WINN: A PDE RESIDUAL LOSS-FREE NETWORK FOR HOMOGENEOUS LONG-TIME WAVE PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Physics-informed neural networks (PINNs) struggle to solve wave equations over long temporal domains. In this work, we investigate the sources of this failure and introduce **WINN** (Wave-Informed Neural Network), a novel neural architecture that incorporates the analytical structure of the wave equation directly into the model. WINN satisfies the homogeneous wave equation with constant wave speed exactly by construction, eliminating the need for a PDE residual loss and allowing training to focus solely on the initial and boundary conditions. We show through extensive experiments that WINN remains accurate across long-time wave problems with different frequencies, spatial dimensions, wave speeds, and domain geometries, achieving 7–86$\times$ lower relative L_2 error than the best-performing baseline. WINN achieves training times that are slightly shorter than Vanilla PINN in 2D and 1.7$\times$ faster than FBPINN in 3D. These results demonstrate that incorporating the governing PDE structure directly into the neural representation can substantially improve the accuracy and robustness of neural solvers for long-time wave dynamics.

1 INTRODUCTION

Physics-informed neural networks (PINNs) (Raissi et al., 2019; Karniadakis et al., 2021) have emerged as a general framework for solving partial differential equations (PDEs) with neural networks. By incorporating the governing differential equation into the training objective, PINNs can learn solutions from initial and boundary conditions without requiring a conventional numerical discretization of the PDE domain. This approach has been applied to a broad range of scientific and engineering problems, including forward simulation, inverse problems, and parameter identification (Karniadakis et al., 2021). Despite this generality, PINNs can encounter substantial difficulties in challenging PDE regimes, particularly when the solution exhibits persistent oscillations or the temporal domain is long (Wang et al., 2021; Hao et al., 2023; Moseley et al., 2023).

This paper focuses on the homogeneous wave equation with constant wave speed, a fundamental model of wave propagation in physics and engineering, with applications in acoustics, electromagnetics, and elasticity (Vandekerckhove et al., 2014; Wapenaar, 2019; de Hoop, 1996; Feynman et al., 1963). Despite its simple and widely studied form, accurately solving this equation over long temporal domains remains a challenging problem for neural PDE solvers, particularly when the solution exhibits persistent oscillations.

This difficulty is evident in the PINNacle benchmark (Hao et al., 2023), which evaluates a broad range of neural PDE solvers on challenging PDE problems. On its long-time Wave2D benchmark (Wave2d-MS), defined over $t \in [0, 100]$, existing PINN methods obtain relative L2 errors on the order of unity, indicating a complete failure to recover the underlying wave solution. Importantly, this behavior is not restricted to a single implementation of a standard PINN. The difficulty persists across different reweighting, resampling, optimizers, loss functions and network architectures. These observations raise a fundamental question: *why are long-time wave equations so difficult for PINNs, and can the structure of the wave equation itself be incorporated into the neural representation to avoid this failure mode?*

In this work, we investigate this question directly. Our analysis in Section 3.1 suggests that the PDE loss becomes increasingly difficult to minimize as the temporal horizon grows. Moreover, prior work has shown that jointly optimizing multiple loss terms can introduce competing objectives and

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.

gradient conflicts (Wang et al., 2020a; 2025). These observations motivate our approach of replacing PDE-residual optimization with a representation that, by construction, satisfies the wave equation.

We introduce **WINN** (**Wave-Informed Neural Network**), a novel neural architecture based on a representation of the general solution to the homogeneous wave equation with constant wave speed given by Whittaker (1903). The resulting network satisfies the wave equation identically, up to the numerical precision of the computation. Consequently, the PDE residual need not appear in the training objective. Training can instead focus entirely on matching the initial and boundary conditions.

We compare[1] WINN against several approaches that address the difficulties of standard PINNs from different perspectives, including conventional PINNs (Raissi et al., 2019), causal training (Wang et al., 2022), Fourier-feature PINNs (Tancik et al., 2020), and finite-basis/domain-decomposition PINNs (Moseley et al., 2023). Our experiments span long-time wave problems in two and three spatial dimensions, including low- and high-frequency solutions and a non-convex L-shaped domain. We evaluate both solution accuracy and computational cost. Across these settings, WINN remains capable of accurately representing the wave solution in regimes where conventional residual-based approaches fail. In our experiments, WINN achieves 7–86$\times$ lower error than the best baseline and 1.7–4$\times$ faster training than the domain-decomposition method FBPINN, while eliminating the need for PDE-residual optimization and reducing the burden of balancing multiple loss objectives.

Our contributions are summarized as follows:

- We investigate the failure of PINNs on long-time wave equations and show that extending the temporal horizon substantially degrades their ability to minimize the PDE residual and recover the correct wave dynamics.
- We introduce **WINN**, a novel wave-informed neural architecture whose representation satisfies the wave equation exactly by construction, removing the need for a PDE residual loss.
- We incorporate Fourier features into WINN to improve its representation of long-time and high-frequency wave solutions.
- We demonstrate through extensive experiments that WINN substantially improves accuracy over conventional PINNs, causal training, Fourier-feature PINNs, and domain-decomposition approaches across long-time wave problems with different frequencies, wave speeds, spatial dimensions, and domain geometries.

2 Related Work

Physics-informed neural networks (PINNs) (Raissi et al., 2019) solve PDEs by representing the solution with a neural network and enforcing the governing equations through a residual loss. While flexible and mesh-free, this formulation requires jointly optimizing PDE, initial-condition, and boundary-condition losses, which can lead to poorly balanced gradients and ill-conditioned optimization (Wang et al., 2020a; 2025). These difficulties become particularly pronounced for long-time, high-frequency, and multi-scale problems (Hao et al., 2023; Krishnapriyan et al., 2021; Wang et al., 2020b).

Several approaches improve PINN optimization without changing this residual-based formulation. Causal training (Wang et al., 2022) incorporates the temporal ordering of dynamical systems by progressively emphasizing later times, improving performance on difficult time-dependent problems. Sequential time-marching methods (Wight & Zhao, 2020) partition the temporal domain into successive subdomains and train a separate network on each segment, using the solution from one segment to initialize the next. This reduces the temporal extent of each optimization subproblem, but introduces a sequential training procedure and can accumulate errors over long time horizons. Domain-decomposition methods such as FBPINNs (Moseley et al., 2023) similarly partition the domain among overlapping local networks, reducing the complexity of individual optimization problems and mitigating spectral bias. Fourier feature mappings (Tancik et al., 2020) provide a complementary strategy by improving the representation of high-frequency and multi-scale solutions. More

[1]The code used to reproduce our experiments will be made publicly available upon acceptance.

recently, PINNsFormer (Zhao et al., 2024) has introduced a Transformer-based architecture that converts point-wise inputs into pseudo-sequences and uses attention mechanisms and a sequential loss to capture temporal dependencies.

Several studies attempt to reduce the burden of enforcing physical constraints by incorporating them directly into the neural representation. Specifically, hard-constrained PINNs construct trial solutions that satisfy prescribed boundary conditions identically, typically by combining a function satisfying the boundary data with an auxiliary function that vanishes on the boundary (Lagaris et al., 1998; Lu et al., 2021). Such constructions remove the corresponding boundary terms from the optimization objective, but their formulation can become problem-specific for complex geometries or boundary conditions, and they do not eliminate the need to enforce the governing PDE through residual minimization. For incompressible Navier–Stokes equations, for example, PINNs can parameterize the velocity field via a stream function, which ensures the divergence-free constraint is satisfied identically (Raissi et al., 2019). Ballini et al. (2026) use semi-holomorphic neural networks to represent harmonic functions by construction for solving the three-dimensional Laplace equation. These approaches demonstrate the potential of embedding known differential structures directly into neural representations rather than enforcing all physical constraints through residual losses.

For wave equations, analytical representations provide a particularly direct way of encoding the governing dynamics. We use in our work results by Whittaker (1903), who showed that solutions of the wave equation admit integral representations in terms of functions of directional projections. This representation provides a natural foundation for constructing neural parameterizations that satisfy the wave equation exactly, rather than enforcing it through a residual loss.

3 METHOD

3.1 PINN'S FAILURE ON THE LONG-TIME WAVE EQUATION

To investigate PINN's failure on the long-time two-dimensional wave equation,

$$u_{tt} = c^2 \left(u_{xx} + u_{yy}\right), \tag{1}$$

we consider the standing-wave solution on $[0,1]^2$ with $c = 2$,

$$u(x, y, t) = \sin(2x)\sin(2y)\cos(4\sqrt{2}t), \tag{2}$$

whose temporal frequency is $\omega = 4\sqrt{2}$, corresponding to a period $P = \frac{2\pi}{4\sqrt{2}} \approx 1.11$. Thus, increasing the temporal horizon increases the number of oscillations while leaving the spatial domain unchanged. We train a Vanilla PINN on

$$t \in [0, T], \qquad T \in 1, 5, 10, 20, 50, 100, \tag{3}$$

with spatial domain $[0,1]^2$. The optimization and training process is described in more detail in Appendix C.3. Figure 1 shows that the relative L_2 error increases rapidly with T, reaching approximately 15% at $T = 10$ and 76% at $T = 20$.

One possible explanation for this failure is that large temporal inputs and the spectral bias of MLPs make long-time oscillations difficult to represent. To test this, we replace the MLP input with a Fourier Feature encoding (Tancik et al., 2020), using frequencies sufficient to represent the ground-truth solution. The results are shown in Figure 1, Fourier Features do not prevent failure at long horizons; the error increases sharply between $T = 5$ and $T = 10$. This is also not due to increasing the domain size while keeping the number of collocation points per step fixed, as we show in Appendix D.1. Thus, neither large temporal inputs nor spectral bias alone appears to explain the observed degradation.

To identify the source of this failure, we examine the individual loss terms in Table 1[2][3]. The initial-condition losses remain small as T increases[4], since they are evaluated only at $t = 0$. By contrast,

[2]We consider $T \in [5, 12]$, where most of the degradation occurs. For $T > 12$, once the relative L_2 error exceeds unity, the solution collapses toward the zero solution, which trivially satisfies the PDE.

[3]We train using learning-loss annealing to balance the loss terms such that they contribute equally to the gradient update.

[4]Their values increase; however, we consider 10^{-5} sufficiently small for the IC and IC-V losses, an assumption supported by WINN, which achieves low relative L_2 error with loss values ranging from 10^{-4} to 10^{-7}.

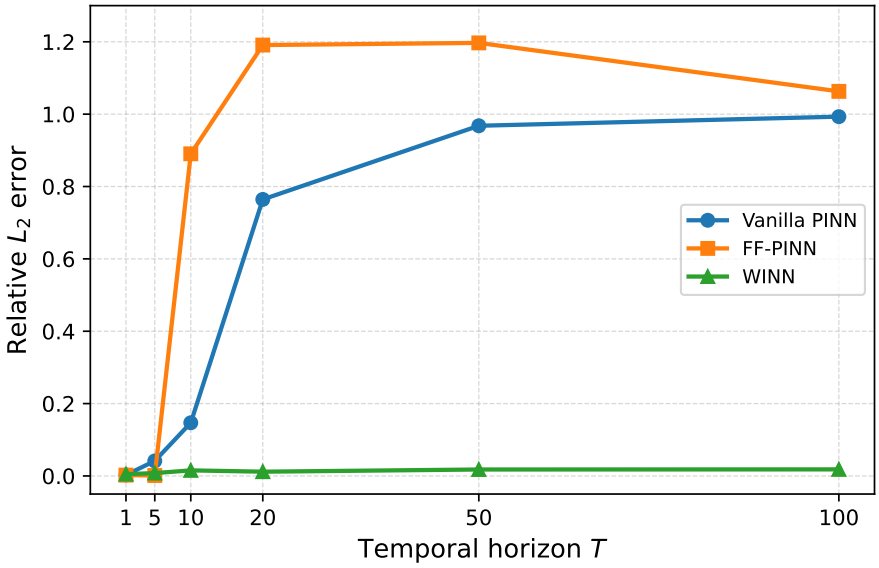

Figure 1: Relative L_2 error of Vanilla PINN (in blue), Fourier-Features PINN (orange), and our WINN method (green) as a function of the temporal horizon T for the two-dimensional wave equation (2).

Table 1: Relative L_2 error of the full loss and of the individual terms of the Fourier Feature PINN for increasing temporal horizons. BC denotes the boundary-condition loss, PDE the PDE-residual loss, IC the initial-displacement loss, and IC-V the initial-velocity loss.

T	Rel. L_2	BC	PDE	IC	IC-V
5	0.121%	1.32×10^{-7}	2.58×10^{-3}	2.72×10^{-8}	1.92×10^{-8}
10	89.0%	7.43×10^{-2}	6.97×10^{-1}	1.97×10^{-4}	2.21×10^{-5}
12	103%	9.21×10^{-2}	3.34	5.78×10^{-5}	2.05×10^{-5}

the PDE and boundary conditions must be satisfied throughout the temporal domain. While both losses increase substantially with T, the PDE loss is particularly computationally expensive. It requires back-propagating through second-order derivatives and enforcing the PDE throughout the full spacetime domain, which typically requires many more collocation points than the boundary and initial conditions.

These observations motivate replacing the PDE-residual optimization with an exact PDE-satisfying representation, eliminating the need to optimize the PDE residual and leaving only the boundary and initial conditions for training.

3.2 Exact Representation of Wave Equation Solutions

We consider the two-dimensional wave equation

$$u_{tt} = c^2 \left(u_{xx} + u_{yy}\right), \tag{4}$$

where $u : \Omega \times [0, T] \to \mathbb{R}$.

Let f denote our MLP, and suppose that we seek an *encoding function* $g : \Omega \times [0, T] \to \mathbb{R}$ such that $u(x, y, t) = f(g(x, y, t))$ satisfies the wave equation by construction. Assume that f and g are twice continuously differentiable. By the chain rule,

$$u_x = f'(g) g_x, \tag{5}$$

$$u_{xx} = f''(g) g_x^2 + f'(g) g_{xx}, \tag{6}$$

and similarly,

$$u_{yy} = f''(g)g_y^2 + f'(g)g_{yy}, \tag{7}$$

$$u_{tt} = f''(g)g_t^2 + f'(g)g_{tt}. \tag{8}$$

Therefore,

$$u_{tt} - c^2\left(u_{xx} + u_{yy}\right) = f''(g)\left[g_t^2 - c^2\left(g_x^2 + g_y^2\right)\right] + f'(g)\left[g_{tt} - c^2\left(g_{xx} + g_{yy}\right)\right]. \tag{9}$$

Thus, if $g(x, y, t)$ itself satisfies both

$$g_t^2 = c^2\left(g_x^2 + g_y^2\right) \tag{10}$$

and the wave equation, then $u = f \circ g$ satisfies the wave equation identically, independently of the choice of f. A family of such functions g is given in the following proposition.

Proposition 1. *For every $\theta \in [0, 2\pi]$, the function*

$$g_\theta(x, y, t) = x\cos\theta + y\sin\theta + ct \tag{11}$$

is a valid choice of g.

The proof is provided in Appendix A.1.

While using a single function such as $u = f \circ g_\theta$ guarantees exact satisfaction of the PDE, it severely restricts the expressivity of the resulting representation. In particular, all points (x, y, t) satisfying $x\cos\theta + y\sin\theta + ct = \text{constant}$ are mapped to the same input of the MLP and therefore must have the same predicted value. Thus, the model cannot distinguish between different points lying on the same level set of g_θ.

Integrating over the family of valid encodings indexed by θ removes this restriction. In particular, Whittaker (1903) showed that general solutions of the wave equation can be represented in the form

$$u(x, y, t) = \int_0^{2\pi} f\left(x\cos\theta + y\sin\theta + ct, \theta\right) d\theta. \tag{12}$$

The integrand satisfies the wave equation for every fixed θ by Proposition 1. Since the wave equation is linear, differentiation under the integral sign implies that u also satisfies the wave equation identically. More details are given in Appendix A.2.

The same construction extends directly to the three-dimensional wave equation. Specifically, we define

$$g_{\theta,\phi}(x, y, z, t) = x\sin\theta\cos\phi + y\sin\theta\sin\phi + z\cos\theta + ct. \tag{13}$$

The corresponding general representation is

$$u(x, y, z, t) = \int_0^{\pi}\int_0^{2\pi} f\left(x\sin\theta\cos\phi + y\sin\theta\sin\phi + z\cos\theta + ct, \theta, \phi\right) d\phi d\theta. \tag{14}$$

A similar argument shows that each integrand satisfies the three-dimensional wave equation exactly[5].

3.3 Neural Parameterization

We parameterize the function f in (12) by a multilayer perceptron (MLP) with tanh activations, denoted by f_ψ, with trainable parameters ψ. We approximate the angular integral (12) using a finite weighted sum,

$$u_\psi(x, y, t) = \Delta\theta \sum_{i=1}^{N_\theta} w_i f_\psi(g_i, \theta_i), \tag{15}$$

where

$$g_i = x\cos\theta_i + y\sin\theta_i + ct. \tag{16}$$

[5]Whittaker (1903) originally established the corresponding representation for the three-dimensional wave equation. The two-dimensional case follows as a special case by fixing $\theta = \pi/2$, which reduces the spatial direction to $(\cos\phi, \sin\phi)$.

The coefficients w_i are trainable parameters optimized jointly with the network. N_θ is a hyperparameter controlling the number of angular directions. We initialize the angular locations uniformly,

$$\bar{\theta}_i = \frac{2\pi i}{N_\theta}, \qquad i = 1, \dots, N_\theta, \tag{17}$$

and allow them to adapt according to

$$\theta_i = \bar{\theta}_i + \delta_i, \qquad \delta_i = \frac{\Delta\theta}{2}\tanh(\alpha_i), \qquad \Delta\theta = \frac{2\pi}{N_\theta}, \tag{18}$$

where α_i are trainable parameters. This constrains each θ_i to its initial angular interval while allowing the integration directions to adapt to the target solution. Similar implementation is used for the 3D case in Appendix B.2.

3.4 Fourier Feature Parameterization for Long-Time Domains

For long-time domains, g_i can span a large range. We therefore normalize g_i and encode it using Fourier features. Importantly, composing the Fourier-feature encoding with the MLP yields a twice continuously differentiable function of g_i. Hence, each summand remains a valid solution of the wave equation, and by linearity, so does their weighted sum. The specific Fourier-feature encoding is described in Appendix B.1.

4 Experiments

We evaluate WINN across a diverse set of wave-equation problems, varying spatial dimension, spatial and temporal frequency, wave speed, domain geometry, solution family, and initial conditions. We consider four settings: (i) 2D low-frequency, (ii) 2D high-frequency, (iii) 3D, and (iv) a 2D non-convex, L-shaped spatial domain. Here, "low-frequency" and "high-frequency" refer to the spatial frequency of the solution. All experiments use long temporal domains. Details of the PDEs, ground-truth solutions, domains, initial and boundary conditions are provided in Appendix C.1. To avoid giving WINN an advantage through the choice of ground-truth solutions, we generally exclude solutions that can be expressed as finite sums of functions of the form $f(x\cos\theta + y\sin\theta + ct)$, which is closely related to WINN's analytical representation. The exception is the first experiment, which uses a commonly adopted standing-wave solution that admits such a representation. We retain this case as a reference point for comparison with prior work and observe the same qualitative behavior as in the more challenging settings without this representation. Each experiment is repeated with three different random seeds to assess the robustness of the results.

4.1 Baselines

We compare WINN against four representative physics-informed neural PDE solvers: **Vanilla PINN** (Raissi et al., 2019), which minimizes PDE residual, boundary, and initial-condition losses; **FF-PINN** (Tancik et al., 2020), which augments PINNs with Fourier features for high-frequency solutions; **Causal Training** (Wang et al., 2022), which encourages sequential satisfaction of the PDE in time; and **FBPINN** (Moseley et al., 2023), which uses domain decomposition with localized neural networks. Detailed model configurations are provided in Appendix C.2.

4.2 Optimization and Training

Vanilla PINN, FF-PINN, Causal PINN, and FBPINN minimize weighted boundary, initial-displacement, initial-velocity, and PDE-residual losses; WINN is trained only on boundary and initial-condition losses because it satisfies the wave equation by construction. Our method, WINN, as well as Vanilla PINN, and FF-PINN, adapt the loss weights using LRA (Wang et al., 2020a), while Causal PINN prioritizes the boundary and initial conditions and progressively learns the domain further in time. WINN, Vanilla PINN, FF-PINN, and Causal PINN use a two-stage optimization procedure consisting of SOAP (Vyas et al., 2025), following Wang et al. (2025), followed by L-BFGS refinement. Boundary and initial-condition points are resampled during SOAP and fixed during L-BFGS, while FBPINN uses its native trainer with the PINNacle configuration. All experiments use double-precision arithmetic and were conducted on an NVIDIA A40 GPU. Detailed optimization and training configurations are provided in Appendix C.3.

Table 2: Mean relative L_2 error and total training time across three random seeds for each method and experimental setting. Best mean results are shown in bold.

Experiment	WINN		PINN		FF-PINN		Causal PINN		FBPINN	
	Rel. L_2 ↓	Time (s)↓	Rel. L_2 ↓	Time (s)↓	Rel. L_2 ↓	Time (s)↓	Rel. L_2 ↓	Time (s)↓	Rel. L_2 ↓	Time (s)↓
Low-frequency 2D	**1.3 %**	334	99.0 %	340	99.9 %	511	100.4 %	510	31.9 %	1378
High-frequency 2D	**0.84 %**	331	99.8 %	338	72.5 %	508	95.8 %	497	81.8 %	1317
3D	**2.6 %**	1799	98.5 %	607	18.0 %	922	22.6 %	1058	111.6 %	3138
L-shaped 2D	**4.2 %**	341	99.8 %	356	74.0 %	527	108.3 %	518	104.4 %	1395

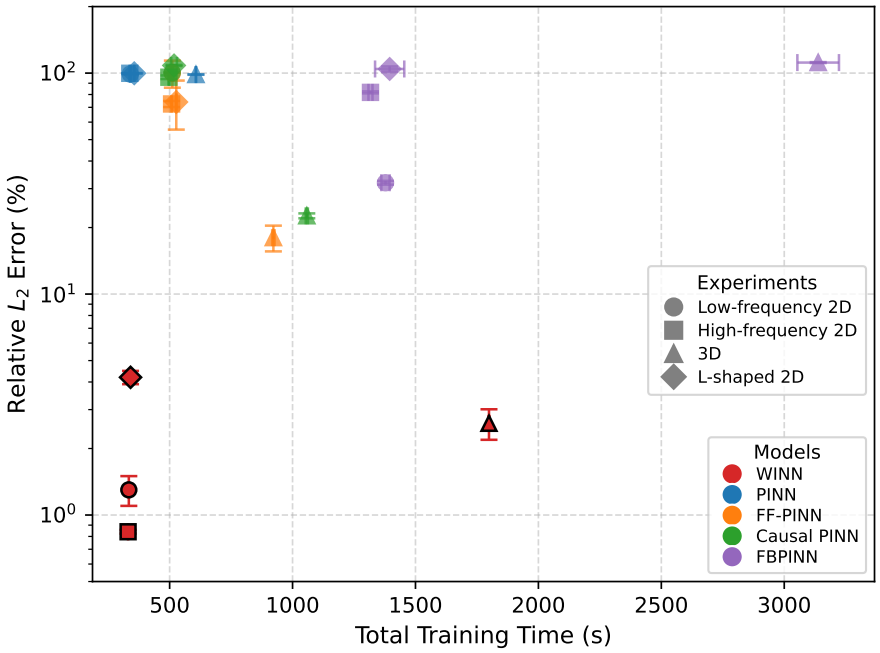

Figure 2: Mean relative L_2 error and total training time across three random seeds for all methods (distinguished by the color of points) and experimental settings (distinguished by their shapes). Error bars indicate the corresponding standard deviations.

4.3 Results

Overall, WINN substantially outperforms the baseline methods in solution accuracy across all four experimental settings. This gap persists across low- and high-frequency regimes, two- and three-dimensional domains, and the L-shaped geometry. WINN also remains computationally competitive in 2D, while its higher-dimensional angular integral increases its training cost in 3D. Table 2 summarizes the mean relative L_2 error and total training time across the four experimental settings, which are also visualized in Figure 2; standard deviations across three random seeds are reported in Appendix D.2.

Qualitative results. Figure 3 visualizes the solution obtained by WINN on the high-frequency 2D problem together with the two strongest-performing baselines. WINN closely reproduces the reference solution throughout the space-time domain, whereas the baseline solutions exhibit substantial deviations. Qualitative results for the low-frequency 2D and L-shaped 2D problems are provided in Appendix D.3.

Accuracy. WINN remains below 5% relative L_2 error in all four settings, demonstrating the effectiveness of the exact wave-equation constraint across dimensions, frequencies, and geometries. Vanilla PINN fails across all problems, with errors near 100%, consistent with the long-time failure mode identified in Section 3.1. FF-PINN improves over Vanilla PINN but remains substantially less accurate than WINN. Notably, its 18.0% error in 3D is accompanied by a large PDE residual.

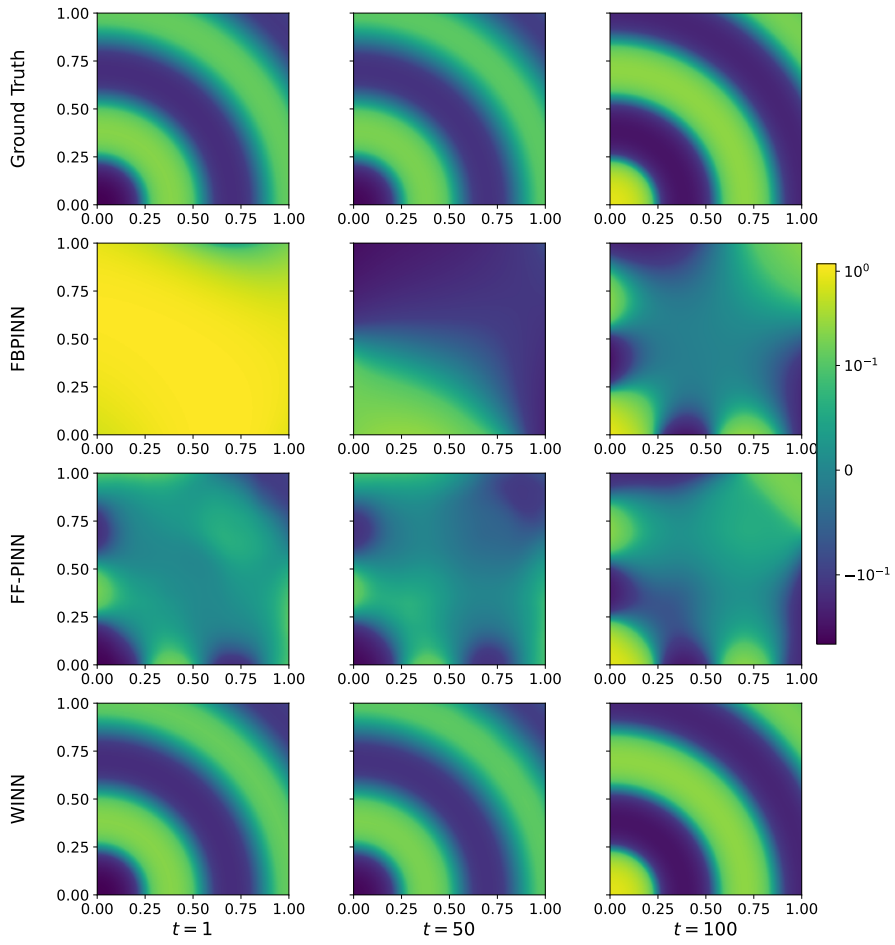

Figure 3: Qualitative comparison of WINN and the two strongest-performing baselines with $T = 100$ on the high-frequency 2D problem at representative time points.

In real-world applications where the ground-truth solution is unavailable, such a residual makes it difficult to assess the accuracy of the learned solution, thereby limiting its reliability. Causal PINN likewise fails to reliably recover accurate solutions and is highly sensitive to temporal discretization and tolerance hyperparameters; after evaluating multiple configurations, we selected the most stable one, which nevertheless remained inaccurate. FBPINN performs best among the baselines on the low-frequency 2D problem but degrades substantially on the remaining settings.

Computational cost. WINN is slightly faster than Vanilla PINN in the 2D experiments, with training times of 331–341 seconds versus 338–356 seconds. This is despite the additional forward evaluations required by WINN's angular integral. These additional evaluations are offset by the absence of interior collocation points and PDE-residual computations. Consequently, WINN avoids computing the PDE residual and its associated derivatives at interior points, reducing the computational cost of each training step. In 3D, WINN is slower than most baseline methods due to the higher-dimensional integral, but still $1.7\times$ faster than FBPINN, with training times of 1799 and 3138 seconds, respectively.

4.4 Ablation Studies

We evaluate the contribution of key WINN components: the number of angular quadrature points N_θ, the optimizer, and Fourier feature encoding. All ablations are performed on Experiment (iv)

Table 3: Ablation study of WINN on the 2D L-shaped domain (Experiment (iv)). Mean relative L_2 error (L2RE) and training time across three random seeds are shown.

Variant	L2RE ↓	Time (s) ↓
$N_\theta = 16$	5.61 %	**219**
$N_\theta = 32$	**4.21 %**	341
$N_\theta = 64$	4.25 %	589
$N_\theta = 128$	4.24 %	1049
Adam instead of SOAP	27.18 %	307
Without Fourier features	100.04 %	291

(2D L-shaped domain) using the same training and evaluation protocol as the main experiments. Results are reported in Table 3.

The default configuration uses $N_\theta = 32$, SOAP, and Fourier features. Varying N_θ reveals a trade-off between accuracy and computational cost: increasing it from 16 to 32 improves accuracy, while larger values plateau. The slight increase in mean error remains within the standard deviation across seeds, while training time increases substantially.

Replacing SOAP with Adam increases the error from 4.21% to 27.18%. Despite eliminating the PDE residual loss, WINN still requires the simultaneous optimization of three objectives, for which SOAP has proven effective (Wang et al., 2025).

Finally, removing Fourier features increases the error to 100.04%, as the transformed inputs g_i become dominated by the temporal coordinate, reducing sensitivity to spatial variations.

These results underscore the importance of both SOAP and Fourier features for accurate WINN training. Notably, FF-PINN uses Fourier features with the same frequencies as WINN, yet achieves an error of 74.0%, highlighting the benefit of WINN's exact satisfaction of the PDE.

5 Limitations and Future Work

WINN also has limitations. The computational cost of the representation grows with the number of angular quadrature points and, more substantially, with spatial dimension because the corresponding integral becomes higher-dimensional. This is reflected in the increased training time observed in the 3D experiment. In addition, the current formulation is specialized to the homogeneous wave equation with constant wave speed. We hope to extend WINN in the future to non-homogeneous wave equations and heterogeneous media.

6 Conclusion

We investigated the failure of PINNs on long-time wave equations and found that optimizing the PDE residual becomes increasingly difficult as the temporal horizon grows. We introduced WINN, which directly incorporates the analytical structure of the homogeneous wave equation with constant wave speed into the neural representation, thereby satisfying the PDE exactly without a residual loss. This shifts the learning problem from jointly enforcing the PDE and fitting the prescribed conditions to only fitting the initial and boundary conditions.

Across four long-time experiments, WINN consistently achieved substantially lower error than conventional PINNs, Fourier-feature PINNs, causal training, and FBPINN, while remaining computationally competitive in two and three dimensions. The results also show that the representation remains effective across different frequencies, spatial dimensions, and domain geometries, suggesting that exact incorporation of PDE structure need not come at the cost of expressivity.

More broadly, our results suggest that for long-time wave problems, embedding known analytical structure into the neural representation can be more effective than improving residual-based optimization alone. This motivates further investigation of structure-preserving neural architectures for PDEs in regimes where conventional residual minimization becomes increasingly difficult.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, assist with translation, and provide feedback on research methodology and experiments. We have not used generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, generate synthetic datasets, clean or reformat datasets, support qualitative and thematic data analysis, or interpret results.

Additionally, we used generative AI tools to create and edit software code, identify relevant literature, and edit the paper to improve readability. We have reviewed all AI-assisted work, and LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the WINN architecture, training procedure, and experimental settings in Sections 3, 4.2 and 4, with additional implementation details and experimental configurations in Appendix B and C. The numerical results reported in the paper were obtained using the procedures and hyperparameters described therein. The code used to reproduce our experiments will be made publicly available upon acceptance.

REFERENCES

Enrico Ballini, Allan Peter Engsig-Karup, and Tito Andriollo. A holomorphic neural network framework for 3d boundary value problems governed by harmonic potentials, 2026. URL https://arxiv.org/abs/2605.31231.

Adrianus T. de Hoop. The initial-value problems in acoustics, elastodynamics and electromagnetics. *Wave Motion*, 23(2):121–137, 1996. ISSN 0165-2125. doi: https://doi.org/10.1016/0165-2125(95)00050-X. URL https://www.sciencedirect.com/science/article/pii/016521259500050X.

Richard Phillips Feynman, Robert B. Leighton, and Matthew Sands. *The Feynman Lectures on Physics*. 1963. ISBN 978-0-201-02115-8.

Zhongkai Hao, Jiachen Yao, Chang Su, Hang Su, Ziao Wang, Fanzhi Lu, Zeyu Xia, Yichi Zhang, Songming Liu, Lu Lu, and Jun Zhu. Pinnacle: A comprehensive benchmark of physics-informed neural networks for solving pdes, 2023. URL https://arxiv.org/abs/2306.08827.

{George Em} Karniadakis, {Ioannis G.} Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, June 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00314-5. Publisher Copyright: © 2021, Springer Nature Limited.

Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks, 2021. URL https://arxiv.org/abs/2109.01050.

I.E. Lagaris, A. Likas, and D.I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998. ISSN 1045-9227. doi: 10.1109/72.712178. URL http://dx.doi.org/10.1109/72.712178.

Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989. URL https://api.semanticscholar.org/CorpusID:5681609.

Lu Lu, Raphael Pestourie, Wenjie Yao, Zhicheng Wang, Francesc Verdugo, and Steven G. Johnson. Physics-informed neural networks with hard constraints for inverse design, 2021. URL https://arxiv.org/abs/2102.04626.

Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics*, 49(4), 2023. ISSN 1572-9044. doi: 10.1007/s10444-023-10065-9. URL http://dx.doi.org/10.1007/s10444-023-10065-9.

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. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains, 2020. URL https://arxiv.org/abs/2006.10739.

Steven Vandekerckhove, Bart Vandewoestyne, Herbert De Gersem, Koen Van Den Abeele, and Stefan Vandewalle. Mimetic discretization and higher order time integration for acoustic, electromagnetic and elastodynamic wave propagation. *Journal of Computational and Applied Mathematics*, 259:65–76, 2014. ISSN 0377-0427. doi: https://doi.org/10.1016/j.cam.2013.02.027. URL https://www.sciencedirect.com/science/article/pii/S0377042713001064. Proceedings of the Sixteenth International Congress on Computational and Applied Mathematics (ICCAM-2012), Ghent, Belgium, 9-13 July, 2012.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. Soap: Improving and stabilizing shampoo using adam, 2025. URL https://arxiv.org/abs/2409.11321.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient pathologies in physics-informed neural networks, 2020a. URL https://arxiv.org/abs/2001.04536.

Sifan Wang, Xinling Yu, and Paris Perdikaris. When and why pinns fail to train: A neural tangent kernel perspective, 2020b. URL https://arxiv.org/abs/2007.14527.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2021.113938. URL https://www.sciencedirect.com/science/article/pii/S0045782521002759.

Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks, 2022. URL https://arxiv.org/abs/2203.07404.

Sifan Wang, Ananyae Kumar Bhartari, Bowen Li, and Paris Perdikaris. Gradient alignment in physics-informed neural networks: A second-order optimization perspective, 2025. URL https://arxiv.org/abs/2502.00604.

Kees Wapenaar. Unified matrix–vector wave equation, reciprocity and representations. *Geophysical Journal International*, 216(1):560–583, 01 2019. ISSN 0956-540X. doi: 10.1093/gji/ggy451. URL https://doi.org/10.1093/gji/ggy451.

E.T. Whittaker. On the partial differential equations of mathematical physics. *Mathematische Annalen*, 57:333–355, 1903. ISSN 0025-5831. doi: 10.1007/bf01444290.

Colby L. Wight and Jia Zhao. Solving allen-cahn and cahn-hilliard equations using the adaptive physics informed neural networks, 2020. URL https://arxiv.org/abs/2007.04542.

Zhiyuan Zhao, Xueying Ding, and B. Aditya Prakash. Pinnsformer: A transformer-based framework for physics-informed neural networks, 2024. URL https://arxiv.org/abs/2307.11833.

A PROOFS

A.1 PROOF OF PROPOSITION 1

Proof. We have

$$g_x = \cos\theta, \qquad g_y = \sin\theta, \qquad g_t = c, \tag{19}$$

and

$$g_{xx} = g_{yy} = g_{tt} = 0. \tag{20}$$

Hence,

$$g_{tt} - c^2(g_{xx} + g_{yy}) = 0. \tag{21}$$

So g_θ satisfies the wave equation (4). Moreover,

$$g_t^2 = c^2 = c^2\left(\cos^2\theta + \sin^2\theta\right) = c^2\left(g_x^2 + g_y^2\right). \tag{22}$$

Therefore, both required conditions are satisfied, and consequently, $u = f \circ g_\theta$ satisfies the wave equation identically. □

A.2 PROOF OF THE EXACT WAVE-EQUATION REPRESENTATION

Here we show that integrating the family of functions $f\left(x\cos\theta + y\sin\theta + ct, \theta\right)$ over θ preserves the exact satisfaction of the two-dimensional wave equation.

Proposition 2. *Let*

$$u(x, y, t) = \int_0^{2\pi} f\left(x\cos\theta + y\sin\theta + ct, \theta\right) d\theta, \tag{23}$$

where $g(x, y, t) = x\cos\theta + y\sin\theta + ct$ and $f : \mathbb{R}\times[0, 2\pi] \to \mathbb{R}$ is twice continuously differentiable. Then u satisfies the two-dimensional wave equation

$$u_{tt} = c^2(u_{xx} + u_{yy}). \tag{24}$$

Proof. Differentiating u under the integral sign gives

$$u_{tt} = \int_0^{2\pi} c^2 f_{gg}\left(x\cos\theta + y\sin\theta + ct, \theta\right) d\theta \tag{25}$$

$$u_{xx} = \int_0^{2\pi} \cos^2\theta f_{gg}\left(x\cos\theta + y\sin\theta + ct, \theta\right) d\theta \tag{26}$$

$$u_{yy} = \int_0^{2\pi} \sin^2\theta f_{gg}\left(x\cos\theta + y\sin\theta + ct, \theta\right) d\theta \tag{27}$$

Consequently,

$$c^2(u_{xx} + u_{yy}) = c^2\int_0^{2\pi} \left(\cos^2\theta + \sin^2\theta\right) f_{gg}\left(x\cos\theta + y\sin\theta + ct, \theta\right) d\theta \tag{28}$$

$$= c^2\int_0^{2\pi} f_{gg}\left(x\cos\theta + y\sin\theta + ct, \theta\right) d\theta \; = u_{tt}, \tag{29}$$

where we used $\cos^2\theta + \sin^2\theta = 1$. Hence, $u_{tt} - c^2(u_{xx} + u_{yy}) = 0$, and therefore u satisfies the two-dimensional wave equation identically. □

B WINN ARCHITECTURE AND IMPLEMENTATION DETAILS

B.1 FOURIER FEATURE ENCODING

For long-time domains, g_i can span a large range. We therefore normalize it according to

$$\tilde{g}_i = \frac{g_i}{cT}, \tag{30}$$

and encode it using Fourier features,

$$\gamma(\tilde{g}_i) = [\sin(\omega_1 \tilde{g}_i), \cos(\omega_1 \tilde{g}_i), \ldots, \sin(\omega_K \tilde{g}_i), \cos(\omega_K \tilde{g}_i)]\,, \tag{31}$$

where

$$\omega_k = 2^{k-1}, \qquad k = 1, \ldots, n_{\text{freq}}. \tag{32}$$

The final representation is therefore

$$u_\psi(x, y, t) = \Delta\theta \sum_{i=1}^{N_\theta} w_i f_\psi\left(\gamma(\tilde{g}_i), \theta_i\right). \tag{33}$$

Note that $u_\psi(x, y, t)$ is a sum of functions of g_i. Therefore, each summand satisfies the wave equation, and because the PDE is linear, the sum also satisfies it.

B.2 Three-Dimensional WINN

For the three-dimensional wave equation, we parameterize the solution using the spherical family of propagation directions introduced in (14). Rather than evaluating the surface integral over the sphere exactly, we approximate it using a tensor-product quadrature over the polar and azimuthal angles. Let N_θ and N_ϕ denote the number of quadrature points in the polar and azimuthal directions, respectively. The resulting neural representation is

$$u_\psi(x, y, z, t) = \Delta\theta\Delta\phi \sum_{i=1}^{N_\theta} \sum_{j=1}^{N_\phi} w_{ij} f_\psi\left(g_{ij}, \theta_i, \phi_j\right), \tag{34}$$

where

$$g_{ij} = x \sin\theta_i \cos\phi_j + y \sin\theta_i \sin\phi_j + z \cos\theta_i + ct, \tag{35}$$

and

$$\Delta\theta = \frac{\pi}{N_\theta + 1}, \qquad \Delta\phi = \frac{2\pi}{N_\phi}. \tag{36}$$

The coefficients w_{ij} are trainable parameters and are optimized jointly with the parameters of the neural network.

We initialize the polar and azimuthal angles on uniform grids. To avoid sampling directly at the poles, the polar locations are initialized at

$$\bar{\theta}_i = \frac{i\pi}{N_\theta + 1}, \qquad i = 1, \ldots, N_\theta, \tag{37}$$

while the azimuthal locations are initialized according to

$$\bar{\phi}_j = \frac{2\pi j}{N_\phi}, \qquad j = 0, \ldots, N_\phi - 1. \tag{38}$$

As in the two-dimensional construction, the angular locations are allowed to adapt during training while remaining close to their initial quadrature cells. Specifically,

$$\theta_i = \bar{\theta}_i + \frac{\Delta\theta}{2} \tanh(\alpha_i), \tag{39}$$

and

$$\phi_j = \bar{\phi}_j + \frac{\Delta\phi}{2} \tanh(\beta_j), \tag{40}$$

where α_i and β_j are trainable parameters. Thus, each polar and azimuthal coordinate can move within a bounded neighborhood of its initial location, allowing the quadrature directions to adapt to the target solution while preserving a regular angular parameterization.

Table 4: Problem definitions for the four experimental settings. Here r denotes the radial coordinate in the corresponding spatial dimensions.

Experiment	Domain	c	Parameters	Ground-truth solution
Low-frequency 2D	$[0,1]^2 \times [0,100]$	2	–	$\sin(2x)\sin(2y)\cos(\sqrt{32}\,t)$
High-frequency 2D	$[0,1]^2 \times [0,100]$	1	$k = 10$	$J_0(kr)\sin(kct)$
3D	$[0,1]^3 \times [0,50]$	1	–	$\sinh(x)\sin(5y)\sin(5z)\cos(7t)$
L-shaped 2D	$\left([0,1]^2 \setminus [0,0.5]^2\right) \times [0,100]$	1	$k = 10$	$Y_0(kr)\sin(kct)$

Fourier-feature parameterization. To improve the representation of high-frequency and long-time solutions, we use Fourier features as in B.1. The final representation is therefore

$$u_\psi(x,y,z,t) = \Delta\theta\Delta\phi \sum_{i=1}^{N_\theta}\sum_{j=1}^{N_\phi} w_{ij} f_\psi\left(\gamma(\tilde{g}_{ij}), \theta_i, \phi_j\right), \tag{41}$$

C Experimental Setup

C.1 Problem Definitions

We evaluate WINN on four long-time wave-equation problems designed to vary the spatial dimension, temporal and spatial frequency, and domain geometry. In all experiments, we consider the homogeneous wave equation

$$u_{tt} = c^2 \Delta u, \tag{42}$$

with the wave speed c specified separately for each experiment. The problem definitions are summarized in Table 4.

For all experiments, Dirichlet boundary conditions are imposed on the spatial boundary throughout the temporal domain. Specifically, the boundary values are obtained from the corresponding ground-truth solution and prescribed on

$$\partial\Omega \times [0,T]. \tag{43}$$

At the initial time, both the displacement and velocity are prescribed throughout the spatial domain:

$$u(x,0) = u_{\mathrm{GT}}(x,0), \qquad u_t(x,0) = \partial_t u_{\mathrm{GT}}(x,0), \qquad x \in \Omega. \tag{44}$$

Thus, the initial conditions are specified over the entire spatial domain, while the boundary conditions are specified over the entire temporal interval.

Low-frequency 2D

$$\sin(2x)\sin(2y)\cos(\sqrt{32}t), \tag{45}$$

This is the low-frequency reference problem used to study the long-time failure of conventional PINNs.

High-frequency 2D

$$J_0(kr)\sin(kct), \tag{46}$$

where J_0 denotes the Bessel function of the first kind of order zero. This experiment tests the ability of the methods to represent higher-frequency wave dynamics over a long temporal horizon.

3D

$$\sinh(x)\sin(5y)\sin(5z)\cos(7t), \tag{47}$$

This experiment evaluates the extension of WINN to three-dimensional wave equations and tests the computational cost of the corresponding two-dimensional angular integration.

L-shaped 2D

$$Y_0(kr)\sin(kct), \qquad k = 10, \tag{48}$$

where Y_0 denotes the Bessel function of the second kind of order zero. The non-convex geometry provides a more challenging spatial domain while retaining the same long-time setting.

C.2 Baseline Configurations

We consider five models in our experiments: Vanilla PINN, FF-PINN, Causal PINN, FBPINN, and WINN. Unless otherwise specified, the neural networks use fully connected MLPs with $\tanh$ activations, five hidden layers, and a hidden width of 64.

Vanilla PINN. The Vanilla PINN uses the standard MLP architecture described above, taking (x, y, t) as input and predicting the solution $u(x, y, t)$.

FF-PINN. FF-PINN uses the same MLP architecture as the Vanilla PINN, but applies Fourier feature encoding to the input before the MLP. We use n_{freq} frequency bands. The input is normalized by cT in the temporal dimension before Fourier encoding. We use the same value of n_{freq} as in WINN to make the comparison between the two Fourier-feature-based architectures as consistent as possible.

Causal PINN. Causal PINN uses the same MLP architecture and Fourier feature encoding as FF-PINN, but differs in its training procedure by enforcing a causal ordering over the temporal domain. We use $n_t = 100$ temporal segments and a tolerance of $\text{tol} = 0.8$. We selected these values after testing several hyperparameter combinations and finding this configuration to provide the most stable training. This sensitivity to hyperparameter choices is also noted in Wang et al. (2022).

FBPINN. FBPINN uses the native FBPINN architecture and trainer. In contrast to the other baselines, FBPINN represents the solution using multiple local neural networks, with one network associated with each subdomain; the windowed outputs of these networks are combined to form the global solution (Moseley et al., 2023). For each subdomain network, we use five hidden layers with a width of 64 and $\tanh$ activations. The configuration follows the native Moseley et al. (2023) implementation and the configuration used in Hao et al. (2023).

WINN. WINN uses the same underlying MLP architecture as FF-PINN, with five hidden layers, width 64, and $\tanh$ activations. As described in Section 3.3, in 2D, WINN evaluates the MLP along characteristic directions parameterized by θ and numerically integrates over these directions. We use $N_\theta = 32$ characteristic directions with learnable angular offsets. In 3D, we use $N_\theta = N_\phi = 12$. The Fourier encoding uses the same number of frequency bands as FF-PINN: $n_{\text{freq}} = 10$ for 2D and $n_{\text{freq}} = 8$ for 3D. The characteristic coordinate is normalized by cT before Fourier encoding. Thus, FF-PINN and WINN have the same MLP architecture, Fourier-feature dimensionality, and input normalization, differing only in how the solution is represented.

C.3 Optimization and Training

Let $\mathcal{B}$ denote the set of boundary points, let $\mathcal{I}_0$ denote the initial time slice, and let $\mathcal{D}$ denote the whole space-time domain. The optimization objective for all baseline methods is

$$\mathcal{L} = \lambda_{\text{bc}}\mathcal{L}_{\text{bc}} + \lambda_{\text{ic}}\mathcal{L}_{\text{ic}} + \lambda_{\text{vel}}\mathcal{L}_{\text{vel}} + \lambda_{\text{pde}}\mathcal{L}_{\text{pde}}, \tag{49}$$

where

$$\mathcal{L}_{\mathrm{bc}} = \frac{1}{|\mathcal{B}|} \sum_{(x,y,t)\in\mathcal{B}} \left|u_\psi(x,y,t) - u(x,y,t)\right|^2, \tag{50}$$

$$\mathcal{L}_{\mathrm{ic}} = \frac{1}{|\mathcal{I}_0|} \sum_{(x,y)\in\mathcal{I}_0} \left|u_\psi(x,y,0) - u(x,y,0)\right|^2, \tag{51}$$

$$\mathcal{L}_{\mathrm{vel}} = \frac{1}{|\mathcal{I}_0|} \sum_{(x,y)\in\mathcal{I}_0} \left|\frac{\partial u_\psi}{\partial t}(x,y,0) - \frac{\partial u}{\partial t}(x,y,0)\right|^2, \tag{52}$$

$$2D: \mathcal{L}_{\mathrm{pde}} = \frac{1}{|\mathcal{D}|} \sum_{(x,y,t)\in\mathcal{D}} \left|\frac{\partial^2 u_\psi}{\partial t^2}(x,y,t) - c^2\left(\frac{\partial^2 u_\psi}{\partial x^2}(x,y,t) + \frac{\partial^2 u_\psi}{\partial y^2}(x,y,t)\right)\right|^2, \tag{53}$$

$$\begin{aligned} 3D: \mathcal{L}_{\mathrm{pde}} = \frac{1}{|\mathcal{D}|} \sum_{(x,y,z,t)\in\mathcal{D}} \Bigg|&\frac{\partial^2 u_\psi}{\partial t^2}(x,y,z,t) \\ &-c^2\left(\frac{\partial^2 u_\psi}{\partial x^2}(x,y,z,t) + \frac{\partial^2 u_\psi}{\partial y^2}(x,y,z,t) + \frac{\partial^2 u_\psi}{\partial z^2}(x,y,z,t)\right)\Bigg|^2. \end{aligned} \tag{54}$$

For Vanilla PINN and FF-PINN, the weighting coefficients λ_{bc}, λ_{ic}, λ_{vel}, and λ_{pde} are updated adaptively during training using the learning-rate annealing (LRA) strategy of Wang et al. (2020a). LRA adjusts the relative weights of the individual objectives based on their gradient magnitudes, reducing the tendency for any single loss component to dominate the optimization.

Since WINN satisfies the wave equation by construction, its training objective does not contain a PDE residual term. Instead, WINN is trained solely by enforcing the boundary and initial conditions:

$$\mathcal{L}_{\mathrm{WINN}} = \lambda_{\mathrm{bc}}\mathcal{L}_{\mathrm{bc}} + \lambda_{\mathrm{ic}}\mathcal{L}_{\mathrm{ic}} + \lambda_{\mathrm{vel}}\mathcal{L}_{\mathrm{vel}}. \tag{55}$$

For Causal PINN, which progressively learns the temporal domain, we follow Wang et al. (2022) and set $\lambda_{\mathrm{bc}} = \lambda_{\mathrm{ic}} = \lambda_{\mathrm{vel}} = 10$ and $\lambda_{\mathrm{pde}} = 1$. This weighting emphasizes the boundary and initial conditions, allowing the solution to first satisfy these constraints before propagating through the domain according to the PDE.

Training is performed in two stages. We first optimize the network using the SOAP optimizer (Vyas et al., 2025) with a learning rate of 3×10^{-3}, $\beta_1 = 0.99$, $\beta_2 = 0.999$, weight decay of 0.01, and a preconditioning frequency of 2, following Wang et al. (2025). For the 2D experiments, SOAP is run for 10K steps, while for the 3D experiment it is run for 15K steps. This is followed by a 200-step L-BFGS (Liu & Nocedal, 1989) refinement stage. During the L-BFGS stage, we use a fixed set of 10,000 boundary points for the 2D experiments and 20,000 boundary points for the 3D experiment.

For all models, we use $N_{\mathrm{Boundary}} = 2048$ boundary and initial-condition points, equally divided among the three corresponding loss terms during the SOAP training stage. For Vanilla PINN, FF-PINN, and Causal PINN, we additionally use 10K interior points. WINN does not require interior points, whereas FBPINN is trained with its native trainer using the same configuration as PINNacle. We use $N_{\mathrm{test}} = 10K$ test points to evaluate the relative L_2 error. All experiments use double-precision arithmetic and were conducted on an NVIDIA A40 GPU.

D Additional Results

D.1 Increasing the Number of Collocation Points Does Not Prevent Failure

One might hypothesize that the failure is caused by increasing the temporal domain while keeping the number of interior collocation points fixed. To test this, instead of using a fixed number of interior points, $N_{PDE} = 10K$, we increase their number proportionally with T. Specifically, we use $10K$, $50K$, and $100K$ points for $T = 1, 5, 10$, respectively. As shown in Figure 4 for FF-PINN, increasing the number of collocation points does not prevent the failure.

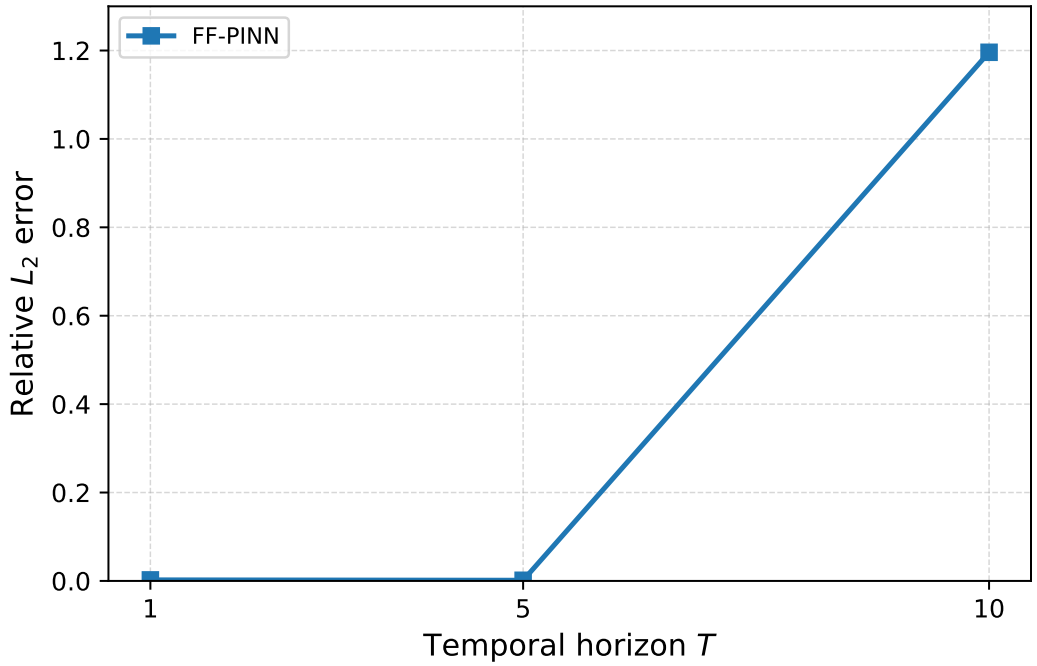

Figure 4: Relative L_2 error of FF-PINN for increasing numbers of interior collocation points: $10K$, $50K$, and $100K$ for $T = 1$, 5, and 10, respectively. Increasing the number of collocation points proportionally with the temporal domain does not prevent the long-time failure.

Table 5: Standard deviations of the relative L_2 error and total training time across three random seeds for each method and experimental setting.

	WINN		PINN		FF-PINN		Causal PINN		FBPINN	
Experiment	Rel. L_2 Std.	Time Std.	Rel. L_2 Std.	Time Std.	Rel. L_2 Std.	Time Std.	Rel. L_2 Std.	Time Std.	Rel. L_2 Std.	Time Std.
Low-frequency 2D	0.20 %	0.40	0.097 %	0.34	14.0 %	0.72	0.047 %	4.2	0.64 %	16.9
High-frequency 2D	0.016 %	0.12	0.019 %	0.47	2.3 %	0.40	1.8 %	1.8	0.85 %	8.2
3D	0.41 %	1.7	0.21 %	0.96	2.4 %	4.8	0.58 %	4.3	0.54 %	84.6
L-shaped 2D	0.29 %	3.0	0.25 %	0.46	18.5 %	1.1	0.54 %	11.6	2.8 %	59.4

D.2 Standard Deviations Across Random Seeds

Table 5 reports the standard deviations of the relative L_2 error and total training time across the three random seeds used for each method and experimental setting. The results show that WINN generally exhibits low variability in relative L_2 error across random seeds, indicating robust performance across different initializations.

D.3 Qualitative Results

D.3.1 Low-frequency 2D

Figure 5 visualizes the solution obtained by WINN on the low-frequency 2D problem together with the two strongest-performing baselines.

D.3.2 L-shaped 2D

Figure 6 visualizes the solution obtained by WINN on the L-shaped 2D problem together with the two strongest-performing baselines.

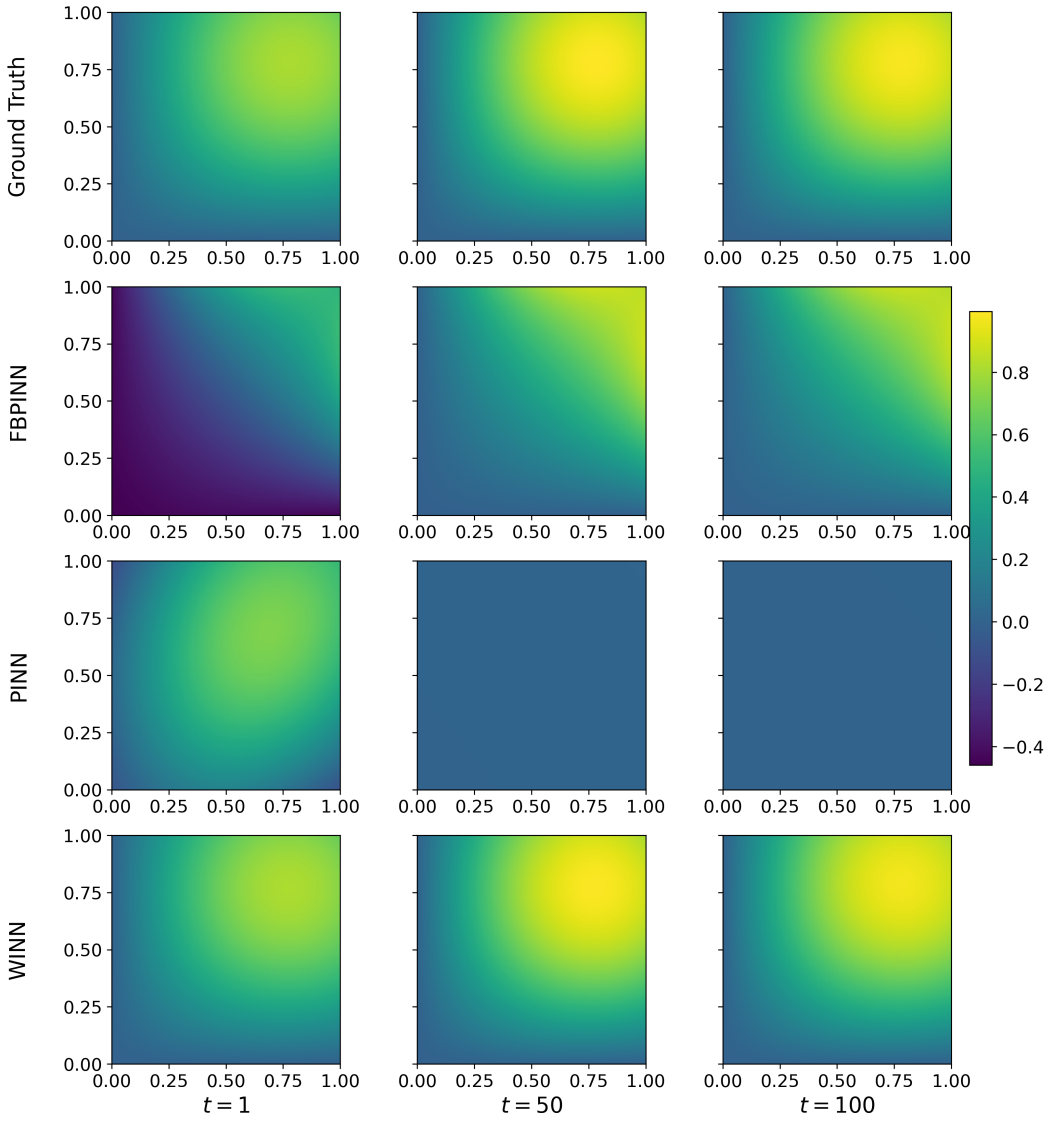

Figure 5: Qualitative comparison of WINN and the two strongest-performing baselines on the low-frequency 2D problem at representative time points.

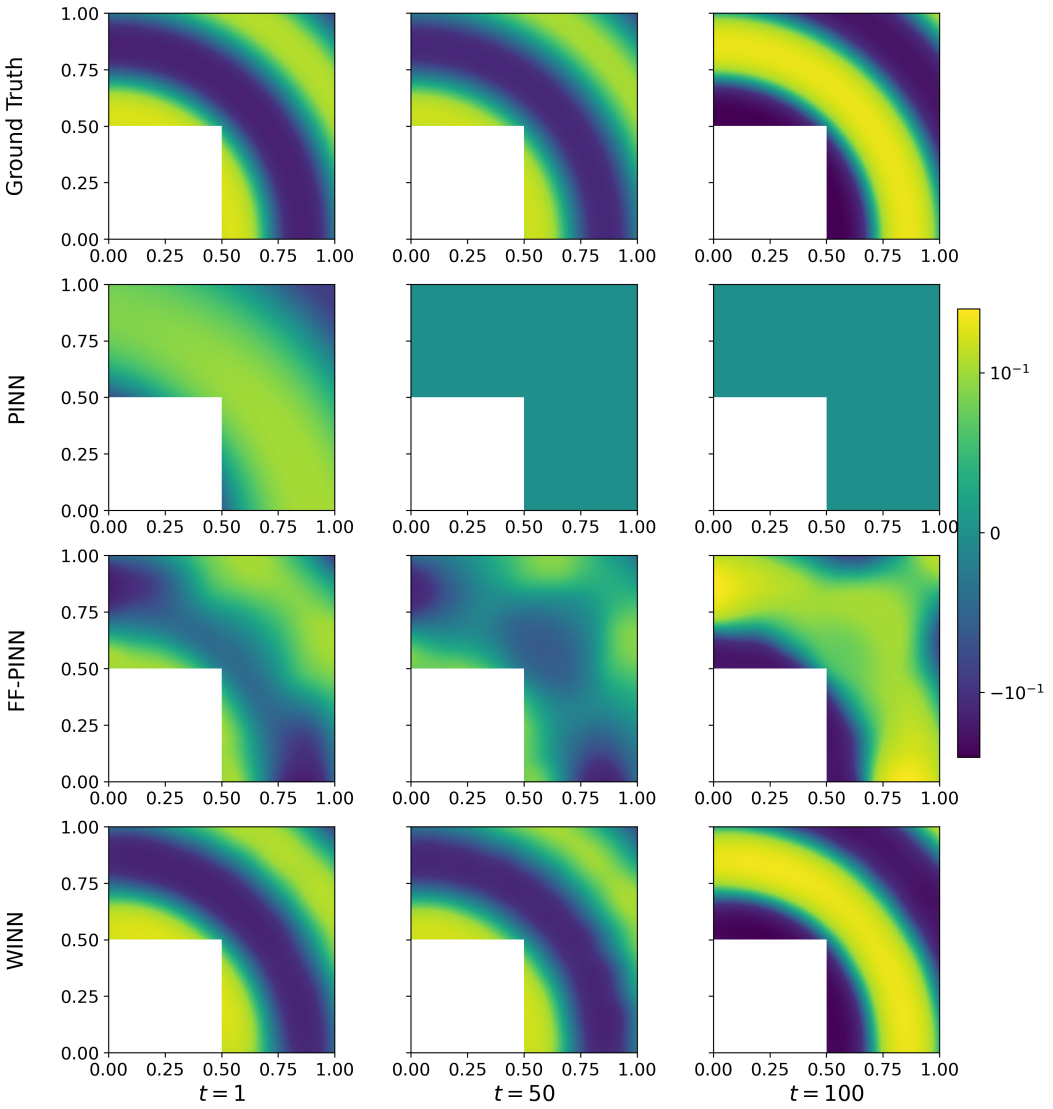

Figure 6: Qualitative comparison of WINN and the two strongest-performing baselines on the L-shaped 2D problem at representative time points.