary problems (Krishnapriyan et al., 2021; Penwarden et al., 2023; Wang et al., 2024). In Bizzi et al. (2025b) the so-called *neuro-spectral architectures* (NeuSAs) were introduced as an alternative to overcome these shortcomings. The main idea behind NeuSA is to represent the vector field governing the ODE system induced by the spectral decomposition of the underlying PDE through a neural network, and to learn its dynamics using the Neural ODE framework (Chen et al., 2018).

However, despite outperforming PINNs in benchmark problems from different areas of physics (Bizzi et al., 2025b), NeuSA still presents important limitations. In particular, it may struggle with equations exhibiting highly complex dynamics as shown in Figure 1 or with problems whose spectral decomposition leads to stiff ODE systems (Stoer et al., 1980; Schiesser, 2012; 2014; Cox & Matthews, 2002). Because NeuSA relies on explicit time integration, these cases become challenging: although implicit methods are more appropriate for stiff problems, coupling them with automatic differentiation typically leads to prohibitively high computational cost.

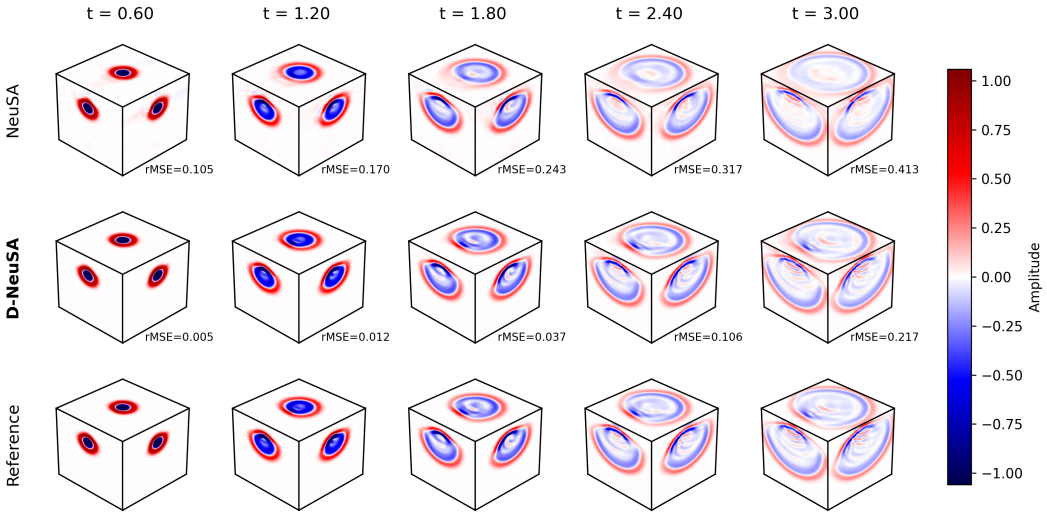


Figure 1: We introduce **D-NeuSA**, a neuro-spectral architecture that explicitly decomposes the temporal domain, enabling localized learning of distinct dynamical regimes. This design improves the modeling of systems exhibiting regime changes or strong temporal heterogeneity, since independent neural networks specialize in different temporal subdomains. The figure presents a visual comparison between the reference solution, the standard NeuSA, and D-NeuSA for wave propagation in the real-world–based Utah FORGE model. Spatial variations in propagation velocity generate strong reflections, and D-NeuSA more accurately captures these features than the standard NeuSA.

To address the aforementioned challenges, we introduce *neuro-spectral architectures with time-domain decomposition* (D-NeuSA). The key idea is to partition the temporal domain into multiple subintervals and to cascade independent neural networks, each specializing in modeling of the vector field corresponding to the differential equation within one of the subintervals interval. This allows us to fine-tune the numerical integration process, thus mitigating the stiffness issue for ODEs formed by the spectral decomposition of a PDE. This also separates the problem into smaller problems, which makes them simpler for the networks to model. In a series of computational experiments, we show that this technique is a promising generalization of NeuSA in terms of overall better prediction accuracy and shorter training time.

1.1 Related work

Neuro-Spectral Architectures. This work builds off of NeuSA, to be described in Section 2, which has achieved strong results in time-dependent PDEs where most PINN architectures struggle, notably waves in heterogeneous domains (Bizzi et al., 2025b), by combining the Neural ODE's causal structure (Chen et al., 2018) with a frequency-domain representation similar to successful approaches in PINNs such as First-Layer Sine (FLS) (Wong et al., 2024) and Fourier features (Ding et al., 2025). D-NeuSA differs from NeuSA's original formulation mainly in the sequential time-decomposition strategy (see Section 3).

In the context of PINNs, **domain decomposition** has been introduced as a systematic approach to mitigate the optimization and expressivity limitations encountered when training a single network over large domains (Jagtap & Karniadakis, 2020). Several modifications have been proposed to regularize this framework, including penalty-based interface conditions (Jagtap et al., 2020) and hard enforcement of continuity constraints (Roy & Castonguay, 2024).

Time-decomposition strategies. There have been various attempts at leveraging time-domain segmentation in order to combat PINN's architectural lack of causality and improve overall accuracy. Ding et al. (2025) proposes training the same network progressively on increasing time intervals starting from $t = 0$, in order to tackle the wave equation in 2D seismic domains. Time marching, on the other hand, entails splitting the time domain into a certain number of intervals and sequen-

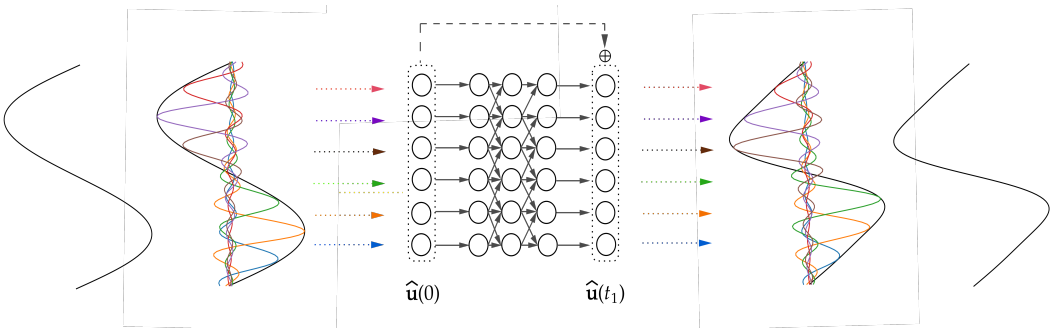


Figure 2: Neuro Spectral Architectures uses a spectral basis of the initial condition to form a new representation representation, through which a NeuralODE can propagate the vector field represented by a neural network to find the solution of the PDE at future times.

tially training a different PINN for each one, using one network's output as the initial condition for the next interval, leading to significantly improved solutions to wave-like problems (Wight & Zhao, 2020; Bihlo & Popovych, 2022; Penwarden et al., 2023). D-NeuSA builds upon this idea, with the added advantage that NeuSA automatically enforces initial conditions for each sub-network, thus ensuring continuity of the solution.

2 Neuro-spectral architectures

We consider an initial-boundary-value problem of the form

$$\begin{aligned} u_t - \mathcal{F}[u] = 0\,, \quad \mathbf{x} \in \Omega, \quad t \in [0, T]\,, \\ u(\mathbf{x}, 0) = g(\mathbf{x})\,, \quad \mathbf{x} \in \Omega\,, \\ u(\mathbf{x}, t) = h(\mathbf{x}, t)\,, \quad \mathbf{x} \in \partial\Omega\,, \quad t \in [0, T]\,, \end{aligned}$$

where $u\colon \Omega \times [0, T] \to \mathbb{R}^d$ is the unknown and $\mathcal{F}$ is a (possibly nonlinear operator which involves only spatial derivatives of u.

Neuro-spectral architectures were proposed to tackle such problems (Bizzi et al., 2025b). A spectral decomposition with respect to a fixed orthogonal basis turns a PDE into a system of initial value ODE problems with the form:

$$\begin{aligned} \hat{\mathbf{u}}_t &= \mathbf{F}(\hat{\mathbf{u}})\,, \\ \hat{\mathbf{u}}(0) &= \hat{\mathbf{u}}_0 \end{aligned} \tag{1}$$

where $\mathbf{F}\colon \mathbb{R}^n \to \mathbb{R}^n$ is a vector field, $\hat{\mathbf{u}}$ denotes the spectral representation of u, and $\hat{\mathbf{u}}_0$ is the spectral representation of the initial condition u_0, as depicted in Figure 2. The boundary conditions are thus enforced by the choice of the spectral basis, leading to an architecture that respects causality by construction.

NeuSA proposes to adopt a neural network as an implicit representation of the vector field $\mathbf{F}$ determining the flow of the ODE system in Equation (1). Let $\mathbf{N}_{\boldsymbol{\theta}}$ represent a neural network parametrized by a set of parameters $\boldsymbol{\theta}$ that approximates $\mathbf{F}$. In order to train $\mathbf{N}_{\boldsymbol{\theta}}$, we use NeuralODE (Chen et al., 2018), which is a tool that allows us to solve ODEs that are parametrized by neural networks, with many applications spanning, e.g., fluid dynamics (Donnelly et al., 2024; Molina Catricheo et al., 2024), materials science (Rezaei et al., 2024; Zhang et al., 2022; Zheng et al., 2022) and time-series analysis (Kidger et al., 2020; Oh et al., 2025), and image morphing (Bizzi et al., 2025a). Specifically, we apply it to obtain the solution $\hat{\mathbf{u}}_{\boldsymbol{\theta}}$ of the ODE

$$\begin{aligned} \hat{\mathbf{u}}_t &= \mathbf{N}_{\boldsymbol{\theta}}(\hat{\mathbf{u}})\,, \\ \hat{\mathbf{u}}(0) &= \hat{\mathbf{u}}_0 \end{aligned} \tag{2}$$

in terms of $\boldsymbol{\theta}$ using some classical numerical method. Then, we use the inverse spectral decomposition to reconstruct the solution $u_{\boldsymbol{\theta}}$ of the PDE associated to (2), and the parameters $\boldsymbol{\theta}$ are learned by imposing the physical loss on $u_{\boldsymbol{\theta}}$. As with PINNs, NeuSA is capable of incorporating known data through an additional loss term.

3 TIME-DOMAIN DECOMPOSITION

When training NeuSA, the neural network $\mathbf{N}_{\boldsymbol{\theta}}$ has a limited amount of information to learn the objective vector field. Namely, we enforce the physical constraint only along a single latent trajectory obtained as a solution of the ODE (1) with respect to the parameters at each training step.

Moreover, even if time sampling along the trajectory provides enough information to conduct the training, we still have to consider the representative power of the adopted neural network architecture. This is especially important when dealing with complicated high-dimensional ODEs, e.g., emerging from a spectral decomposition of a PDE. Indeed, their dimension is related to the number of the spectral basis elements. As a limitation of the method, we note the computational overhead might become unbearable if we add too many hidden layers.

It is also important to mention that NeuSA heavily relies on the numerical ODEs solver. For the sake of computational efficiency, we employ a fourth-order Runge-Kutta method, which often fails to resolve stiff problems.

To address these challenges, we propose splitting the time domain into smaller segments, and training an independent neural network for each subdomain. This time-domain decomposition improves the effective representational capacity of neural models by replacing a single global approximation with multiple localized ones. In particular, this approach offers the following advantages:

- Learning the vector field over shorter trajectory segments reduces the temporal complexity of the approximation problem and typically involves less dynamical variability, allowing each sub-network to focus on a small portion of the dynamics.
- Independent sub-networks naturally specialize in distinct subdomains of the high-dimensional vector field, which is especially beneficial for systems exhibiting regime-switching behavior, such as problems with variable coefficients e.g. reflection phenomena.
- A smaller domain allows us to achieve convergence in a few training steps. We can then adopt finer time-domain discretizations. This is suitable to deal with stiff equations, which is frequently the result of PDE spectral decompositions.

3.1 ALGORITHM

In a nutshell, D-NeuSA consists in decomposing the time domain into sub-intervals and sequentially training an independent NeuSA model for each one of them. After training each model, we integrate the vector field it produces from start to end of its respective time subdomain, passing the final state obtained as the initial condition of the next subdomain. Finally, we proceed iteratively until all subdomains are done.

Precisely, let the time domain be decomposed into p subdomains, each consisting of K time steps:

$$\underbrace{t_0, t_1, \ldots, t_{K-1}}_{\text{first subdomain}}, \underbrace{t_K, t_{K+1}, \ldots, t_{2K-1}}_{\text{second subdomain}}, \ldots, t_{(p-1)K-1}, \underbrace{t_{(p-1)K}, \ldots, t_{pK-1}, t_{pK}}_{p-\text{th subdomain}}.$$

We train the first neural network $\mathbf{N}_{\boldsymbol{\theta}_0}$ for $t_0, \ldots, t_K$. From the solution $\hat{\mathbf{u}}_{\boldsymbol{\theta}_0}$ provided by this neural network after training, we compute $\hat{\mathbf{u}}_{\boldsymbol{\theta}_0}(t_K)$ and use this as initial condition in the training of the second neural network $\mathbf{N}_{\boldsymbol{\theta}_1}$ for $t_K, \ldots, t_{2K}$. To avoid any possible redundancy, in the inference stage we remove t_K from the domain of $\mathbf{N}_{\boldsymbol{\theta}_0}$. Proceeding iteratively, we get the solution for all time steps. The algorithm is depicted in Figure 3.

When we look carefully at the algorithm, we notice an immediate trade-off: to train each subdomain but the first, we use a somewhat "corrupted" initial condition. In fact, these initial conditions are not exact, since each of them is obtained from the training of the respective previous subdomain. Thus, we "accept" these inaccuracies to get all the benefits described above. Our empirical results show that this option pays off.

It is worth mentioning that in Neural ODEs, a central modeling choice is whether parameters are shared or unshared across depth or time. Parameter sharing corresponds to repeatedly applying the same transformation or vector field, introducing a strong inductive bias and reduced computational cost, whereas unshared parameters allow the dynamics to vary across layers or time, increasing

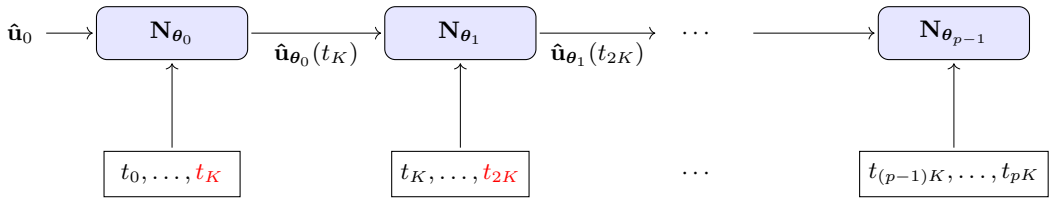


Figure 3: D-NeuSA architecture. Each NeuSA network learns the dynamics on a short time subdomain. The terminal state of each subdomain provides the initial condition for the next.

representational capacity. From this perspective, NeuSA adopts a shared-parameter formulation, enforcing a single vector field representation across the domain, while D-NeuSA relaxes this constraint by allowing depth-dependent or subdomain-dependent parameters.

4 Experiments and Results

Our experiments involve training D-NeuSA models for several forward initial/boundary value problems. The produced solutions are compared to reference solutions (all obtained by pseudo-spectral methods), and we adopt the standard *relative rooted mean squared error* (or *relative L_2 error*) as accuracy metric, as in Zhongkai et al. (2024). Our baseline models are the standard NeuSA (Bizzi et al., 2025b), the original PINNs (Raissi et al., 2019), QRes (Bu & Karpatne, 2021) and FLS Wong et al. (2024). From now on, we will refer to the three last models as *MLP-based models*, since they represent the solution of the PDE directly as a multilayer perceptron. The adopted configurations of the MLP-based models are the same as in (Wu et al., 2024) and (Zhao et al., 2023). We also implemented versions of the MLP-based models with direct time-domain decomposition, as in D-NeuSA. From now on, we refer to these models as *D-MLP-based models*. All models were trained Adam optimization method (Kingma & Ba, 2015) until they reach convergence and/or stability.

Each sub-network in D-NeuSA has the same architecture adopted for NeuSA. As a consequence, D-NeuSA has a larger number of parameters as NeuSA. This does not cause any performance overhead, since the overall number of operations is essentially the same for training both models (the same holds for inference). This is also true for the MLP-based models with time-domain decomposition.

Experiments were conducted using an NVIDIA RTX 4090 GPU (24GB VRAM). The quantitative results are displayed in Table 1.

4.1 Allen-Cahn equation

The initial/boundary value problem considered is the following:

$$u_t - 0.0001u_{xx} - 5u + 5u^3 = 0 \tag{3}$$

for $(x,t) \in [-1,1] \times [0,1]$, with initial condition $u(x,0) = x^2\cos(\pi x)$ for $x \in [-1,1]$, and boundary conditions $u(-1,t) = u(1,t)$ and $u_x(-1,t) = u_x(1,t)$ for $t \in [0,1]$. This is a common problem in the PINNs literature (Raissi et al., 2019; Roy & Castonguay, 2024).

For the Allen-Cahn equation, the NeuSA model was trained for 1000 Adam steps. In the D-NeuSA model, each sub-network was trained for 100 Adam steps. We adopted a learning rate of 0.02 for both cases. The MLP-based models were trained for 20000 Adam steps, with learning rate of 0.001 in a 201×201 (space, time) grid. Each subdomain in the D-MLP-based models was trained with the same configuration.

Because the associated ODE is stiff, we needed a small time step. Thus, we considered 1001 points in the time domain. To avoid each sub-network being responsible to many time points, we divided the time domain in 100 subdomains. Therefore, in accordance with the other equations, each sub-network learns a vector field which integrates 10 time steps.

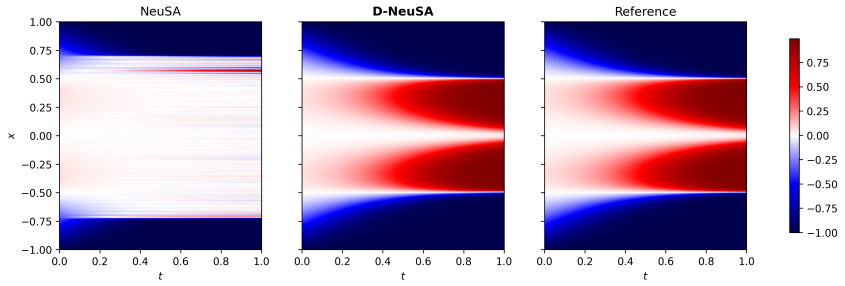


Figure 4: NeuSA, D-NeuSA and reference solutions for the Allen-Cahn equation. D-NeuSA accurately reconstructs the spatialtemporal solution, particularly in the more challenging regions.

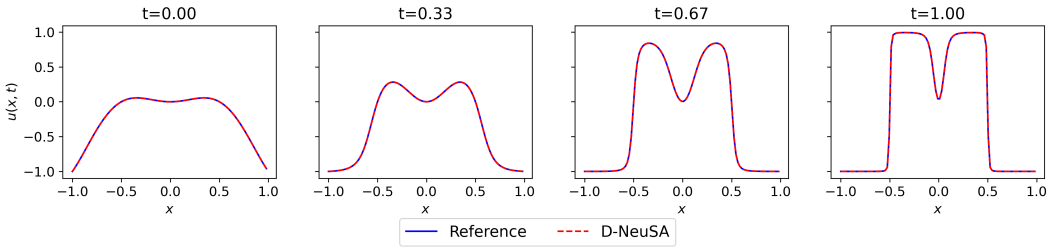


Figure 5: Time snapshots of D-NeuSA predicted solution for the Allen-Cahn equation. Observe that they are visually indistinguishable from the corresponding reference solution snapshots.

4.2 BURGERS EQUATION

We solve the equation

$$u_t + uu_x - \nu u_{xx} = 0 \tag{4}$$

for $(x, t) \in [-1, 1] \times [0, 1]$, with initial condition $u(x, 0) = -\sin(\pi x)$ for $x \in [-1, 1]$, boundary condition $u(-1, t) = u(1, t) = 0$ for $t \in [0, 1]$, and where $\nu = 0.01/\pi$.

NeuSA model was trained for 200 Adam steps, with a learning rate of 0.001. On the other hand, each sub-network in D-NeuSA was trained for 100 Adam steps, and the learning rate adopted was also 0.001 (in both NeuSA and D-NeuSA, we observed numerical instability when trying to use larger learning rates). We have used 1001 time steps and 501 points in the spatial grid for both NeuSA and D-NeuSA (we noticed some numerical instability for smaller grids). The time domain was divided into 100 subdomains for the D-NeuSA model, and each sub-model was trained for 100 Adam steps. PINN, FLS and QRes models were trained for 20000 Adam steps with learning rate of 0.001 over a 201×201 grid (just as in the case of the Allen-Cahn equation). Their corresponding versions with time-domain decomposition were trained with the same configuration (for each subdomain).

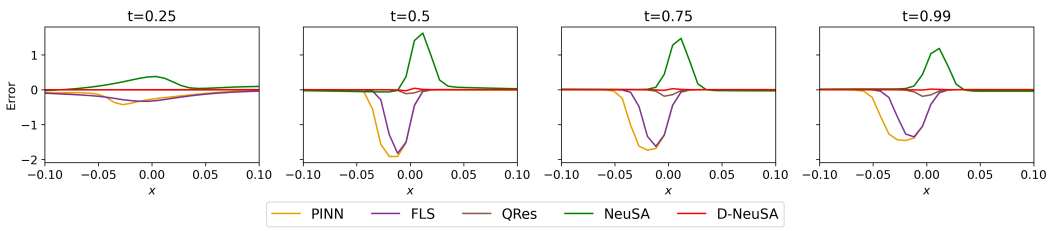


Figure 6: Snapshots of the error of each model in the shock region of Burgers equation (around $x = 0$). D-NeuSA is more accurate in this challenging part of the evolution than the other models.

4.3 WAVE PROPAGATION IN A THREE-LAYERED MEDIUM

We consider an initial-boundary value problem for the wave equation with two spatial variables:

$$u_{tt} - c^2 \Delta u = 0 \quad \text{for} \quad (x, y, t) \in [-2, 2] \times [-2, 2] \times [0, 2], \tag{5}$$

with the Cauchy data $u(x, y, 0) = \exp\left(-(x^2 + y^2)/2\sigma^2\right)$, $\sigma = 0.1$, and $u_t(x, y, 0) = 0$, and the sound speed distribution c corresponding to the three-layered medium model used in (Bizzi et al., 2025b) (see Figure 8a). Although we impose homogeneous Neumann conditions on the boundary of the spatial domain, our solution does not actually reach it for the considered time interval (hence our solution also coincides with that of a Cauchy problem on an infinite domain). A different approach was taken in (Marques et al., 2025), where absorbing boundary conditions were used in the framework of a PINN-based solver of the wave equation.

In NeuSA and D-NeuSA models, a cosine basis is adopted, and the training is performed over an extended spatial domain. Thus, in practice, a homogeneous Neumann condition is enforced as a hard constraint in the extended domain. For this equation, NeuSA was trained for 2000 Adam steps, whereas each sub-network in D-NeuSA was trained for 200 Adam steps (the time domain was divided into 20 subdomains). In both cases, the adopted learning rate was 0.01. On both NeuSA and D-NeuSA models, the spatial grid is 101x101 (201x201 on the extended domain), and we considered 201 time steps. The MLP-based models were trained for 20000 Adam steps with a learning rate of 0.001. At each training step, we uniformly sample 2000 points for the initial condition, 1000 points for the boundary condition and 20000 collocation points for the PDE residue (because of memory constraints, training with the full grid is not possible in the two-dimensional case). In the case of the D-MLP-based models, these numbers are 200, 500 and 5000, respectively, for each subdomain. Each sub-model was trained for 5000 Adam steps.

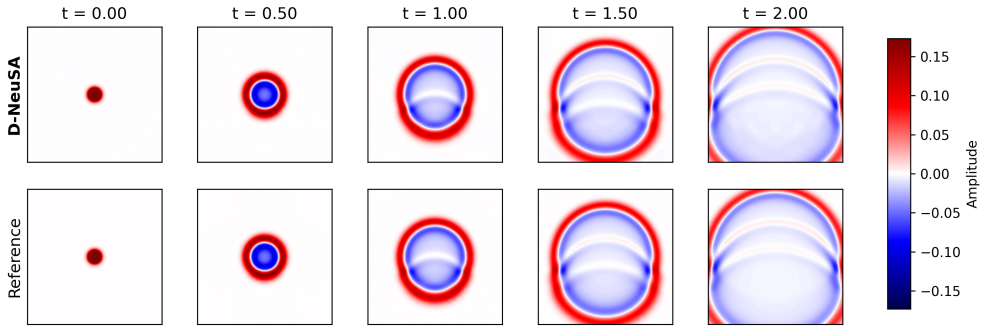


Figure 7: A comparison between the D-NeuSA predicted solution and the reference solution for the two-dimensional wave equation in a three-layered medium.

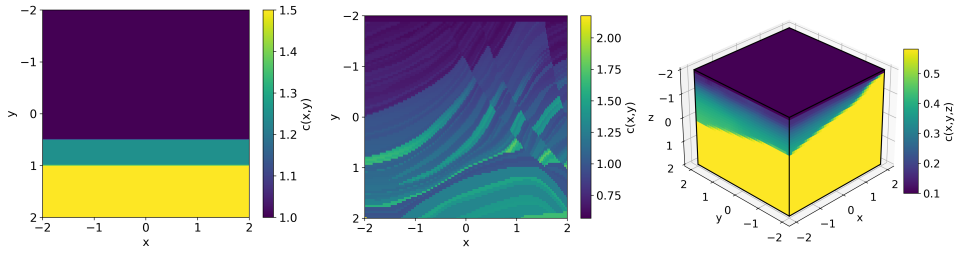
4.4 WAVE PROPAGATION IN A MARMOUSI MEDIUM

The setup for this experiment is similar to the wave equation in a three-layered medium. The only difference is that the center of the Gaussian in the initial condition is $(0, -0.5)$, instead of $(0, 0)$.

Here we also consider an extension of the domain to simulate an infinite propagation domain. The Marmousi medium (Brougois et al., 1990) that we adopted in this experiment is widely regarded in the geophysics literature (Sajeva et al., 2016; Favorskaya et al., 2018) and depicted in Figure 8b.

We adopted a 201x201 spatial grid (401x401 for the extended domain), and 201 time steps on both NeuSA and D-NeuSA models. NeuSA model was trained for 2000 Adam steps, and each sub-model of D-NeuSA was trained by this same number of steps (we decomposed the time domain in 40 subdomains). In this case, D-NeuSA does not present any advantage in terms of training time, but the accuracy is much better, as the table below shows. Both models were trained with learning rate 0.01. The training setup of PINN, FLS and QRes models, and of their respective counterparts with time-domain decomposition, for this problem is exactly the same as for the three-layered medium problem described above.

Figure 9 shows the comparison between the D-NeuSA predicted solution and the reference solution.



(a) Three layers (b) Marmousi (c) Utah FORGE

Figure 8: (a) Three-layered domain considered in Bizzi et al. (2025b). (b) Marmousi (Brougois et al., 1990), a two-dimensional model based on realistic geology, (c) Utah FORGE (Vasco & Chan, 2022), a three-dimensional asymmetrical velocity model for a real-world site.

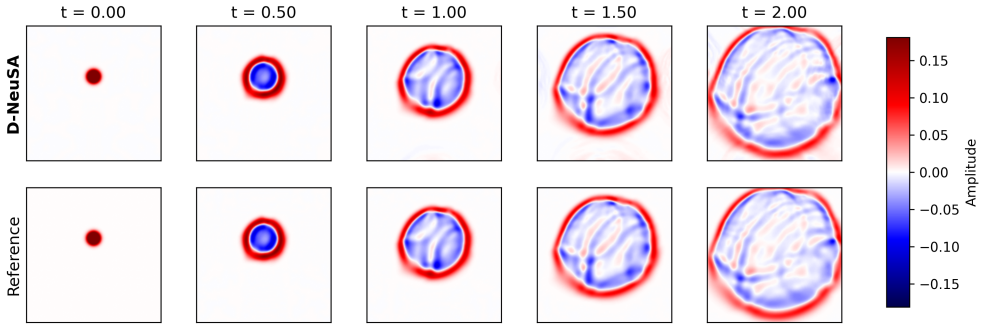


Figure 9: A comparison between the D-NeuSA predicted solution and the reference solution for the two-dimensional wave equation in the Marmousi medium.

4.5 Wave propagation in a realistic 3D environment

We have used NeuSA, D-NeuSA and the MLP-based models to simulate wave propagation in a medium known as *Utah FORGE*; see (Vasco & Chan, 2022). Since we are dealing with an acoustic equation, we considered the compressed velocity field.

To simulate wave propagation, the adopted spatial domain was $[-2, 2] \times [-2, 2] \times [-2, 2]$ (represented in a $41 \times 41 \times 41$ grid), and the time domain was set to be $[0, 3]$. The initial condition is a scaled Gaussian centered at $(-1, 0, 0)$ (thus near the interface of main variation of the velocity; see Figure 8c), and with standard deviation $\sigma = 0.05$. For the time domain, we considered a discretization of 301 points. To achieve stability, we normalize the velocity by dividing it by 10. We imposed homogeneous Neumann boundary conditions by adopting a cosine basis for the spectral decomposition.

Both NeuSA and D-NeuSA models were trained with learning rate of 0.01 for 2000 Adam steps; for D-NeuSA, each sub-model was trained for 2000 steps. PINN, FLS, and QRes models were trained for 5000 Adam steps with learning rate of 0.001. Due to memory constraints when training on the full grid, we employed a uniform sampling strategy at each iteration: 10000 points for the initial condition, 2000 for the boundary condition, and 70000 collocation points for the PDE residue. For the D-MLP-based models, these numbers are respectively 4000, 1000, and 20000. Each sub-model was trained for 2000 Adam steps.

In Figure 1, we plot time snapshots of the NeuSA and D-NeuSA predicted solutions together with the reference solution in orthogonal slices intersecting in the source position. It is visually clear that the D-NeuSA solution is much closer to the reference solution.

Table 1: **Quantitative evaluation on PDE benchmarks.** Results are averaged over five runs with different random seeds. We report relative RMSE (rRMSE) and training time (TT, seconds). Models prefixed with "D-" use time-domain decomposition. D-NeuSA (ours) consistently achieves lower reconstruction error while maintaining competitive training cost.

Model	Allen-Cahn		Burgers		Three layers		Marmousi		Utah FORGE	
	rRMSE	TT	rRMSE	TT	rRMSE	TT	rRMSE	TT	rRMSE	TT
PINN	0.9938	1202.65	0.1803	617.17	0.9197	1274.85	1.0715	1316.12	1.1323	**1593.70**
QRes	0.3162	1660.23	0.0237	824.68	0.1983	1616.40	0.5052	1657.62	0.4042	2253.17
FLS	0.8989	1244.96	0.1989	625.84	0.7981	1305.31	1.0957	1346.35	1.0451	1615.79
D-PINN	1.1997	1669.88	0.6701	1552.22	1.0057	1606.48	1.0449	2103.10	1.7221	1927.29
D-QRes	0.9775	2833.55	0.4233	2468.10	1.0622	1781.83	1.0977	1874.69	1.0344	2328.46
D-FLS	1.1919	1692.94	0.6612	1530.57	1.0121	1613.94	1.0118	1698.20	1.1059	1868.36
NeuSA	0.7544	**88.74**	0.2372	227.78	0.1254	251.18	0.1682	717.88	0.2017	2946.39
D-NeuSA	**0.0007**	89.53	**0.0017**	**90.76**	**0.0200**	**51.60**	**0.0398**	**682.02**	**0.0790**	1864.40

5 Results and analysis

D-NeuSA is particularly efficient in the case of Allen-Cahn equation as compared to the other methods because we may adopt a small time step (namely, 10^{-3}) and to train each sub-network for only a few steps. This makes the Runge-Kutta scheme efficient at a low computational cost. For the same reason, in the case of Burgers equation D-NeuSA model is at least one order of magnitude better than the MLP-based models in both predictive accuracy and training time. It is worth noticing that these models perform better than NeuSA in this equation.

For the 2D wave equation with a three-layered medium, D-NeuSA attains a slightly lower error than NeuSA while requiring approximately one order of magnitude less training time. In the Marmousi case, both methods exhibit comparable training times, but D-NeuSA achieves a substantially better accuracy. At the same time, MLP-based models were unable to produce satisfactory solution for this problem in comparable time.

For the 3D wave equation in the Utah FORGE scenario only D-NeuSA was able to compute a reasonably accurate solution, while all other considered models failed. This suggests that, due to effective reduction of the learning burden for each sub-network, time-domain decomposition becomes increasingly beneficial as the medium complexity increases.

The results highlight a fundamental trade-off associated with enforcing causality through time-domain decomposition in PINNs. Convergence of the training loss within a given temporal subdomain is not, by itself, indicative of accuracy if the solution in the preceding subdomains has not yet converged. Because each subdomain inherits its initial conditions from the previous one, errors introduced early in the time integration propagate forward in time and may be amplified, irrespective of the apparent local convergence. Consequently, achieving accurate convergence in the initial subdomains is crucial for ensuring the quality of the global solution, a behavior that closely mirrors classical time-marching numerical schemes.

The straightforward time-domain decomposition implemented for MLP-based models, where PDEs and initial-condition losses are enforced simultaneously, exhibits slow convergence (Table 1). While accuracy could be improved with more iterations, the training time, even under our current setup, already exceeds that of D-NeuSA by orders of magnitude. For instance, while hard-enforcing initial conditions at subdomain interfaces can stabilize training and mitigate error propagation, the resulting computational cost remains significantly higher than our approach (Roy & Castonguay, 2024).

6 Conclusion

In this study, we introduced D-NeuSA, an improved version of NeuSA architecture, in which the time domain is partitioned into subintervals, and an independent neural network is trained to learn the evolution along each subinterval. Such partitioning results in an easier learning task for each of them, and therefore large numerical errors emerging from training of a single neural network for too many time steps are avoided.

In our experiments, D-NeuSA outperformed NeuSA in various benchmark scenarios involving different-type PDEs, including ones in which the latter has already demonstrated already remarkable performance as compared to other state-of-the-art architectures (e.g., the wave equation in a Marmousi medium). In addition, the decomposition-based model presented here turns out to be capable of solving certain problems for which NeuSA struggles to provide a satisfactory solution, e.g., the Allen-Cahn equation. A downside of D-NeuSA consists in a larger number of trainable parameters (since the model involves multiple neural networks) as compared to NeuSA. However, this does not lead to additional memory constraints and actually requires less training time as compared to the original NeuSA due to a simpler computational graph.

ACKNOWLEDGMENTS

This research was supported by Petrobras.

REFERENCES

A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018.

Alex Bihlo and Roman O. Popovych. Physics-informed neural networks for the shallow-water equations on the sphere. *Journal of Computational Physics*, 456:111024, 2022. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2022.111024.

Arthur Bizzi, Matias Grynberg, Vitor Matias, Daniel Perazzo, João Paulo Lima, Luiz Velho, Nuno Gonçalves, João Pereira, Guilherme Schardong, and Tiago Novello. Flowing: Implicit neural flows for structure-preserving morphing. *arXiv preprint arXiv:2510.09537*, 2025a.

Arthur Bizzi, Leonardo M. Moreira, Márcio Marques, Leonardo Mendonça, Christian Júnior de Oliveira, Vitor Balestro, Lucas dos Santos Fernandez, Daniel Yukimura, Pavel Petrov, João M. Pereira, Tiago Novello, and Lucas Nissenbaum. Neuro-spectral architectures for causal physics-informed networks. In *Advances in Neural Information Processing*, 2025b.

Aline Brougois, Marielle Bourget, Patriek Lailly, Michel Poulet, Patrice Ricarte, and Roelof Versteeg. Marmousi, model and data. In *EAEG workshop-practical aspects of seismic data inversion*, pp. cp–108. European Association of Geoscientists & Engineers, 1990.

J. Bu and A. Karpatne. Quadratic residual networks: A new class of neural networks for solving forward and inverse problems in physics involving pdes. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pp. 675–683, 2021.

Stefano Buoso, Thomas Joyce, and Sebastian Kozerke. Personalising left-ventricular biophysical models of the heart using parametric physics-informed neural networks. *Medical Image Analysis*, 71:102066, 2021.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 6572–6583, Red Hook, NY, USA, 2018. Curran Associates Inc.

Steven M Cox and Paul C Matthews. Exponential time differencing for stiff systems. *Journal of Computational Physics*, 176(2):430–455, 2002.

Yi Ding, Su Chen, Hiroe Miyake, and Xiaojun Li. Physics-informed neural networks with fourier features for seismic wavefield simulation in time-domain nonsmooth complex media. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–13, 2025. doi: 10.1109/TGRS.2025.3581638.

James Donnelly, Alireza Daneshkhah, and Soroush Abolfathi. Physics-informed neural networks as surrogate models of hydrodynamic simulators. *Science of The Total Environment*, 912:168814, 2024.

Alena V Favorskaya, Nikolay I Khokhlov, Vasiliy I Golubev, Anton V Ekimenko, Yurij V Pavlovskiy, Inga Yu Khromova, and Igor B Petrov. Wave processes modelling in geophysics. In *Innovations in Wave Processes Modelling and Decision Making: Grid-Characteristic Method and Applications*, pp. 187–218. Springer, 2018.

Clara Herrero Martin, Alon Oved, Rasheda A Chowdhury, Elisabeth Ullmann, Nicholas S Peters, Anil A Bharath, and Marta Varela. Ep-pinns: Cardiac electrophysiology characterisation using physics-informed neural networks. *Frontiers in Cardiovascular Medicine*, 8:768419, 2022.

Rikuto Ito, Kenji Ishikawa, Risako Tanigawa, and Yasuhiro Oikawa. Three-dimensional sound field reconstruction from optical projections using physics-informed neural networks. *JASA Express Letters*, 5(6), 2025.

Ameya D Jagtap and George Em Karniadakis. Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5), 2020.

Ameya D Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020.

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.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Georgios Kissas, Yibo Yang, Eileen Hwuang, Walter R. Witschey, John A. Detre, and Paris Perdikaris. Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4d flow mri data using physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 358:112623, 2020. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2019.112623.

A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 26548–26560, 2021.

Zhongya Lin, Jinshuai Bai, Shuang Li, Xindong Chen, Bo Li, and Xi-Qiao Feng. A physics-informed neural network framework for simulating creep buckling in growing viscoelastic biological tissues. *Computer Methods in Applied Mechanics and Engineering*, 452:118715, 2026.

Márcio Marques, Leonardo Mendonça, Arthur Bizzi, Leonardo Moreira, Christian Oliveira, Deborah Oliveira, Lucas Fernandez, Vitor Balestro, João Pereira, Daniel Yukimura, Tiago Novello, Pavel Petrov, and Lucas Nissenbaum. Stable adaptive training for physics-informed neural networks in acoustic wave propagation. *JASA Express Letters*, 5(11), 2025.

Constanza A. Molina Catricheo, Fabrice Lambert, Julien Salomon, and Elwin van 't Wout. Modeling global surface dust deposition using physics-informed neural networks. *Communications Earth & Environment*, 5(1):778, 2024. doi: 10.1038/s43247-024-01942-2.

YongKyung Oh, Seungsu Kam, Jonghun Lee, Dong-Young Lim, Sungil Kim, and Alex A. T. Bui. Comprehensive review of neural differential equations for time series analysis. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25, 2025. ISBN 978-1-956792-06-5. doi: 10.24963/ijcai.2025/1179.

Michael Penwarden, Ameya D Jagtap, Shandian Zhe, George Em Karniadakis, and Robert M Kirby. A unified scalable framework for causal sweeping strategies for physics-informed neural networks (pinns) and their temporal decompositions. *Journal of Computational Physics*, 493:112464, 2023.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 09–15 Jun 2019.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Shahed Rezaei, Ahmad Moeineddin, and Ali Harandi. Learning solutions of thermodynamics-based nonlinear constitutive material models using physics-informed neural networks. *Computational Mechanics*, 74(2):333–366, 2024.

Pratanu Roy and Stephen T Castonguay. Exact enforcement of temporal continuity in sequential physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 430:117197, 2024.

Angelo Sajeva, Mattia Aleardi, Eusebio Stucchi, Nicola Bienati, and Alfredo Mazzotti. Estimation of acoustic macro models using a genetic full-waveform inversion: Applications to the Marmousi model. *Geophysics*, 81(4):R173–R184, 2016.

William E Schiesser. *The numerical method of lines: integration of partial differential equations*. Elsevier, 2012.

William E Schiesser. *Computational mathematics in engineering and applied science: ODEs, DAEs, and PDEs*. CRC press, 2014.

Stefan Schoder. Physics-informed neural networks for modal wave field predictions in 3d room acoustics. *Applied Sciences*, 15(2):939, 2025.

Josef Stoer, Roland Bulirsch, R Bartels, Walter Gautschi, and Christoph Witzgall. *Introduction to numerical analysis*, volume 1993. Springer, 1980.

Don Vasco and Coral Chan. Utah forge: Development of a reservoir seismic velocity model and seismic resolution study. Geothermal Data Repository, Array Information Technology, https://doi.org/10.15121/1989942, 2022. Accessed: 2025-11-26.

S. Wang, X. Yu, and P. Perdikaris. When and why pinns fail to train: a neural tangent kernel perspective. *Journal of Computational Physics*, 449, 2022.

S. Wang, S. Sankaran, and P. Perdikaris. Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421, 2024.

Colby L Wight and Jia Zhao. Solving allen-cahn and cahn-hilliard equations using the adaptive physics informed neural networks. *arXiv preprint arXiv:2007.04542*, 2020.

J. C. Wong, C. C. Ooi, A. Gupta, and Y.-S. Ong. Learning in sinusoidal spaces with physics-informed neural networks. *IEEE Transactions on Artificial Intelligence*, 5(3), 2024.

H. Wu, H. Luo, Y. Ma, J. Wang, and M. Long. Ropinn: Region optimized physics-informed neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

Shirindokht Yazdani and Mojtaba Tahani. Data-driven discovery of turbulent flow equations using physics-informed neural networks. *Physics of Fluids*, 36(3), 2024.

Enrui Zhang, Ming Dao, George Em Karniadakis, and Subra Suresh. Analyses of internal structures and defects in materials using physics-informed neural networks. *Science Advances*, 8(7): eabk0644, 2022. doi: 10.1126/sciadv.abk0644.

Leo Zhiyuan Zhao, Xueying Ding, and B. Aditya Prakash. Pinnsformer: A transformer-based framework for physics-informed neural networks. *arXiv preprint arXiv:2307.11833*, 2023. Accepted at ICLR 2024 Poster.

Bin Zheng, Tongchun Li, Huijun Qi, Lingang Gao, Xiaoqing Liu, and Li Yuan. Physics-informed machine learning model for computational fracture of quasi-brittle materials without labelled data. *International Journal of Mechanical Sciences*, 223:107282, 2022. doi: 10.1016/j.ijmecsci.2022.107282.

Hao Zhongkai, Jiachen Yao, Chang Su, Hang Su, Ziao Wang, Fanzhi Lu, Zeyu Xia, Yichi Zhang, Songming Liu, Lu Lu, et al. Pinnacle: A comprehensive benchmark of physics-informed neural networks for solving pdes. *Advances in Neural Information Processing Systems*, 37:76721–76774, 2024.