

Article

Physics-informed neural networks for the heat equation with dynamic (Wentzell) boundary conditions

Muhammad Awais

Graduate School of Engineering, Oita University, 700 Dannoharu, Oita-shi, Oita 870-1192, Japan;
awaiseducation@gmail.com

Received: 08 Feb 2026; Revised: 05 Mar 2026; Accepted: 13 Mar 2026; Published: 07 May 2026

Abstract: Dynamic boundary conditions provide a stringent verification setting for physics-informed neural networks (PINNs), because the approximation must satisfy an interior parabolic equation together with an evolution law posed on the boundary. This paper develops a transparent benchmark study for the one-dimensional heat equation with Wentzell-type dynamic boundary conditions. Two manufactured solutions are employed. Benchmark A is a cosine solution for which homogeneous boundary forcing is admissible only under the compatibility condition $k = \alpha\pi^2$; it is intentionally retained as a reduced Wentzell test because the endpoint derivatives vanish and the boundary-gradient term is inactive. Benchmark B is a fully active test with nonzero endpoint derivatives at both boundaries, so that the βu_x contribution is explicitly exercised. The numerical evidence is based on directly computed norms, residuals, five independent random-feature seeds, an implemented finite-difference reference solver, and a targeted ablation study. The reported neural solver is a least-squares physics-informed random-feature network with hard initial-condition enforcement, 250 hidden features, 3980 interior collocation points, and 401 boundary times per endpoint. Over five seeds, the mean relative errors are $E_{L^2} = (8.65 \pm 1.27) \times 10^{-6}$ and $E_{L^\infty} = (1.12 \pm 0.20) \times 10^{-5}$ for Benchmark A, and $E_{L^2} = (9.45 \pm 2.42) \times 10^{-8}$ and $E_{L^\infty} = (2.77 \pm 0.99) \times 10^{-7}$ for Benchmark B. The finite-difference baseline displays approximately second-order convergence under the parabolic scaling $\Delta t = 2h^2$. These results show that the two-case manufactured suite provides a useful verification framework for smooth one-dimensional Wentzell-type dynamics, while the scope of the claims remains restricted to this benchmark setting rather than general neural-solver superiority.

Keywords: physics-informed neural networks, Wentzell boundary conditions, dynamic boundary conditions, heat equation, manufactured solution, finite differences, benchmark verification

MSC: 35K20, 65M22.

1. Introduction

The heat equation is a classical verification problem for numerical methods because it is analytically tractable while retaining the principal structure of parabolic diffusion. In many applications, however, the boundary is not a passive location at which only a value or flux is prescribed. Surface storage, kinetic exchange, membrane effects, and active interfacial dynamics may lead to boundary laws containing a time derivative of the unknown on the boundary. Such conditions are commonly described as Wentzell-type, generalized Wentzell, kinetic, or dynamic boundary conditions, depending on the analytical context and sign convention [1–3].

Physics-informed neural networks (PINNs) approximate solutions of differential equations by embedding the governing equations and auxiliary constraints into a residual-minimization objective [4–6]. Their appeal lies in the use of automatic differentiation or exact differentiated basis functions to evaluate differential operators at scattered collocation points, thereby allowing a mesh-free approximation strategy. PINNs and related physics-informed methods have been investigated for heat-transfer problems, diffusion models, inverse thermal problems, and phase-change models [7–11]. At the same time, the literature has identified several recurrent difficulties, including loss imbalance, optimization pathologies, seed sensitivity, and deceptively small residuals that may coexist with poor physical fidelity [12–14].

Dynamic boundary conditions are especially well suited for exposing these issues. A reliable solver must satisfy the interior heat equation and, simultaneously, endpoint evolution laws involving u_t , u , and a spatial derivative. This introduces several opportunities for benchmark misinterpretation. A manufactured solution may satisfy the dynamic boundary law while accidentally nullifying the boundary-gradient term. Likewise, a small interior residual does not imply that the dynamic boundary conditions are satisfied, and visual agreement in plots does not replace explicitly tabulated error and residual calculations.

This study addresses those concerns by constructing and documenting a small but fully specified benchmark suite. Two manufactured solutions are used: a reduced dynamic-boundary test that makes the compatibility condition for homogeneous forcing explicit, and a fully active Wentzell-gradient test that activates the βu_x contribution at both endpoints. The paper reports directly computed error and residual measures, an executed finite-difference baseline, five-seed variability, and an ablation study. The neural approximation adopted for the numerical tables is a least-squares physics-informed random-feature network, closely related to physics-informed extreme learning machines [15,16]. This choice is deliberate. Because the benchmark problem is linear, fixing the hidden features and solving a regularized linear least-squares system yields a reproducible residual-minimizing neural approximation while avoiding ambiguity associated with optimizer settings, local minima, and learning-rate schedules. Standard fully trainable PINNs can also be applied to the same benchmark, but the results reported here are based only on the directly computed least-squares random-feature formulation.

2. Model problem and sign convention

Let $Q = (0, 1) \times (0, T]$. The interior equation is

$$u_t(x, t) = \alpha u_{xx}(x, t), \quad (x, t) \in Q, \quad (1)$$

with initial condition

$$u(x, 0) = u_0(x), \quad x \in [0, 1]. \quad (2)$$

The dynamic boundary conditions are written as

$$u_t(0, t) + ku(0, t) + \beta u_x(0, t) = g_0(t), \quad (3)$$

$$u_t(1, t) + ku(1, t) - \beta u_x(1, t) = g_1(t), \quad (4)$$

for $t \in (0, T]$, where $\alpha > 0$, $k \geq 0$, and $\beta \in \mathbb{R}$. The outward normal derivative is $\partial_\nu u(0, t) = -u_x(0, t)$ and $\partial_\nu u(1, t) = u_x(1, t)$. Hence (3)–(4) may also be written compactly as

$$u_t + ku - \beta \partial_\nu u = g \quad \text{on } \partial(0, 1), \quad (5)$$

with endpoint-specific forcing values. The sign convention is stated explicitly because different formulations of Wentzell and dynamic boundary conditions appear in the literature.

All numerical experiments in this paper use

$$\alpha = 1, \quad \beta = 1, \quad k = \pi^2, \quad T = 0.5. \quad (6)$$

3. Manufactured benchmark suite

3.1. Benchmark A: reduced Wentzell test

Benchmark A is defined by

$$u_A^*(x, t) = e^{-\alpha \pi^2 t} \cos(\pi x). \quad (7)$$

It satisfies the interior equation because $(u_A^*)_{xx} = -\pi^2 u_A^*$. The initial condition is $u_0(x) = \cos(\pi x)$. Substitution into the boundary laws gives

$$g_{0,A}(t) = (k - \alpha\pi^2)e^{-\alpha\pi^2 t}, \quad (8)$$

$$g_{1,A}(t) = (\alpha\pi^2 - k)e^{-\alpha\pi^2 t}. \quad (9)$$

Thus homogeneous forcing, $g_{0,A} \equiv g_{1,A} \equiv 0$, is valid if and only if

$$k = \alpha\pi^2. \quad (10)$$

This compatibility condition is imposed in (6).

Benchmark A is intentionally retained even though it does not activate the boundary-gradient term. Since

$$\partial_x u_A^*(x, t) = -\pi e^{-\alpha\pi^2 t} \sin(\pi x), \quad (11)$$

we have $\partial_x u_A^*(0, t) = \partial_x u_A^*(1, t) = 0$. Consequently, the βu_x contribution in (3)–(4) is inactive. Benchmark A is therefore a reduced Wentzell test: it verifies the interior diffusion equation, the dynamic boundary time derivative, the boundary reaction term, and the compatibility condition for homogeneous forcing, but not the flux contribution itself.

3.2. Benchmark B: fully active Wentzell test

Benchmark B is constructed to test the boundary-gradient contribution directly. It is defined by

$$u_B^*(x, t) = e^{-\alpha\omega^2 t} \{\cos(\omega x) + \sin(\omega x)\}, \quad (12)$$

with $\omega > 0$. The interior equation is again satisfied because

$$\partial_{xx} \{\cos(\omega x) + \sin(\omega x)\} = -\omega^2 \{\cos(\omega x) + \sin(\omega x)\}. \quad (13)$$

The forcing terms required by (3)–(4) are

$$g_{0,B}(t) = (k - \alpha\omega^2 + \beta\omega)e^{-\alpha\omega^2 t}, \quad (14)$$

$$g_{1,B}(t) = e^{-\alpha\omega^2 t} \left[(k - \alpha\omega^2)(\cos \omega + \sin \omega) - \beta\omega(\cos \omega - \sin \omega) \right]. \quad (15)$$

The endpoint derivatives are

$$\partial_x u_B^*(0, t) = \omega e^{-\alpha\omega^2 t}, \quad (16)$$

$$\partial_x u_B^*(1, t) = \omega e^{-\alpha\omega^2 t} (\cos \omega - \sin \omega). \quad (17)$$

For the numerical study,

$$\omega = \frac{\pi}{2}. \quad (18)$$

Then both endpoint derivatives are nonzero:

$$\partial_x u_B^*(0, t) = \frac{\pi}{2} e^{-\pi^2 t/4}, \quad \partial_x u_B^*(1, t) = -\frac{\pi}{2} e^{-\pi^2 t/4}. \quad (19)$$

With the parameter choice (6), the two forcing terms coincide,

$$g_{0,B}(t) = g_{1,B}(t) = \left(\frac{3\pi^2}{4} + \frac{\pi}{2} \right) e^{-\pi^2 t/4}. \quad (20)$$

Table 1 summarizes the two manufactured benchmarks. Benchmark A is a reduced compatibility and implementation test, whereas Benchmark B is a fully active Wentzell-gradient test. Figure 1 visualizes the two benchmark solutions and their endpoint behavior. The lower-left panel highlights the flux-term degeneracy

of Benchmark A, while the lower-right panel shows that Benchmark B has nonzero endpoint derivatives and nonzero boundary forcing throughout the time interval.

Table 1. Summary of the two manufactured benchmarks. Benchmark A is a reduced compatibility and implementation test, whereas Benchmark B is the fully active Wentzell-gradient test

Quantity	Benchmark A	Benchmark B
Exact solution	$e^{-\pi^2 t} \cos(\pi x)$	$e^{-\pi^2 t/4} \{\cos(\pi x/2) + \sin(\pi x/2)\}$
Boundary forcing	$g_0 = g_1 = 0$ under $k = \pi^2$	$g_0 = g_1 = (3\pi^2/4 + \pi/2)e^{-\pi^2 t/4}$
Endpoint derivatives	$u_x(0, t) = u_x(1, t) = 0$	$u_x(0, t) = \pi e^{-\pi^2 t/4}/2, u_x(1, t) = -\pi e^{-\pi^2 t/4}/2$
Role of βu_x	Inactive	Active at both endpoints
Primary use	Compatibility and reduced dynamic-boundary verification	Fully active Wentzell-gradient verification

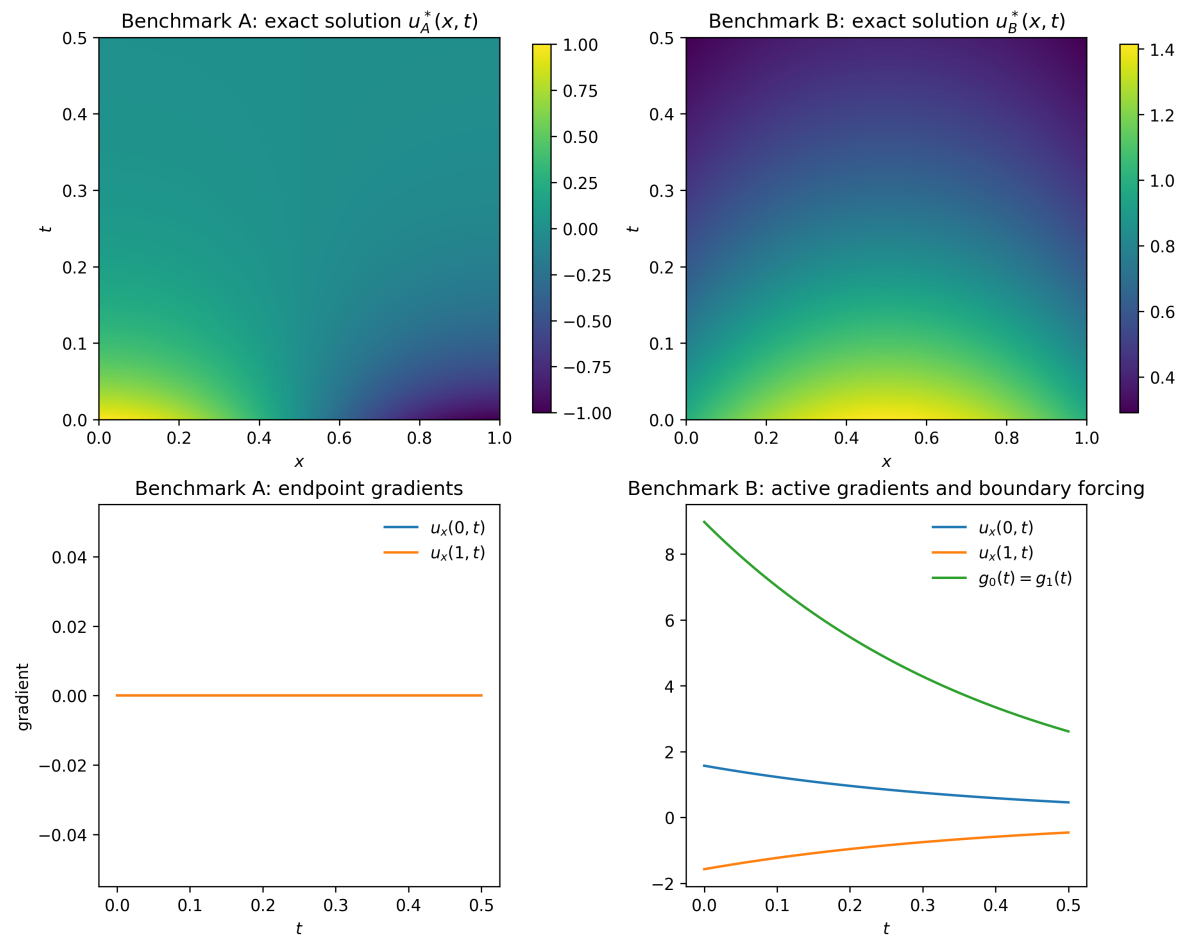


Figure 1. Manufactured benchmark suite. The upper panels show the exact solutions for Benchmarks A and B on $[0, 1] \times [0, 0.5]$. The lower-left panel confirms that Benchmark A has zero endpoint gradients. The lower-right panel shows the active endpoint gradients and the coincident boundary forcing $g_0(t) = g_1(t)$ for Benchmark B

4. Least-squares physics-informed neural approximation

4.1. Hard initial-condition ansatz

The neural approximation used for the reported tables is

$$u_\theta(x, t) = u_0(x) + t \sum_{m=1}^M c_m \phi_m(x, t), \tag{21}$$

where

$$\phi_m(x, t) = \tanh(a_m \hat{x} + b_m \hat{t} + d_m), \quad \hat{x} = 2x - 1, \quad \hat{t} = 2t/T - 1. \quad (22)$$

The factor t enforces $u_\theta(x, 0) = u_0(x)$ exactly. Consequently, the initial condition is not included as an active loss term in the hard-constrained runs. It is monitored separately through

$$D_{\text{IC}} = \left(\frac{1}{N_{\text{IC}}} \sum_i |u_\theta(x_i, 0) - u_0(x_i)|^2 \right)^{1/2}, \quad (23)$$

which is zero up to roundoff for (21).

The hidden-feature parameters (a_m, b_m, d_m) are sampled independently for each seed, with $a_m, b_m \sim \mathcal{N}(0, 1.5^2)$ and $d_m \sim \mathcal{U}[-1, 1]$. Only the output coefficients c_m are solved. This random-feature formulation is more restrictive than a fully trainable deep PINN, but it is appropriate for a linear benchmark study because it produces a reproducible residual-minimizing approximation without the confounding influence of optimizer trajectories.

4.2. Residual equations and least-squares system

The interior residual is

$$r_f(x, t) = \partial_t u_\theta(x, t) - \alpha \partial_{xx} u_\theta(x, t). \quad (24)$$

The endpoint residuals are

$$r_0(t) = \partial_t u_\theta(0, t) + k u_\theta(0, t) + \beta \partial_x u_\theta(0, t) - g_0(t), \quad (25)$$

$$r_1(t) = \partial_t u_\theta(1, t) + k u_\theta(1, t) - \beta \partial_x u_\theta(1, t) - g_1(t). \quad (26)$$

Because (21) is linear in the unknown vector $c = (c_1, \dots, c_M)^T$, collocation of r_f , r_0 , and r_1 leads to a linear least-squares problem

$$c = \arg \min_{z \in \mathbb{R}^M} \|Az - \mathbf{b}\|_2^2 + \lambda \|z\|_2^2, \quad (27)$$

where $\lambda = 10^{-8}$ in all reported runs. The ridge term regularizes the normal equations while remaining small relative to the residual scales.

For each random-feature seed, the collocation set contains 2000 pseudorandom interior points and a 44×45 tensor-product interior grid, giving $N_f = 3980$ interior collocation points. The boundary residuals are collocated at 401 equally spaced times per endpoint. Unless otherwise stated, $M = 250$ hidden features are used. All reported solution errors are computed on an independent 301×301 evaluation grid. PDE residual RMS values are computed on 12000 evaluation points sampled from that grid, and boundary residual RMS values are computed on 2001 equally spaced boundary times.

4.3. Differentiated feature formulas

For reproducibility, the differentiated basis terms are stated explicitly. Let

$$z_m = a_m \hat{x} + b_m \hat{t} + d_m, \quad s_m = 1 - \tanh^2(z_m). \quad (28)$$

For the hard feature $\psi_m(x, t) = t \tanh(z_m)$ appearing in (21),

$$\partial_t \psi_m = \tanh(z_m) + t s_m b_m \frac{2}{T}, \quad (29)$$

$$\partial_x \psi_m = 2t a_m s_m, \quad (30)$$

$$\partial_{xx} \psi_m = -8t a_m^2 s_m \tanh(z_m). \quad (31)$$

Eqs. (29)–(31) are sufficient to assemble the least-squares matrix without automatic differentiation.

5. Finite-difference reference solver

A deterministic classical solver is included to provide an independent baseline. Let $x_i = ih, i = 0, \dots, N$, with $h = 1/N$, and $t^n = n\Delta t$. For interior nodes $i = 1, \dots, N - 1$, backward Euler gives

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h^2}. \tag{32}$$

The dynamic boundary laws are discretized as

$$\frac{u_0^{n+1} - u_0^n}{\Delta t} + ku_0^{n+1} + \beta \frac{-3u_0^{n+1} + 4u_1^{n+1} - u_2^{n+1}}{2h} = g_0(t^{n+1}), \tag{33}$$

$$\frac{u_N^{n+1} - u_N^n}{\Delta t} + ku_N^{n+1} - \beta \frac{3u_N^{n+1} - 4u_{N-1}^{n+1} + u_{N-2}^{n+1}}{2h} = g_1(t^{n+1}). \tag{34}$$

The one-sided spatial derivatives in (33)–(34) are second-order accurate. The time discretization is first-order, so the overall method is not second-order for an arbitrary time step. In the convergence study below, the parabolic scaling $\Delta t = 2h^2$ is used, making the first-order temporal error $O(h^2)$ and allowing the spatial second-order behavior to be observed.

6. Error and residual metrics

For a computed approximation u_h , the reported relative errors are

$$E_{L^2} = \frac{\|u_h - u^*\|_{L^2([0,1] \times [0,T])}}{\|u^*\|_{L^2([0,1] \times [0,T])}}, \quad E_{L^\infty} = \frac{\|u_h - u^*\|_{L^\infty([0,1] \times [0,T])}}{\|u^*\|_{L^\infty([0,1] \times [0,T])}}. \tag{35}$$

The continuous norms are approximated by uniform-grid quadrature or evaluation on the stated grids. Residual magnitudes are reported as

$$R_f = \left(\frac{1}{N_q} \sum_{i=1}^{N_q} |r_f(x_i, t_i)|^2 \right)^{1/2}, \quad R_j = \left(\frac{1}{N_b} \sum_{i=1}^{N_b} |r_j(t_i)|^2 \right)^{1/2}, \quad j = 0, 1. \tag{36}$$

These metrics are reported explicitly to avoid the limitations of figure-only assessment.

7. Numerical results

7.1. Five-seed neural benchmark results

Table 2 reports five-seed statistics for Benchmarks A and B. Benchmark A is solved with homogeneous boundary forcing under the compatibility condition $k = \pi^2$. Benchmark B uses the fully active forcing (20). The statistics are computed over seeds $s = 0, 1, 2, 3, 4$.

Table 2. Least-squares random-feature PINN results over five independent feature seeds. Values are mean \pm sample standard deviation. Benchmark B is the fully active Wentzell-gradient case

Benchmark	E_{L^2}	E_{L^∞}	R_f	R_0	R_1
A	$(8.65 \pm 1.27) \times 10^{-6}$	$(1.12 \pm 0.20) \times 10^{-5}$	$(2.30 \pm 0.39) \times 10^{-4}$	$(1.94 \pm 0.84) \times 10^{-4}$	$(1.80 \pm 0.49) \times 10^{-4}$
B	$(9.45 \pm 2.42) \times 10^{-8}$	$(2.77 \pm 0.99) \times 10^{-7}$	$(9.87 \pm 2.77) \times 10^{-6}$	$(7.41 \pm 4.06) \times 10^{-6}$	$(8.15 \pm 2.00) \times 10^{-6}$

The principal numerical observation is that Benchmark B is not only specified analytically but also solved and evaluated quantitatively. Its relative errors remain below 10^{-6} , while the RMS boundary residuals remain below 10^{-5} . Because $u_x(0, t)$ and $u_x(1, t)$ are nonzero in Benchmark B, these residuals test the active βu_x contribution in (3)–(4). Benchmark A produces larger errors and residuals than Benchmark B in the present implementation, primarily because the faster decay $e^{-\pi^2 t}$ is more demanding for the fixed random-feature set than the milder decay $e^{-\pi^2 t/4}$ in Benchmark B. Nevertheless, both cases display small relative errors and stable seed-to-seed behavior.

Figure 2 shows representative exact time traces for Benchmark B at several spatial locations. The figure highlights the smooth temporal decay and the variation in amplitude across the interval, which together make Benchmark B a useful complementary test to the flux-inactive cosine benchmark.

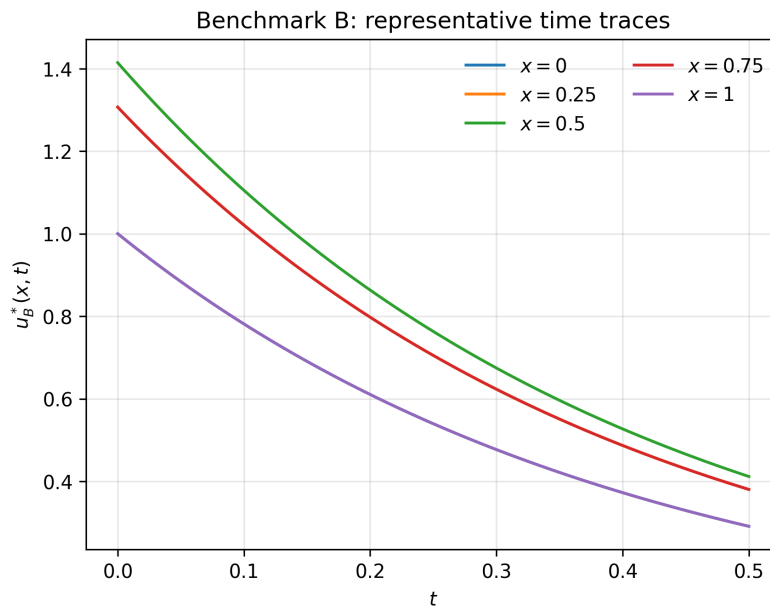


Figure 2. Benchmark B exact time traces at representative spatial locations. The slower decay relative to Benchmark A and the nontrivial spatial structure support a stable active-boundary verification case

The individual seed values are listed in Appendix A, so the means in Table 2 do not obscure seed-to-seed variability.

7.2. Finite-difference comparison

Table 3 reports the finite-difference baseline. The observed convergence rate is computed from successive E_{L^2} values under grid refinement. Both benchmarks show rates close to two, as expected when second-order spatial differences are combined with the parabolic scaling $\Delta t = 2h^2$.

Table 3. Finite-difference reference errors. The time step is chosen as $\Delta t = 2h^2$ and then adjusted only to hit $T = 0.5$ exactly

Benchmark	N	Δt	E_{L^2}	E_{L^∞}	observed order in E_{L^2}
A	40	1.2500×10^{-3}	4.3439×10^{-3}	2.2374×10^{-3}	–
	80	3.1250×10^{-4}	1.0911×10^{-3}	5.5944×10^{-4}	1.99
	160	7.8125×10^{-5}	2.7294×10^{-4}	1.3974×10^{-4}	2.00
B	40	1.2500×10^{-3}	3.8424×10^{-4}	3.6238×10^{-4}	–
	80	3.1250×10^{-4}	9.6462×10^{-5}	9.0692×10^{-5}	1.99
	160	7.8125×10^{-5}	2.4151×10^{-5}	2.2672×10^{-5}	2.00

The finite-difference results serve two purposes. First, they confirm that the manufactured solutions and forcing expressions are consistent with an independent discretization. Second, they show that the benchmark is not intrinsically difficult for a classical one-dimensional solver. The neural results should therefore be interpreted as verification of a physics-informed approximation strategy, not as evidence that the neural method is preferable to finite differences for this linear benchmark. Finite-difference convergence under the scaling $\Delta t = 2h^2$ is shown in Figure 3. Both manufactured benchmarks exhibit approximately second-order decay of the relative L^2 error with respect to h .

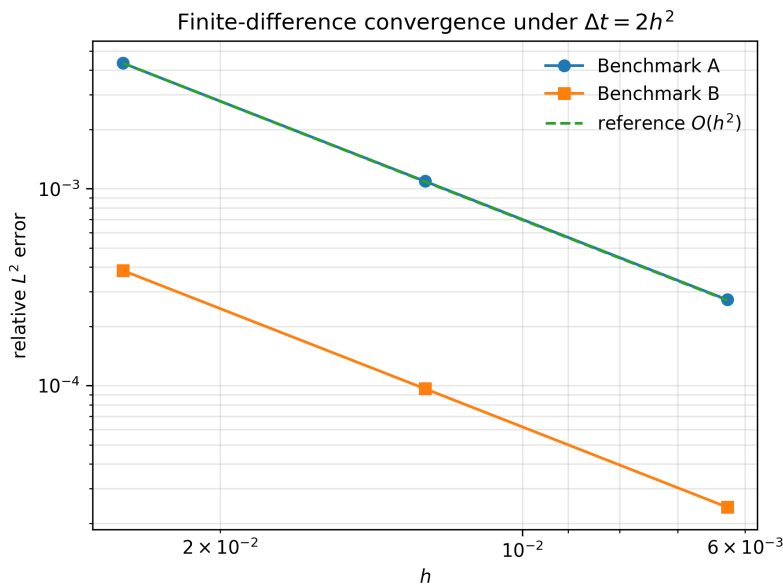


Figure 3. Finite-difference convergence under the scaling $\Delta t = 2h^2$. Both manufactured benchmarks exhibit approximately second-order decay of the relative L^2 error with respect to h

7.3. Ablation study on the fully active benchmark

Table 4 reports a targeted ablation study on Benchmark B using seed 0. The most informative comparison is the removal of the boundary residual. When the least-squares system includes only the interior PDE residual while retaining the hard initial condition, the PDE residual remains small but the boundary residuals become $O(1)$ and the relative L^2 error increases to 6.28×10^{-2} . This directly demonstrates why dynamic boundary residuals must be reported separately. A method can satisfy the interior equation quite well and still fail the Wentzell-type boundary law.

Table 4. Ablation study for Benchmark B, seed 0. The full setting uses $M = 250$, scaled inputs, hard initial-condition enforcement, and both endpoint dynamic-boundary residuals

Variant	E_{L^2}	E_{L^∞}	R_f	R_0	R_1
Full setting	5.61×10^{-8}	1.41×10^{-7}	6.49×10^{-6}	4.46×10^{-6}	4.76×10^{-6}
No boundary residual	6.28×10^{-2}	1.75×10^{-1}	4.50×10^{-6}	4.16×10^{-1}	1.37
Raw inputs, no scaling	9.96×10^{-8}	1.35×10^{-7}	4.53×10^{-6}	4.00×10^{-6}	4.29×10^{-6}
Soft initial condition	1.67×10^{-6}	5.59×10^{-6}	7.61×10^{-6}	2.83×10^{-6}	2.74×10^{-6}
$M = 100$ features	7.12×10^{-5}	1.29×10^{-4}	3.09×10^{-3}	2.92×10^{-3}	3.23×10^{-3}
$M = 50$ features	2.48×10^{-3}	4.86×10^{-3}	4.31×10^{-2}	4.51×10^{-2}	6.11×10^{-2}

The scaling ablation is also informative. On this simple one-dimensional domain, replacing the scaled inputs with raw (x, t) coordinates does not degrade the result and is slightly better for seed 0. Input scaling is therefore not claimed to be essential for this particular problem. The soft-initial-condition variant also performs well, with relative errors on the order of 10^{-6} , but its initial mismatch is no longer identically zero because the initial condition is enforced as a least-squares penalty rather than imposed exactly. Reducing the feature count has a clear effect: $M = 100$ remains accurate, whereas $M = 50$ is visibly under-resolved in both residual and solution-error metrics.

8. Discussion

The numerical study supports four main conclusions. First, the benchmark suite benefits from a clear separation between a reduced dynamic-boundary test and a fully active Wentzell-gradient test. Benchmark A remains useful because it exposes the compatibility condition for homogeneous forcing and verifies the dynamic boundary time derivative and reaction terms. However, its endpoint derivatives vanish, so it cannot

probe the boundary-gradient contribution. Benchmark B removes that degeneracy and therefore provides the more decisive verification case for the Wentzell-type term.

Second, the evidence is quantitative rather than purely visual. The principal claims are supported by relative L^2 and L^∞ errors, RMS PDE residuals, and RMS boundary residuals, all computed on independent evaluation sets. The appendix gives per-seed values, so the reported mean values are traceable.

Third, the finite-difference baseline clarifies the nature of the contribution. A standard implicit finite-difference method solves the one-dimensional linear problem accurately and with predictable convergence. The contribution of the present paper is therefore not algorithmic dominance over classical solvers; rather, it is the provision of a transparent verification problem and a documented neural approximation workflow for dynamic boundary conditions.

Fourth, the ablation study shows why boundary-specific reporting is indispensable. The no-boundary-residual variant achieves a small interior PDE residual while simultaneously producing large boundary residuals and a much larger solution error. For dynamic boundary conditions, endpoint residuals should be reported individually rather than merged into a single opaque loss number.

Several limitations should also be stated clearly. The problem is linear, one-dimensional, smooth, and manufactured. The random-feature PINN used here is intentionally simpler and more reproducible than a fully trainable deep network; accordingly, the results do not imply that Adam-LBFGS PINNs, adaptive loss-balancing PINNs, or residual-adaptive PINNs will behave identically on the same benchmark. The finite-difference study is intended as a consistency check rather than a full computational-cost comparison. No claim is made about nonlinear Wentzell problems, higher-dimensional domains, irregular geometries, noisy data, or inverse parameter estimation.

9. Conclusion

This paper has presented a reproducible benchmark study of a physics-informed neural approximation for the heat equation with Wentzell-type dynamic boundary conditions. Two manufactured solutions were used: Benchmark A as a reduced compatibility test and Benchmark B as a fully active Wentzell-gradient test. The reported least-squares random-feature PINN solved both cases with small directly measured errors and residuals, while the finite-difference solver independently confirmed the correctness of the manufactured solutions and forcing terms.

The main contribution is a verification-oriented benchmark framework rather than a claim of general neural-solver superiority. Within the smooth one-dimensional setting studied here, the benchmark suite provides a transparent test bed for evaluating whether a neural method satisfies both the interior diffusion equation and the dynamic endpoint laws. This makes the benchmark suitable as a compact reference problem for future studies of PINNs and related methods under dynamic or Wentzell-type boundary conditions.

Conflicts of Interest: The author declares no conflict of interest

References

- [1] Goldstein, G. R. (2006). Derivation and physical interpretation of general boundary conditions. *Advances in Differential Equations*, 11(4), 457–480.
- [2] Chorfi, S.-E., El Guermai, G., Maniar, L., & Oukdach, W. (2023). Boundary null controllability for the heat equation with dynamic boundary conditions. *Evolution Equations and Control Theory*, 12(1), 83–102.
- [3] Chorfi, S.-E., Ismailov, M. I., Maniar, L., & Öner, I. (2026). Boundary null controllability of the heat equation with Wentzell boundary condition and Dirichlet control. *Evolution Equations and Control Theory*, 15(1), 157–176.
- [4] Lagaris, I. E., Likas, A., & Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5), 987–1000.
- [5] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). 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.
- [6] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422–440.
- [7] Kadeethum, T., Jørgensen, T. M., & Nick, H. M. (2020). Physics-informed neural networks for solving nonlinear diffusivity and Biot's equations. *PLOS ONE*, 15(5), e0232683.

- [8] Cai, S., Wang, Z., Wang, S., Perdikaris, P., & Karniadakis, G. E. (2021). Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6), 060801.
- [9] Bowman, B., Qian, C., Kurz, J., Khan, T., Gil, E., & Gamez, N. (2023). Physics-informed neural networks for the heat equation with source term under various boundary conditions. *Algorithms*, 16(9), 428.
- [10] Li, J., Wu, W., & Feng, X. (2023). Improved physics-informed neural networks combined with small sample learning to solve two-dimensional Stefan problem. *Entropy*, 25(4), 675.
- [11] Madir, B.-E., Luddens, F., Lothodé, C., & Danaila, I. (2025). Physics-informed neural networks for heat conduction with phase change. *International Journal of Heat and Mass Transfer*, 252, 127430.
- [12] Wang, S., Teng, Y., & Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5), A3055–A3081.
- [13] Krishnapriyan, A. S., Gholami, A., Zhe, S., Kirby, R. M., & Mahoney, M. W. (2021). Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34, 26548–26560.
- [14] Martínez-Esteban, A., Calvo-Barlés, P., Martín-Moreno, L., & Rodrigo, S. G. (2025). Physics-informed neural networks with dynamical boundary constraints. *arXiv preprint arXiv:2507.21800*.
- [15] Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3), 489–501.
- [16] Dwivedi, V., & Srinivasan, B. (2020). Physics informed extreme learning machine (PIELM): A rapid method for the numerical solution of partial differential equations. *Neurocomputing*, 391, 96–118.

Appendix A. Per-seed neural results

Table 5 gives the individual seed values underlying Table 2.

Table 5. Per-seed least-squares random-feature PINN results

Benchmark	Seed	E_{L^2}	E_{L^∞}	R_f	R_0	R_1
A	0	7.8779×10^{-6}	9.1372×10^{-6}	1.7664×10^{-4}	1.8964×10^{-4}	1.6546×10^{-4}
A	1	9.1684×10^{-6}	1.3170×10^{-5}	2.7907×10^{-4}	2.0399×10^{-4}	1.8062×10^{-4}
A	2	8.4795×10^{-6}	1.2272×10^{-5}	2.0810×10^{-4}	6.3552×10^{-5}	1.6208×10^{-4}
A	3	7.2177×10^{-6}	8.9865×10^{-6}	2.4523×10^{-4}	2.9908×10^{-4}	1.3225×10^{-4}
A	4	1.0520×10^{-5}	1.2235×10^{-5}	2.4242×10^{-4}	2.1243×10^{-4}	2.6127×10^{-4}
B	0	5.6107×10^{-8}	1.4103×10^{-7}	6.4871×10^{-6}	4.4643×10^{-6}	4.7574×10^{-6}
B	1	9.8526×10^{-8}	2.3459×10^{-7}	8.9956×10^{-6}	7.8165×10^{-6}	9.9158×10^{-6}
B	2	8.9069×10^{-8}	2.6842×10^{-7}	8.4953×10^{-6}	4.3671×10^{-6}	8.1352×10^{-6}
B	3	1.1844×10^{-7}	3.9317×10^{-7}	1.2041×10^{-5}	1.4208×10^{-5}	9.0877×10^{-6}
B	4	1.1037×10^{-7}	3.4885×10^{-7}	1.3329×10^{-5}	6.1750×10^{-6}	8.8428×10^{-6}

Appendix B. Normal-equation assembly for the hard random-feature PINN

For each interior point (x_i, t_i) , the corresponding row of the least-squares matrix is

$$A_{im}^{(f)} = \partial_t \psi_m(x_i, t_i) - \alpha \partial_{xx} \psi_m(x_i, t_i), \quad (37)$$

and the right-hand side is

$$b_i^{(f)} = \alpha u_0''(x_i), \quad (38)$$

because the base function $u_0(x)$ contributes $-\alpha u_0''(x)$ to the PDE residual. For the left endpoint,

$$A_{im}^{(0)} = \partial_t \psi_m(0, t_i) + k \psi_m(0, t_i) + \beta \partial_x \psi_m(0, t_i), \quad (39)$$

with

$$b_i^{(0)} = g_0(t_i) - k u_0(0) - \beta u_0'(0). \quad (40)$$

For the right endpoint,

$$A_{im}^{(1)} = \partial_t \psi_m(1, t_i) + k \psi_m(1, t_i) - \beta \partial_x \psi_m(1, t_i), \quad (41)$$

with

$$b_i^{(1)} = g_1(t_i) - ku_0(1) + \beta u_0'(1). \quad (42)$$

Stacking these rows yields (27). This formulation is included so that the reported tables can be regenerated without relying on automatic differentiation.

Appendix C. Reproducibility and implementation details

The reported neural calculations use double-precision NumPy linear algebra. For each seed $s \in \{0, 1, 2, 3, 4\}$, the pseudorandom generator is initialized with that seed before drawing the random-feature weights, random-feature biases, and pseudorandom interior collocation points. The deterministic tensor-product interior points, boundary times, and evaluation grids are identical for all seeds. The finite-difference matrix is constant in time and is factorized once for each grid size. No external dataset is used.

The numerical values in the tables were generated using the following fixed settings: $M = 250$, random-feature scale 1.5, ridge parameter 10^{-8} , $N_f = 3980$ interior collocation points, 401 boundary times per endpoint, a 301×301 solution evaluation grid, 12000 sampled PDE-residual evaluation points, and 2001 boundary-residual evaluation times. For the finite-difference study, $N \in \{40, 80, 160\}$ and $\Delta t = 2h^2$ adjusted to reach $T = 0.5$ exactly.

The source package accompanying this manuscript contains the cleaned Python implementation used to assemble the least-squares system, compute the finite-difference baseline, evaluate the reported metrics, and generate the figures. This organization is intended to make independent verification straightforward.



© 2026 by the authors; licensee PSRP, Lahore, Pakistan. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).