Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Physics-informed neural networks Electromagnetic Wave Reflection at a Dielectric Interface

Aakarsh Sinha
Indian Institute of Technology Roorkee
Mayank Goswami
mayankfph@iitr.ac.in

Indian Institute of Technology Roorkee https://orcid.org/0000-0001-8829-9980

Method Article

Keywords: PINN; XPINN; domain decomposition; wave equation; dielectric interface; Fresnel coefficients

Posted Date: November 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11710084/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

PINN and XPINN Models for Electromagnetic Wave Reflection at a Dielectric Interface

Aakarsh Sinha
Department of Physics, IIT Roorkee
aakarsh_s@ph.iitr.ac.in

Mayank Goswami
Department of Physics, IIT Roorkee
mayank.goswami@ph.iitr.ac.in

November 2026

Abstract

Physics-informed neural networks (PINNs) solve partial differential equations by minimizing the equation residual together with initial- and boundary-condition errors, without interior solution data. We apply PINNs to the electromagnetic wave equation and obtain relative L^2 errors of 3.056×10^{-3} and 1.714×10^{-2} on one- and two-dimensional standing-wave benchmarks. We then study a Gaussian pulse at normal incidence on a dielectric interface ($n_1 = 1$, $n_2 = 2$) and compare, under identical data, schedule and initial weights, a two-network PINN with an XPINN-style model that adds PDE-residual continuity at the interface and adaptive activations. A single-stage PINN failed to learn wave propagation ($R + T = 0.862$), so both models were trained in two stages. The PINN recovers $R = 0.105$ and $T = 0.856$ ($R + T = 0.961$), and the XPINN-style model $R = 0.116$ and $T = 0.867$ ($R + T = 0.983$), against the Fresnel values 1/9 and 8/9. The XPINN-style model reduces the space–time relative L^2 error from 3.31×10^{-2} to 2.79×10^{-2}, mainly by reducing the error in the transmitting medium by a factor of 2.6, at twice the training cost. Because a repeated run of the PINN differed from this one by more than the PINN–XPINN gap in $R + T$, the overall advantage requires confirmation over several random seeds.

Keywords: PINN; XPINN; domain decomposition; wave equation; dielectric interface; Fresnel coefficients.

1 Introduction

Physics Informed Neural Networks (PINNs) are represented by a neural network which obtains derivatives in the governing equation with the help of automatic differentiation, and is trained by the residual of the equation and the initial and final Boundary condition. It is a mesh free way to solve differential equations and does not require interior solution data which in some cases might be hard or very expensive to obtain. In this project we first attempt to model 1D and 2D waves in a perfectly insulated cavity and then try to model a wave interacting with a change in medium i.e. getting reflected and transmitted. In the second case we diagnose a failure of a single - stage training of the PINN and try to resolve it with two stage schedule. In the end we also try to compare two-network PINN and domain decomposition method (XPINN-style model) under controlled conditions.

2 Background: PINNs and XPINNs

2.1 Physics-informed neural networks (PINNs)

A PINN [1] is a neural network that takes the coordinates (x, t) as input and returns an approximation $E_{\mathrm{NN}}(x, t)$ of the field. Because the network is a smooth function, its derivatives $\partial_t^2 E_{\mathrm{NN}}$ and $\partial_x^2 E_{\mathrm{NN}}$

can be computed exactly by automatic differentiation, without a grid. This gives the *residual* of the wave equation,

$$F = \partial_t^2 E_{\mathrm{NN}} - c^2\, \partial_x^2 E_{\mathrm{NN}}, \tag{1}$$

which is zero if the network is an exact solution. The network is trained by minimizing

$$\mathcal{L} = \underbrace{\overline{F^2}}_{\text{obey the equation}} + w \underbrace{\overline{(E_{\mathrm{NN}} - E_{\mathrm{given}})^2}}_{\text{match initial and boundary data}}, \tag{2}$$

where the bars denote averages over sample points: the first over points inside the domain, the second over points on the boundary and at $t = 0$. Both terms are needed. The equation alone has many solutions (including $E = 0$), and the initial and boundary data pick out the correct one.

PINNs need no solution data inside the domain and give a smooth formula for $E(x, t)$. However, training can be difficult when the solution changes sharply or must be followed over a long time [3, 4]. A single network also struggles to represent a kink, such as the jump in $\partial_x^2 E$ at a material interface.

2.2 Extended PINNs (XPINNs)

An XPINN [7] splits the domain into pieces and gives each piece its own network. In our problem there are two pieces, one per material: $E_1(x, t)$ for $x < 0$ and $E_2(x, t)$ for $x > 0$. Each network is trained on the loss (2) in its own region, with its own wave speed. The two networks are then joined at the interface $x = 0$ by extra loss terms. XPINN uses two:

- *Matching values:* E_1 and E_2 should agree at the interface.
- *Matching residuals:* $F_1 = F_2$ at the interface, i.e. each network should obey the wave equation in the same way where they meet.

Other conditions can be added when the physics requires them [7]; here we add continuity of $\partial_x E$, which follows from Maxwell's equations. The interface terms are also how information reaches a region that has no data of its own [8]. This matters here, because material 2 receives the pulse only through the interface.

XPINNs often also use *adaptive activation functions* [9]: each layer computes $\tanh(5\, a\, h)$ instead of $\tanh(h)$, where a is a number the network learns during training. This can speed up training.

3 Physical problems

In a source-free, non-magnetic medium, Maxwell's equations give $\partial_t^2 E = c^2 \nabla^2 E$ [2]; we use normalized units ($c = 1$ in vacuum). At a perfect conductor the tangential field vanishes (by the use of Maxwell's equation), so $E = 0$ on the walls, and both E and $\partial_t E$ are prescribed at $t = 0$.

Benchmarks. With $E_t = 0$ at $t = 0$ and $E = 0$ on the walls, the 1D problem on $x, t \in [0, 1]$ has solution $E = \sin(\pi x)\cos(\pi t)$, and the 2D TM_z cavity mode on the unit square has $E = \sin(\pi x)\sin(\pi y)\cos(\sqrt{2}\,\pi t)$.

Dielectric interface. On $x \in [-1, 1]$ the index is n_1 for $x < 0$ and n_2 for $x > 0$, with speeds $c_i = 1/n_i$. Continuity of tangential E and H requires E and $\partial_x E$ to be continuous at $x = 0$, while $\partial_x^2 E$ jumps by c_1^2/c_2^2. The Fresnel coefficients are

$$r = \frac{n_1 - n_2}{n_1 + n_2}, \quad \tau = \frac{2n_1}{n_1 + n_2}, \quad R = r^2, \quad T = \frac{n_2}{n_1}\tau^2, \quad R + T = 1, \tag{3}$$

so $n_1 = 1$, $n_2 = 2$ gives $r = -1/3$ (a phase reversal), $\tau = 2/3$, $R = 1/9$, $T = 8/9$. A Gaussian pulse $f(\xi) = \exp[-(\xi - x_0)^2/2\sigma^2]$ with $x_0 = -0.5$, $\sigma = 0.1$ reaches the interface at $t = 0.5$. The reference solution is $f(x - c_1 t) + r f(-x - c_1 t)$ for $x < 0$ and $\tau f(c_1(x/c_2 - t))$ for $x > 0$. Note that the field in material 2 is essentially zero until $t = 0.5$, so that region receives no pulse information from its initial condition: everything it must learn enters through the interface.

4 Methods

Networks and loss. Each network is an MLP (5 hidden layers of 64 tanh units, Xavier initialization) mapping coordinates to E. For the benchmarks the loss is $\mathrm{MSE}(E_{tt} - c^2\nabla^2 E) + 10\,\mathcal{L}_{\mathrm{IC}} + 10\,\mathcal{L}_{\mathrm{BC}}$. The interface problem uses one network per material, $E^{(1)}$ on $[-1, 0]$ and $E^{(2)}$ on $[0, 1]$, each with its own residual $F_q = E_{tt}^{(q)} - c_q^2 E_{xx}^{(q)}$, initial and wall terms; the residual, velocity and $\partial_x E$ terms are scaled by σ so all terms are of the similar order i.e one. The two models differ only in the interface coupling at 400 points on $x = 0$ and in the activation:

$$\mathcal{L}_{\mathrm{IF}}^{\mathrm{PINN}} = 10\big[\mathrm{MSE}(E^{(1)} - E^{(2)}) + \mathrm{MSE}(\sigma\partial_x E^{(1)} - \sigma\partial_x E^{(2)})\big], \tag{4}$$

$$\mathcal{L}_{\mathrm{IF}}^{\mathrm{XPINN}} = 20\sum_q \mathrm{MSE}(E^{(q)} - E_{\mathrm{avg}}) + 10\ \mathrm{MSE}(\sigma^2 F_1 - \sigma^2 F_2) + 10\ \mathrm{MSE}(\sigma\partial_x E^{(1)} - \sigma\partial_x E^{(2)}). \tag{5}$$

The XPINN-style loss uses the average-solution and residual-continuity terms of [7], plus $\partial_x E$ continuity as the physical flux-type condition. Since $\sum_q \mathrm{MSE}(E^{(q)} - E_{\mathrm{avg}}) = \frac{1}{2}\,\mathrm{MSE}(E^{(1)} - E^{(2)})$, the weight 20 keeps solution continuity equal to the PINN's, so the new term is residual continuity, which holds for the exact solution ($F_1 = F_2 = 0$). The XPINN-style networks also use layer-wise adaptive activations $\tanh(na_k h)$ with $n = 5$ and trainable a_k initialized to $1/n$ [9], adding 10 parameters in total (33,804 versus 33,794). Unlike the original XPINN, we use two spatial subdomains, one joint optimizer, equal architectures and no slope recovery.

Data and training. Collocation points come from scrambled Sobol sequences (1D: 8,192; 2D: 16,384; interface: 8,192 per subdomain), with separate initial, wall and interface points; test grids labelled with the exact solution are used only for evaluation. Training uses Adam [10] (learning rate 10^{-3}, halved each quarter) followed by L-BFGS [11]: $5{,}000 + 500$ iterations in 1D and $8{,}000 + 300$ in 2D. Each interface model is trained in two stages of $10{,}000 + 500$ iterations: first on $t \in [0, 0.5]$, then from those weights on $t \in [0, 1]$ at learning rate 5×10^{-4}. Both interface models share data, schedule, shared loss weights and initial weights (seed 0). Runs used PyTorch on a Google Colab GPU.

Metrics. We report the relative L^2 error $\|E_{\mathrm{NN}} - E_{\mathrm{exact}}\|_2 / \|E_{\mathrm{exact}}\|_2$ on the test grid (for the interface: $2{,}001 \times 41$ points in (x, t)), the RMS of the scaled residual $\sigma^2 F_q$ on 4,000 fresh random points per subdomain, and the maximum mismatches of E, $\partial_x E$ and $\sigma^2 F$ across $x = 0$. At $t = 1$, r and τ are read from the pulse peaks, and R and T from the pulse energies $n_i^2 \int E^2 \mathrm{d}x$ normalized by the incident energy $n_1^2 \sigma\sqrt{\pi}$. Applied to the reference solution, this procedure reproduces Eq. (3) to four decimals.

5 Results

5.1 Benchmarks

The PINN ia able to achieve relative L^2 errors of 3.056×10^{-3} in 1D and 1.714×10^{-2} in 2D. In 1D (Fig. 1) the exact solution and the one predicted by PINN are visually identical; the maximum pointwise error is about 5×10^{-3} and the error is spread over the whole domain (Fig. 4, left). In 2D (Fig. 2) the expected shapes are faithfully reproduced , but the error grows with time: the maximum error rises from about 7×10^{-3} at $t = 0$ to about 3×10^{-2} at $t = 1$, where it concentrates near the walls, and the normalized error grows by roughly an order of magnitude over the interval (Fig. 4, right). The PDE loss is the largest term at the end of training in both cases (Fig. 3), reaching about 2×10^{-4} in 1D and 10^{-3} in 2D, and in 2D it is still decreasing.

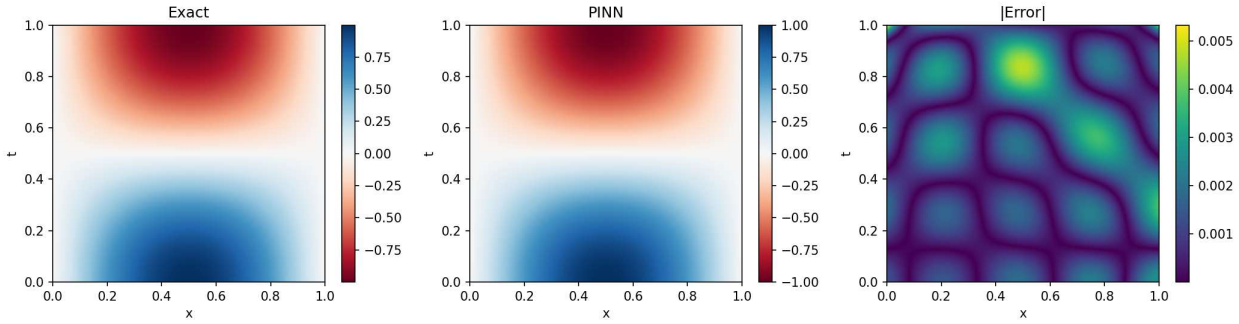


Figure 1: 1D standing wave: exact field, PINN prediction and absolute error.

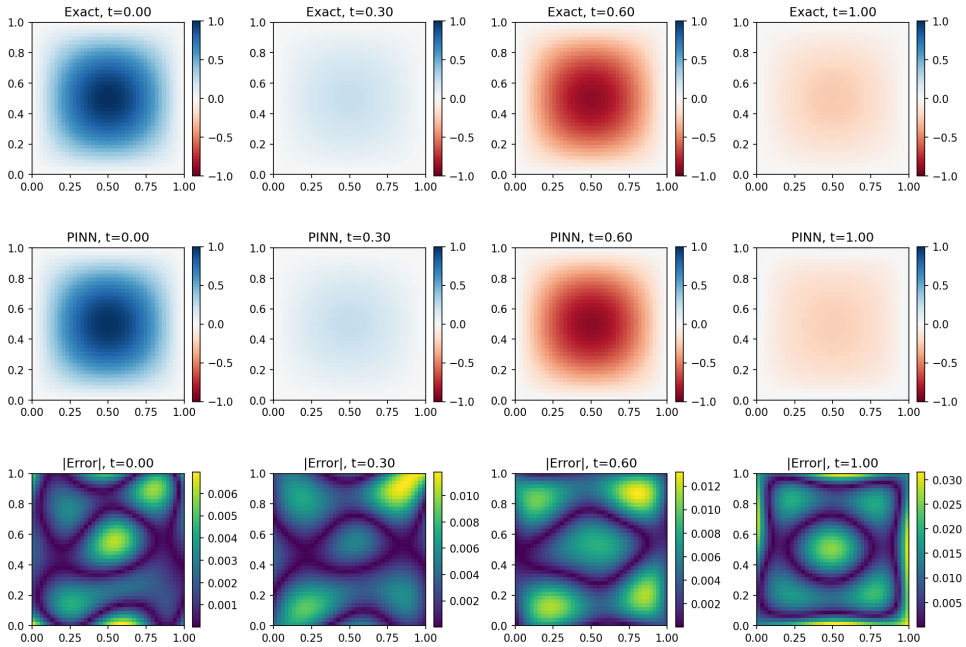
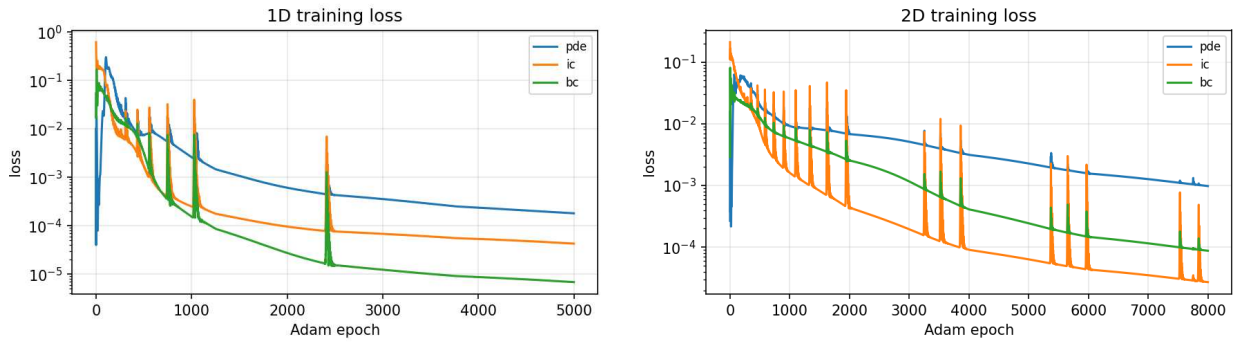


Figure 2: 2D cavity mode at $t = 0,\ 0.3,\ 0.6,\ 1$: exact (top), PINN (middle) and absolute error (bottom; colour scales differ between panels).



(a) 1D. (b) 2D.

Figure 3: Benchmark training losses (unweighted terms, Adam phase).

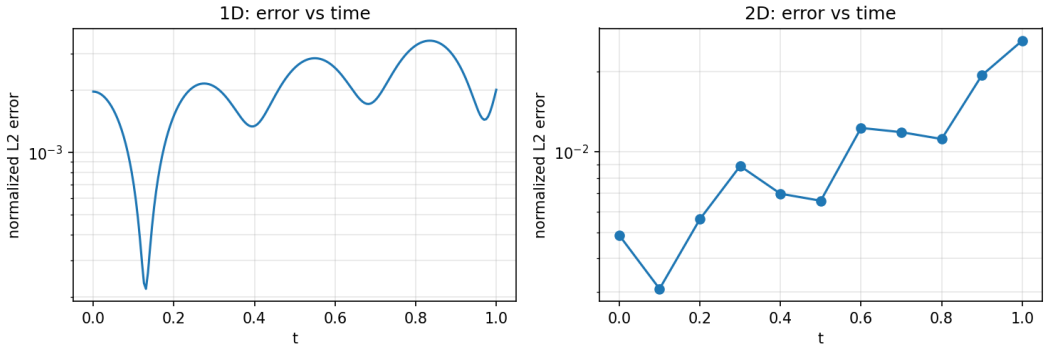


Figure 4: Normalized L^2 error versus time for the 1D (left) and 2D (right) benchmarks.

5.2 Interface: single-stage training

A preliminary single-stage PINN on $t \in [0, 1]$ gave $r = -0.427$, $\tau = 0.414$ and $R + T = 0.862$. Its initial, wall and interface losses were small ($\lesssim 10^{-5}$) but its PDE loss stayed at 7.9×10^{-3}, and its error was already large before $t = 0.5$: it fitted the constraints without propagating the pulse. Training on $t \leq 0.5$ alone reached a relative error of 1.8×10^{-2}, which motivated the two-stage schedule.

5.3 Interface: PINN versus XPINN

Both two-stage models reproduce the incident, reflected and transmitted pulses, including the sign reversal of the reflected pulse (Fig. 5). Table 1 gives the recovered coefficients and Table 2 the error and other metrics.

Table 1: Reflection and transmission at $t = 1$ ($n_1 = 1$, $n_2 = 2$; energy-based R, T). Relative errors with respect to Fresnel in parentheses.

Model	r	τ	R	T	$R+T$
Fresnel theory	-0.3333	0.6667	0.1111	0.8889	1.0000
PINN, single-stage	-0.427	0.414	0.412	0.451	0.862
PINN	-0.3232 (3.1%)	0.6576 (1.4%)	0.1053 (5.2%)	0.8555 (3.8%)	0.9608 (3.9%)
XPINN-style	-0.3380 (1.4%)	0.6616 (0.8%)	0.1157 (4.1%)	0.8670 (2.5%)	0.9827 (1.7%)

Table 2: Interface metrics for the two-stage models. "Ratio" is PINN divided by XPINN-style (values above 1 favour XPINN-style).

Metric	PINN	XPINN-style	Ratio
Relative L^2, all (x, t)	3.31×10^{-2}	2.79×10^{-2}	1.19
Relative L^2, material 1 ($x < 0$)	2.58×10^{-2}	2.91×10^{-2}	0.88
Relative L^2, material 2 ($x > 0$)	5.50×10^{-2}	2.13×10^{-2}	2.58
Relative L^2, $t \leq 0.5$	1.08×10^{-2}	1.56×10^{-2}	0.69
Relative L^2, $t > 0.5$	6.35×10^{-2}	4.87×10^{-2}	1.30
Max. absolute error	3.76×10^{-2}	3.22×10^{-2}	1.17
RMS scaled residual, material 1	4.45×10^{-3}	4.04×10^{-3}	1.10
RMS scaled residual, material 2	1.72×10^{-3}	9.82×10^{-4}	1.75
Max. $\lvert\Delta E\rvert$ at $x = 0$	9.5×10^{-4}	7.0×10^{-4}	1.35
Max. $\lvert\Delta \partial_x E\rvert$ at $x = 0$	6.3×10^{-3}	9.3×10^{-3}	0.67
Max. $\lvert\Delta(\sigma^2 F)\rvert$ at $x = 0$	4.2×10^{-2}	6.3×10^{-4}	67.6
Amplitude-based R / T	0.1044 / 0.8649	0.1143 / 0.8753	–
Training time (s)	782	1574	0.50

Accuracy. The XPINN-style model has a 16% lower relative L^2 error overall and recovers every Fresnel quantity more accurately, with $R + T = 0.983$ against 0.961. The gain is not uniform. In

material 2 its error is 2.6 times smaller, and after the pulse reaches the interface it is 23% smaller; in material 1, and before the interaction ($t \leq 0.5$), the PINN is more accurate: its error is 11% and 31% lower, respectively. Fig. 7a shows the same picture in time: the XPINN-style error is higher than the PINN's until $t \approx 0.48$ and lower afterwards.

Where the errors are. The error maps (Fig. 6) show that both models have their largest error in material 1 just left of the interface during the interaction ($t \approx 0.6$–0.85), where the reflected pulse forms; the peak is about 0.035 for the PINN and 0.032 for the XPINN-style model. The PINN also has a broad error band in material 2 along the transmitted pulse, which appears in Fig. 5 as a small negative undershoot behind the transmitted pulse at $t = 0.75$ and 1; this band is largely absent for the XPINN-style model. The XPINN-style model has instead an error patch in material 1 before the interaction ($x \approx -0.6$, $t \approx 0.3$), consistent with its larger $t \leq 0.5$ error.

Physics and interface. The XPINN-style model has a lower residual in both materials, by 43% in material 2. As expected from its loss, it matches the residual across the interface 68 times more closely. Both models keep E and $\partial_x E$ continuous to below 10^{-2} (typical $|\partial_x E|$ is 6–8), with the PINN slightly better on $\partial_x E$.

Energy. Amplitude- and energy-based estimates agree to within about 1% for both models (Table 2), so the remaining deficit in $R + T$ reflects the predicted pulses themselves rather than the measurement. The PINN slightly underestimates both pulses, whereas the XPINN-style model slightly overestimates the reflected pulse ($|r|$ is 1.4% high) and underestimates the transmitted one, so its closer $R + T$ partly results from compensating errors.

Training. Both models reach small constraint losses ($\leq 3 \times 10^{-7}$ after L-BFGS), with the PDE term largest at the end (Fig. 8; final PDE loss 2.4×10^{-5} for the PINN and 1.8×10^{-5} for the XPINN-style model). In Stage 1 the XPINN-style PDE loss plateaus near 10^{-2} for about 4,500 epochs before dropping, which may explain its larger error before the interaction. The XPINN-style model took twice as long to train (1,574 s against 782 s).

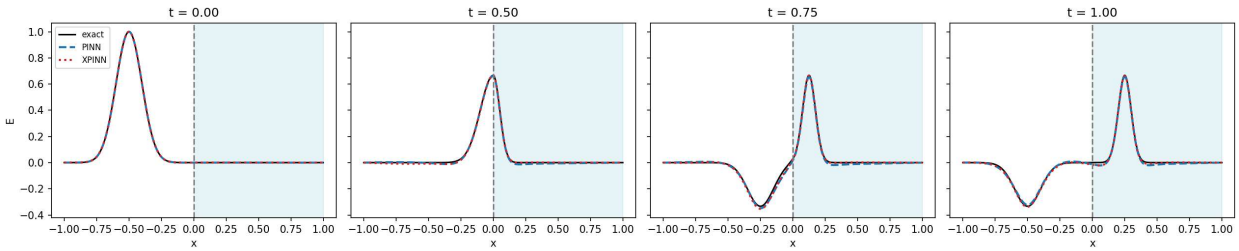


Figure 5: Field at $t = 0$, 0.5, 0.75, 1: reference (solid), PINN (dashed) and XPINN-style (dotted); shaded region is material 2.

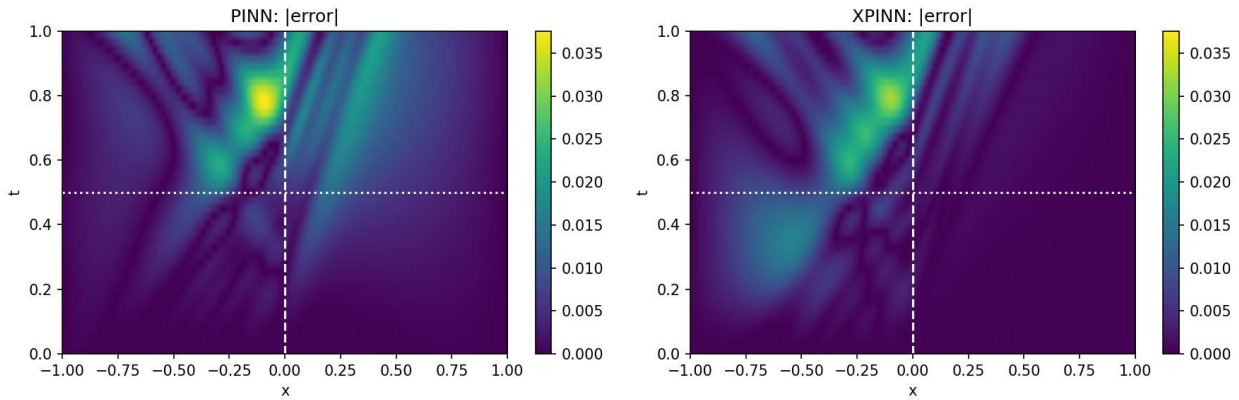
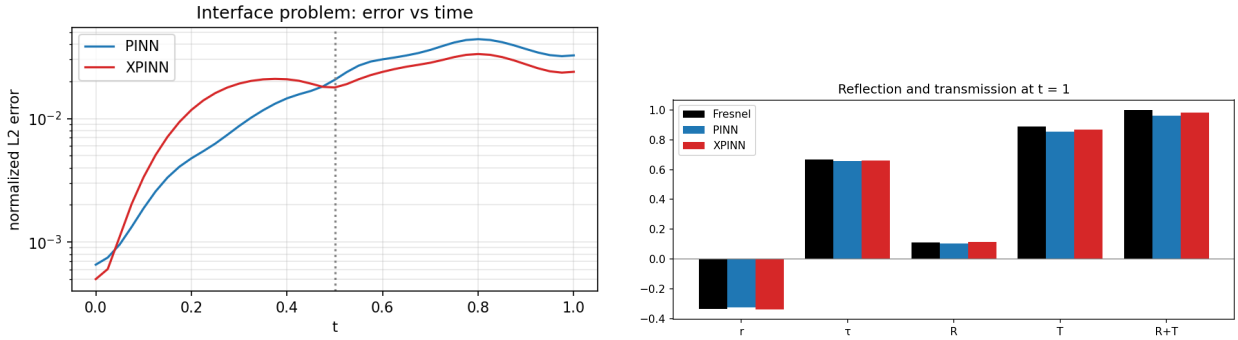


Figure 6: Space–time absolute error on a common colour scale. Dashed line: interface; dotted line: $t = 0.5$.



(a) Normalized L^2 error versus time.

(b) r, τ, R, T, $R+T$ at $t = 1$.

Figure 7: Comparison of the interface models.

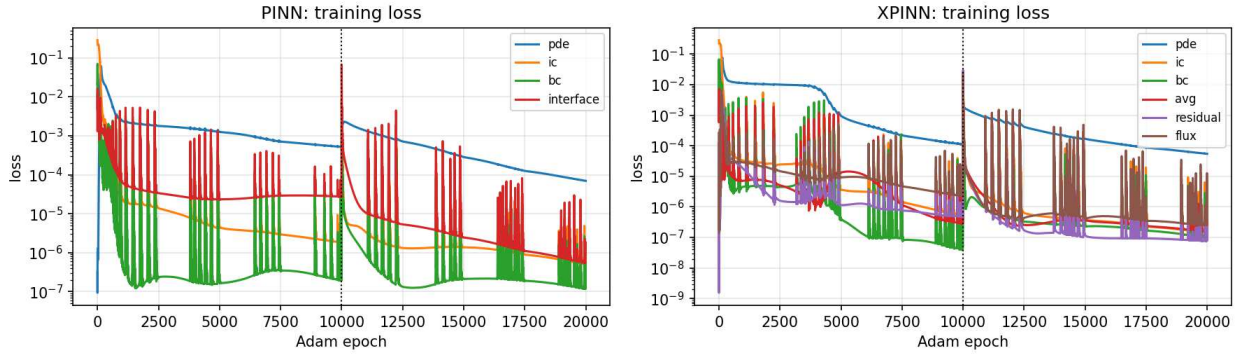


Figure 8: Training losses over both stages (unweighted terms; the dotted line marks the start of Stage 2).

6 Discussion

Two-stage training. The single-stage failure, small constraint losses with a large residual, is consistent with known PINN difficulties on time-dependent problems [4]. Learning free propagation first gives a starting point for the interaction, after which both models recover the Fresnel coefficients to within a few percent. We did not test this explanation directly.

Why the XPINN-style model helps in material 2. The largest gain is in material 2, the subdomain whose solution must be learned entirely through the interface, since its initial field carries no pulse. Residual continuity adds information there: it requires the transmitted field at $x = 0^+$ to satisfy the wave equation consistently with the field arriving from material 1, rather than only matching its value and slope. This matches the observation in [8] that interface conditions are especially important for subdomains without training data. The lower residual in material 2 and the disappearance of the PINN's undershoot behind the transmitted pulse support this interpretation. The XPINN-style model is, however, less accurate in material 1 and before the interaction, so the extra interface terms trade some accuracy in the incident region for accuracy in the transmitted region.

Which ingredient matters. The XPINN-style model differs from the PINN in two ways, residual continuity and adaptive activations, and the ablation separating them (XPINN-noadapt in the notebook) was not run. The improvements in material 2 and in interface-residual matching are consistent with residual continuity, but we cannot attribute them to it alone.

Limitations. One seed and one index pair; 1D, normal incidence; soft constraints; normalized units; no ablation of the adaptive activation. For simple forward problems, mesh-based solvers are expected to be faster and more accurate than either network; the case for the neural approach rests on mesh-free evaluation, smooth differentiable solutions and inverse problems, which were not studied here.

7 Conclusion

PINNs reproduced the 1D and 2D standing-wave benchmarks with relative L^2 errors of 3.056×10^{-3} and 1.714×10^{-2}, with the 2D error growing in time. For a Gaussian pulse at a dielectric interface, single-stage training failed to learn wave propagation, and a two-stage schedule was needed. With it, a two-network PINN recovered the Fresnel coefficients to within 1.4–5.2% ($R + T = 0.961$), and an XPINN-style model with residual continuity and adaptive activations to within 0.8–4.1% ($R + T = 0.983$), with a 2.6 times lower error in the transmitting medium but higher error in the incident medium and twice the training cost. Future work should repeat the comparison over several seeds and refractive indices, run the activation ablation, and extend the study to oblique incidence (Snell's law), hard constraints and inverse recovery of n_2.

In this work we:

1. **Built the datasets** for the 1D and 2D wave equation: collocation points (Sobol sampling), initial, boundary and test points, with the analytical solution used only for testing.
2. **Trained PINNs** on 1D and 2D standing-wave benchmarks, reaching relative L^2 errors of L^2 errors of 3.056×10^{-3} and 1.714×10^{-2}, and found that the 2D error grows with time.
3. **Set up a dielectric interface problem** with a closed-form solution, and verified that our measurement method recovers the Fresnel coefficients exactly from it.
4. **Diagnosed a training failure**: a single-stage PINN fitted the initial, boundary and interface data but not the wave equation, losing 14% of the energy ($R + T = 0.862$).
5. **Fixed it with two-stage training** (first before, then after the pulse reaches the interface), which brought the PINN to $R + T = 0.961$ and the reflection coefficient to within 3.1% of theory.
6. **Built an XPINN-style model** with residual continuity and adaptive activations, and compared it with the PINN under identical conditions. It gave $R + T = 0.983$, a 16% lower overall error and a 2.6 times lower error in the transmitting medium, at twice the training cost.

Together, these results show that neural networks trained only on Maxwell's wave equation and its initial, boundary and interface conditions, with no solution data, can reproduce reflection and transmission at a dielectric interface to within a few percent of the Fresnel predictions,

Code and data. [**Google Drive Repository.**]

References

[1] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.

[2] D. J. Griffiths, *Introduction to Electrodynamics*, 4th ed., Cambridge University Press, 2017.

[3] S. Wang, Y. Teng, P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks, *SIAM J. Sci. Comput.* 43 (2021) A3055–A3081.

[4] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, M. W. Mahoney, Characterizing possible failure modes in physics-informed neural networks, *NeurIPS* 34 (2021).

[5] S. Wang, S. Sankaran, P. Perdikaris, Respecting causality for training physics-informed neural networks, *Comput. Methods Appl. Mech. Eng.* (2024).

[6] A. D. Jagtap, E. Kharazmi, G. E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws, *Comput. Methods Appl. Mech. Eng.* 365 (2020) 113028.

[7] A. D. Jagtap, G. E. Karniadakis, Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, *Commun. Comput. Phys.* 28(5) (2020) 2002–2041.

[8] A. D. Jagtap, G. E. Karniadakis, Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, *AAAI Spring Symposium (MLPS)*, CEUR Workshop Proceedings 2964 (2021).

[9] A. D. Jagtap, K. Kawaguchi, G. E. Karniadakis, Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks, *Proc. R. Soc. A* 476 (2020) 20200334.

[10] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, *ICLR* (2015).

[11] D. C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, *Math. Program.* 45 (1989) 503–528.