

DEEP OPERATOR NETWORKS FOR THERMAL ENERGY STORAGE: LEARNING TEMPERATURE AND LIQUID-FRACTION DYNAMICS IN PHASE CHANGE MATERIALS

Shahbaz Ahmad^{*1}, Aftab Alam²

^{*1,2}Department of Mathematics, Islamia College University Peshawar, Khyber Pakhtunkhwa, Pakistan

^{*1}shahbazktk00@gmail.com

DOI: <https://doi.org/10.5281/zenodo.23012766>

Keywords

Deep Operator Networks, Phase Change Materials, Thermal Energy Storage, Operator Learning, Physics-Informed Machine Learning, Heat Transfer

Article History

Received: 21 June 2026

Accepted: 12 September 2026

Published: 28 September 2026

Copyright @Author

Corresponding Author: *

Shahbaz Ahmad

Abstract

Predicting transient thermal transport and phase transition in Phase Change Materials (PCMs) subjected to varying boundary heating conditions typically requires repeatedly evaluating heat conduction models coupled with an effective heat capacity framework that accounts for latent heat absorption and release. To bypass this computational burden, we implement Deep Operator Networks (DeepONets) to learn the functional mapping from arbitrary wall heat-flux inputs to interior temperature distributions $T(x, z, t)$ within a PCM domain. The local liquid-fraction field $\phi(x, z, t)$ is directly recovered during post-processing via the analytical solidus–liquidus temperature dependency. Furthermore, we outline a Physics-Informed DeepONet (PI-DeepONet) conceptual framework incorporating momentum, mass, and energy residuals for the complete Navier–Stokes system with enthalpy-porosity and Carman–Kozeny drag terms; this conceptual extension serves theoretical completeness and is not numerically evaluated in this manuscript. We establish an error bound associated with input sensor interpolation and evaluate its algorithmic impact. Numerical experiments covering 1D thermal diffusion, nonlinear dynamical systems, and 2D PCM melting indicate that an unstacked, normalized DeepONet reaches a relative L^2 temperature field error of 3.2×10^{-4} averaged over time trajectories (10 saved time steps) and 3.5×10^{-4} at the final temporal snapshot, while recovering the liquid-fraction interface with a relative L^2 deviation of 7.2%. Independent execution benchmarks yield an average FDM solve time of 6.90 ± 0.02 s compared to an average DeepONet forward-pass execution time of 8.18 ± 0.85 ms across 10,000 evaluation points. This corresponds to an approximate $844\times$ speedup relative to our non-vectorized Python finite-difference benchmark (vectorizing the FDM code lowers this acceleration ratio to $\sim 100\times$). Finally, key physical and methodological constraints are analyzed – including pure conduction assumptions, single-material/geometry evaluations, the absence of an empirical PI-DeepONet baseline, preliminary convergence metrics, single random-seed evaluations, and a lack of alternative operator baselines like FNOs.

1. Introduction

The classical universal approximation property indicates that standard feedforward

architectures can fit continuous functions on compact domains to arbitrary precision [1, 2]. An extension established by Chen and Chen

[3, 4] demonstrates that single-hidden-layer networks are capable of accurately approximating continuous nonlinear operators – mappings between infinite-dimensional function spaces. This theorem provides the mathematical foundation for learning operators directly from continuous fields, a concept realized practically through Deep Operator Networks (DeepONets) [5]. DeepONets generalize effectively by using a dual-network setup that separates input function encoding (handled by a branch network) from coordinate domain evaluation (handled by a trunk network).

In computational heat transfer and Phase Change Material (PCM) thermal energy storage (TES) modeling [6], conventional numerical techniques solve transient energy equations incorporating nonlinear effective heat capacities to account for phase change. Let G denote an operator mapping a continuous boundary profile u (such as an applied heat flux) to a corresponding output field $G(u)$. For any spatio-temporal coordinate point $y = (x, z, t)$, the network output $G(u)(y)$ estimates the localized thermal state $T(x, z, t)$, from which the phase distribution $\phi(x, z, t)$ is determined via constitutive relations. The boundary condition input u is discretely evaluated at m fixed sensor points $\{x_1, x_2, \dots, x_m\}$ (Fig. 1).

The effective heat capacity formulation incorporates phase change latent heat directly into a temperature-dependent specific heat within the mushy-zone interval $[T_s, T_l]$, avoiding explicit interface-tracking algorithms. However, re-running conventional solvers for each new boundary condition or design parameter remains computationally intensive. A trained DeepONet evaluates these parametric mappings rapidly during forward passes, bypassing full time-stepping procedures (detailed timing measurements are presented in Section 4.3).

While Fourier Neural Operators (FNOs) [9] process PDE mapping functions effectively over uniform spatial grids, DeepONet's branch-trunk design is well suited for boundary-value problems where inputs are supplied along domain boundaries and outputs are requested at non-uniform or arbitrary spatio-temporal locations. The theoretical landscape of operator

learning continues to expand across different structural paradigms [10], leading to architecture variations designed for multiple functional inputs [11], complex geometries [12], and convolutional operator networks suitable for irregular domains [13].

Machine learning applications in PCM and thermal storage systems. Data-driven modeling strategies for latent thermal storage are developing rapidly. Physics-informed architectures have been used to evaluate PCM-integrated energy exchanger performance, predicting thermal and humidity fields directly from residual constraints [17]. Similarly, convolutional networks have been deployed to infer internal melting-front temperatures from experimental imagery [18]. Recent developments include adaptive deep learning frameworks applied to multi-phase Stefan problems in finned enclosures [19], and energy-consistent operator surrogates (EC-DeepONet) utilized to accelerate material selection in geothermal systems modeled in COMSOL [20]. Although these applications illustrate the utility of operator learning, existing literature has not explicitly formulated a DeepONet mapping for effective heat capacity PCM systems alongside an explicit sensor-interpolation error bound. This work addresses that theoretical and applied gap. Related physics-informed frameworks and adaptive training strategies are reviewed comprehensively in recent domain surveys [16].

This manuscript is structured as follows. Section 2 outlines the mathematical formulation of DeepONets applied to PCM transport processes, including governing conduction models and the proposed PI-DeepONet formulation. Section 3 details the sensor interpolation theory alongside mathematical proofs. Section 4 discusses benchmark cases, performance evaluations, and convergence behaviors. Section 5 provides a detailed examination of current model limitations, and Section 6 summarizes key conclusions.

Theorem 1 (Universal Operator Approximation Theorem). *Let σ be a continuous non-polynomial activation function, let X represent a*

Banach Space, and let $K_1 \subset \mathcal{X}$, $K_2 \subset \mathbb{R}^d$ be compact domains. Let $V \subset C(K_1)$ be a compact subset, and let $G : V \rightarrow C(K_2)$ represent a continuous nonlinear operator. For any arbitrary error tolerance $\varepsilon > 0$, there exist integers n, p, m , real coefficients $c_i^k, \xi_{ij}^k, \theta_i^k, \zeta^k \in \mathbb{R}$, domain vectors $w_k \in$

$\mathbb{R}^d$, and discrete sensor nodes $x_j \in K_1$ ($i = 1, \dots, n, k = 1, \dots, p, j = 1, \dots, m$) such that

$$|G(u)(y) - \sum_{k=1}^p [\sum_{i=1}^n c_i^k \sigma(\sum_{j=1}^m \xi_{ij}^k u(x_j) + \theta_i^k)] \sigma(w_k \cdot y + \zeta_k)| < \varepsilon \quad (1)$$

is satisfied for all $u \in V$ and $y \in K_2$.

(In Eq. (1), the bracketed term is the **branch network** output and $\sigma(w_k \cdot y + \zeta_k)$ is the **trunk network** output.)

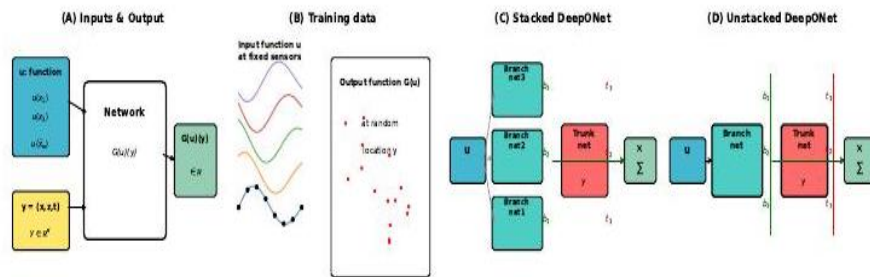


Figure 1: DeepONet operator learning schematic for PCM thermal dynamics.

(A) Network pipeline mapping boundary parameters $[u(x_1), u(x_2), \dots, u(x_m)]$ and coordinates $y = (x, z, t)$ to local temperatures $T(x, z, t)$, with liquid fraction $\varphi(x, z, t)$ recovered post-hoc. (B) Sensor location strategy and query point discretization. (C) Standard stacked DeepONet using p independent branch outputs. (D) Unstacked DeepONet utilizing a shared branch network output vector $[b_1, \dots, b_p]^T$.

2. Methodology

2.1. DeepONet Network Architecture

The DeepONet architecture processes input spaces through two functional components. A **trunk network** accepts domain query coordinates $y = (x, z, t)^T \in \mathbb{R}^d$ to produce basis vectors $[t_1, t_2, \dots, t_p]^T \in \mathbb{R}^p$. Concurrently, a **branch network** processes discrete sensor values $[u(x_1), u(x_2), \dots, u(x_m)]^T$ into embedding vectors $[b_1, b_2, \dots, b_p]^T \in \mathbb{R}^p$. Combining these sub-networks yields the target scalar output:

$$G(u)(y) \approx \sum_{k=1}^p b_k t_k + b_0 \quad (2)$$

We distinguish between stacked architectures (which build p distinct branch sub-networks outputting individual scalars b_k) and unstacked architectures (which construct a single branch network yielding the vector $[b_1, \dots, b_p]^T$). Unstacked variants contain significantly fewer

trainable parameters and generally exhibit superior generalization performance [5]; hence, an unstacked structure is employed across all primary benchmarks. Hidden layers in the trunk network utilize tanh activations ($t_k = \tanh(\cdot)$), while the branch network uses standard ReLU activations. Including a global scalar bias b_0 improves overall optimization stability.

Variable Normalization. The simulation time window spans $t_{\max} = 200$ s. For the 2D benchmark, features are scaled prior to training: input sensor vectors u are standardized to zero mean and unit variance, coordinate parameters $y = (x, z, t)$ are bounded within $[0, 1]$ via scaling ($x/L, z/H, t/t_{\max}$), and output thermal targets T are normalized using $(T - \mu_T)/\sigma_T$, where μ_T and σ_T denote training set distribution statistics. Proper normalization prevents numerical smearing along phase change boundaries, where phase transitions occur within a tight temperature interval ($T_s = 315$ K to $T_l = 325$ K).

2.2. Governing Transport Equations (Conduction with Effective Heat Capacity)

We consider a 2D square PCM cavity ($L = H = 50$ mm) subject to an arbitrary wall heat flux

profile $u(z)$ along the left boundary, while maintaining the right boundary at a constant temperature T_c . The top and bottom walls are modeled as adiabatic boundaries. Thermal diffusion and latent heat transition are governed by the transient heat equation incorporating an effective heat capacity formulation:

$$\rho C_p^e(T) \frac{\partial T}{\partial t} = k (\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2}) \quad (3)$$

where $C_p^e(T)$ is expressed as:

$$\begin{aligned} C_p^e(T) &= C_p, \text{ for } T < T_s \text{ (solid state)} \\ C_p^e(T) &= C_p + L_f / (T_l - T_s), \text{ for } T_s \leq T \leq T_l \text{ (mushy region)} \\ C_p^e(T) &= C_p, \text{ for } T > T_l \text{ (liquid state)} \end{aligned} \quad (4)$$

and the corresponding liquid fraction $\phi(T)$ follows a piecewise linear profile:

$$\begin{aligned} \phi(T) &= 0, \text{ for } T < T_s \text{ (solid state)} \\ \phi(T) &= (T - T_s) / (T_l - T_s), \text{ for } T_s \leq T \leq T_l \text{ (mushy region)} \\ \phi(T) &= 1, \text{ for } T > T_l \text{ (liquid state)} \end{aligned} \quad (5)$$

Here, ρ , k , C_p , L_f , T_s , and T_l represent material density, thermal conductivity, baseline specific heat, latent heat of fusion, solidus temperature, and liquidus temperature, respectively.

Boundary Conditions:

$$\begin{aligned} \text{Left wall (} x = 0 \text{): } & -k \frac{\partial T}{\partial x} \big|_{x=0} = u(z) \\ \text{Right wall (} x = L \text{): } & T \big|_{x=L} = T_c \\ \text{Top/Bottom (} z = 0, H \text{): } & \frac{\partial T}{\partial z} \big|_{z=0,H} = 0 \end{aligned} \quad (6)$$

Initial State:

$$T(x, z, 0) = T_c \text{ for all } (x, z) \in \Omega \quad (7)$$

Dimensionless Parameters. Based on properties in Table 3, the baseline thermal diffusivity is $\alpha = k / (\rho C_p) \approx 1.30 \times 10^{-7} \text{ m}^2/\text{s}$, the Stefan number is $Ste = C_p(T_l - T_c) / L_f = 0.25$, and the cavity aspect ratio is $AR = H/L = 1$. Because fluid motion is omitted in this baseline solver, listed Rayleigh and Prandtl numbers (Table 3) serve solely as reference descriptors for convective extensions.

Convective Transport Considerations. The formulation above models heat transfer dominated by thermal conduction. In physical PCM enclosures, buoyancy-driven natural convection within the liquid melt phase can alter melt-front development. Comprehensive

convection modeling requires coupling energy equations to the Navier-Stokes equations alongside Boussinesq approximations and Darcy-Carman-Kozeny drag terms [7, 8]. While discussed conceptually in Section 2.3, convective momentum transport is omitted from our reference numerical solver. Reported quantitative results evaluate pure conduction behavior defined by Eqs. (3)–(7).

2.3. Proposed Physics-Informed Extension

This data-driven formulation can be extended by integrating physics-informed residual losses derived from the coupled Navier-Stokes/enthalpy-porosity equations [14, 15]. Incorporating momentum balance, fluid continuity, Boussinesq buoyancy, and Carman-Kozeny porous damping requires addressing stiffness near liquid-solid interfaces, as highlighted in PINN training literature [16]. Numerical validation of this physics-informed framework remains an objective for subsequent studies.

2.4. Output Processing Strategy

The primary DeepONet architecture directly predicts the spatial temperature field $T(x, z, t)$. Local liquid fraction distributions $\phi(x, z, t)$ are recovered as a post-processing step via Eq. (5), ensuring thermodynamic consistency at phase boundaries.

This model can be expanded into multi-output configurations where separate prediction channels (T , u_x , u_z , p) share branch sensor representations while using dedicated trunk sub-networks. This expansion would be necessary for full Navier-Stokes convection systems but is unneeded for the conduction models evaluated here.

2.5. Boundary Function Dataset Generation

Spatial boundary functions $u(z)$ corresponding to non-uniform heat fluxes are constructed using Gaussian Random Fields (GRF) with Radial Basis Function (RBF) covariance kernels:

$$u \sim G(0, k_l(z_1, z_2)), \quad k_l(z_1, z_2) = \exp(-\|z_1 - z_2\|^2 / (2l^2)) \quad (8)$$

where characteristic length scale l determines sample profile smoothness. Alternatively,

Chebyshev polynomial spaces can be constructed via:

$$V_{\text{poly}} = \{ \sum_{i=0}^{N-1} a_i T_i(z) : |a_i| \leq M \} \quad (9)$$

where T_i denotes the i -th order Chebyshev polynomial and coefficients a_i are sampled uniformly from $[-M, M]$. Baseline datasets are computed using a second-order explicit finite-difference solver evaluated on an 80×80 spatial grid.

2.6. Training Workflow

Algorithm 1: DeepONet Training for PCM Thermal Dynamics

Require: Branch sub-network $B\theta$ (ReLU), trunk sub-network $T\theta$ (tanh), dataset $\{(\hat{u}_i, \hat{y}_j, T_{ref,ij})\}$

1. Apply feature scaling: $\hat{u}_i = (u_i - \mu u)/\sigma u$, $\hat{y} = (x/L, z/H, t/t_{\max})$, $T = (T - \mu T)/\sigma T$
2. Compute branch embedding vector: $b_i = B\theta([\hat{u}_i(x_1), \dots, \hat{u}_i(x_m)])$
3. Compute trunk feature representation: $t_j = T\theta(\hat{y}_j)$
4. Evaluate output prediction: $T_{ij} = b_i \cdot t_j + b_0$
5. Calculate empirical loss: $L = (1/N) \sum_{i,j} |T_{ij} - T_{ref,ij}|^2$
6. Update network weights θ via Adam optimizer (mini-batch size 32) using exponential learning rate decay (decay factor 0.5 every 5000 iterations)
7. Iterate until loss convergence criteria are met

3. Sensor Interpolation Bounds in Dynamical Systems

Consider a thermal dynamical system governed by state-space ODE representations:

$$d/dt s(t) = f(s(t), u(t), t), s(0) = s_0 \quad (10)$$

where input function $u \in V$ represents spatial boundary conditions and vector $s(t) \in \mathbb{R}^K$ encapsulates spatial thermal states. The system operator G maps boundary functions u directly to trajectories s :

$$(Gu)(t) = s_0 + \int_0^t f((Gu)(\tau), u(\tau), \tau) d\tau \quad (11)$$

Selecting $m+1$ uniform node points $x_j = a + j(b-a)/m$ for $j = 0, 1, \dots, m$ along $[a, b]$, we formulate the piecewise-linear interpolant:

$$u_m(x) = u(x_j) + (u(x_{j+1}) - u(x_j))/(x_{j+1} - x_j), \\ x_j \leq x \leq x_{j+1} \quad (12)$$

Let L_m denote the interpolation operator, and let $U_m = \{L_m(u) : u \in V\}$. For function $u \in V$ and corresponding interpolant $u_m \in U_m$, there exists a constant $\kappa(m, V)$ dependent on sample density m and space V such that:

$$\max_{x \in [a, b]} |u(x) - u_m(x)| \leq \kappa(m, V), \\ \text{where } \kappa(m, V) \rightarrow 0 \text{ as } m \rightarrow \infty \quad (13)$$

Lemma 2. When function space V represents a Gaussian Random Field generated with RBF kernel k , the interpolation bound scales as:

$$\kappa(m, V) \sim 1/(m^2 l^2) \quad (14)$$

Proof. Refer to Appendix C.

Assuming system vector field f satisfies uniform Lipschitz continuity with constant $c > 0$:

$$\|f(s_1, u, x) - f(s_2, u, x)\|_2 \leq c \|s_1 - s_2\|_2, \\ |f(s, u_1, x) - f(s, u_2, x)| \leq c |u_1 - u_2| \quad (15)$$

Theorem 3 (Sensor Density Bounds for Operator Approximations).

Let m be an integer selected such that $c(b-a)\kappa(m, V)e^{c(b-a)} < \epsilon$. For any evaluation coordinate $d \in [0, T]$, there exist matrix weights $W_1 \in \mathbb{R}^{n \times m+1}$, $W_2 \in \mathbb{R}^{K \times m}$ and bias vectors $b_1 \in \mathbb{R}^n$, $b_2 \in \mathbb{R}^K$ such that:

$$\|(Gu)(d) - [W_2 \cdot \sigma(W_1 \cdot [u(x_0), \dots, u(x_m)])^T + b_1] + b_2\|_2 < \epsilon$$

holds uniformly across all continuous boundary functions $u \in V$.

Proof. Refer to Appendix D.

Remark 1. In pure conduction systems, Lipschitz constant c is determined by thermal diffusivity and latent capacity scales. If natural convection were incorporated, Rayleigh numbers $Ra = \rho g \beta (T_h - T_c) L^3 / (\mu \alpha)$ would dictate transport scales; higher Ra values introduce steeper gradients, increasing c and requiring higher sensor densities m to achieve comparable error bounds.

4. Numerical Results

Optimization is performed using Adam with an initial learning rate of 0.001 combined with step-wise decay. Hyperparameter details are provided in Tables 1 and 2, and material properties are listed in Table 3.

Table 1: Default execution parameters for operator learning cases.

Case 4.3 dataset size accounts for all 10 saved temporal snapshots.

Case	Input Distribution u	m	Train	Test	Iterations	Physics Domain
4.1	GRF ($l = 0.2$)	100	10,000	100,000	50,000	1D Diffusion
4.2	GRF ($l = 0.2$)	100	10,000	100,000	100,000	Nonlinear ODE
4.3	GRF ($l = 0.2$)	100	800,000	200,000	20,000	2D PCM Melting

Table 2: DeepONet network configurations.

Case 4.3 uses normalized inputs and expanded network dimensions. Trunk sub-networks use tanh activations; branch sub-networks use ReLU activations.

Case	Type	Trunk Depth	Trunk Width	Branch Depth	Branch Width	p	Total Params
4.1	Stacked/Unstacked	3	40	2	40	40	$\sim 10K$
4.2	Unstacked	3	40	2	40	40	$\sim 10K$
4.3	Unstacked (norm)	5	200	5	200	100	$\sim 302K$

]

Table 3: Thermophysical properties of Paraffin wax (RT82).

Properties in the upper block are used in the reference solver; lower block parameters relate to proposed convective model extensions.

Symbol	Parameter Description	Value
Reference Solver Properties		
ρ	Density	770 kg/m ³
C_p	Specific heat capacity	2000 J/(kg·K)
k	Thermal conductivity	0.2 W/(m·K)
L_f	Latent heat of fusion	200,000 J/kg
T_s	Solidus temperature	315 K
T_l	Liquidus temperature	325 K
T_c	Initial/Cold wall temperature	300 K
L	Domain length	50 mm
H	Domain height	50 mm
Navier–Stokes Extension Parameters (Reference Only)		
μ	Dynamic viscosity	0.02685 Pa·s
β	Volumetric thermal expansion	0.001 K ⁻¹
$Amush$	Mushy zone morphology constant	10 ⁵
ε	Division stability parameter	10 ⁻³
Derived Dimensionless Metrics		
Ste	Stefan number	0.25
AR	Domain aspect ratio	1
Pr	Prandtl number (reference only)	≈ 268.5
Ra	Rayleigh number (reference only)	$\approx 6.8 \times 10^6$

4.1. 1D Benchmark Validations

Before evaluating 2D phase change systems, the DeepONet model was verified using two 1D operator mapping benchmarks. For the linear integration mapping $G : u(x) \mapsto \int_0^x u(\tau) d\tau$, the unstacked network reached a test Mean

Squared Error (MSE) of $\sim 10^{-4}$ while maintaining minimal generalization gap (Appendix A). For the non-autonomous dynamical system $ds/dx = -s^2 + u$, test MSE proved roughly four times larger than training loss, illustrating performance sensitivity when

mapping stiff nonlinear dynamics without physics-informed constraints (Appendix B).

4.2. 2D Phase Change and Thermal Energy Storage Evaluation

The network was evaluated on 2D PCM melting processes (Eqs. (3)–(7)) using properties listed in Table 3. Baseline reference datasets were calculated using an explicit finite-difference code on an 80×80 grid.

4.2.1. Convergence Analysis

Validation error declines as trunk basis rank p and training sample count $\#u$ increase, eventually reaching an error plateau (Fig. 2).

Prior to reaching this plateau, error behavior satisfies power-law relationships:

$$\text{MSE}_{\text{test}} \propto p^{-\alpha}, \text{MSE}_{\text{test}} \propto (\#u)^{-\beta} \quad (16)$$

An empirical fit of $\text{MSE} \propto N^{-1.80}$ was calculated across dataset sample sizes ($N = 20, 40, 60, 80, 100$; Fig. 2). This convergence relationship represents preliminary data generated during earlier pipeline tests; while the overall trend of monotonic error reduction remains consistent, exact scaling exponents require multi-seed verification. This empirical power law reflects empirical sample size scaling N , which is distinct from sensor density bounds m established in Theorem 3.

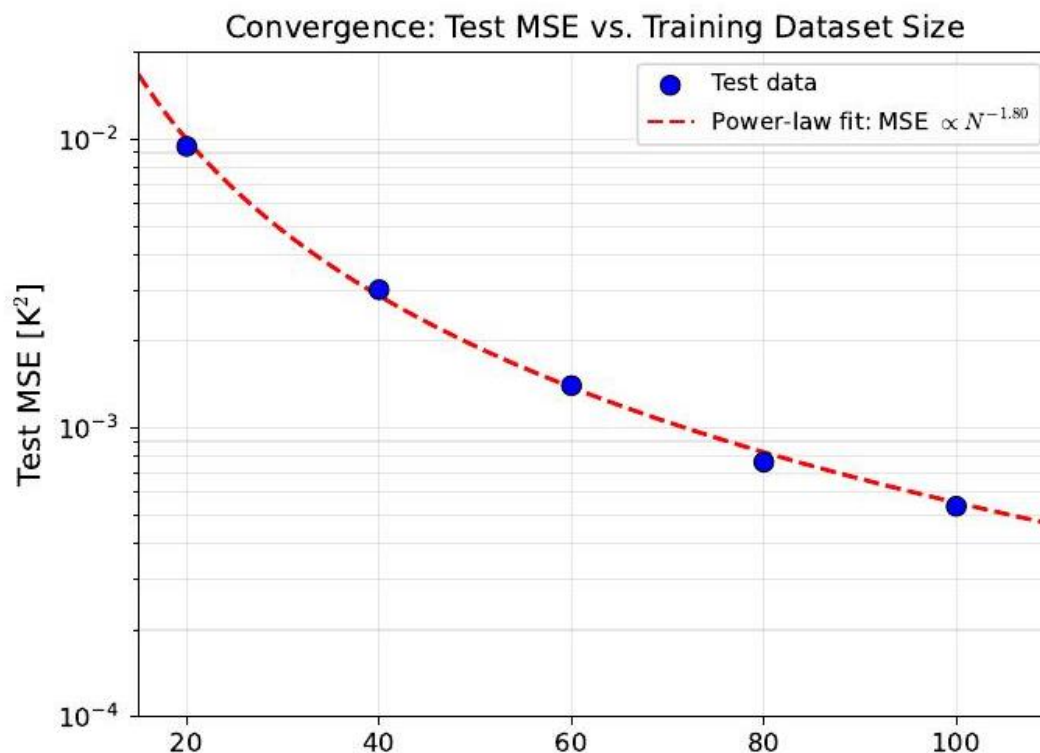


Figure 2: Test error convergence with respect to training sample size.

Test MSE trajectory across input sample sizes ($N = 20$ to $N = 100$), illustrating an empirical power-law trend of $\text{MSE} \propto N^{-1.80}$ (red dashed line). Exponent values are preliminary as noted in Limitation 6.

4.2.2. Temperature Field Reconstruction

Figure 3 compares FDM reference solutions with DeepONet thermal field predictions at the final evaluation step ($t = 180$ s). The reference

field exhibits thermal boundary layer development adjacent to the left boundary. Because transport is governed by pure conduction, isotherms remain primarily parallel to the heated boundary, with minor vertical variation introduced by boundary flux non-uniformity $u(z)$. The DeepONet framework accurately reconstructs this spatial thermal field.

Model performance was evaluated across 20 held-out test simulations using two metrics:

- **Trajectory-averaged relative L^2 error** across 10 temporal steps ($t = 0, 20, \dots, 180$ s) sampled across 1,000 spatial query coordinates: 3.2×10^{-4} (0.032%).

- **Final-snapshot relative L^2 error** evaluated at $t = 180$ s: 3.5×10^{-4} (0.035%).

Snapshot relative errors range from 0.025% at early time steps (where thermal conditions remain relatively uniform) to 0.039% at late

stages (where localized thermal gradients steepen). These error levels improve upon naive baseline approximations (predicting $T = T_c$ yields 1.24% error; predicting global training spatial means yields 0.19% error) by an order of magnitude. Normalizing input/output distributions preserves sharp temperature gradients across the narrow mushy zone ($T_s = 315$ K to $T_l = 325$ K) without interface smearing.

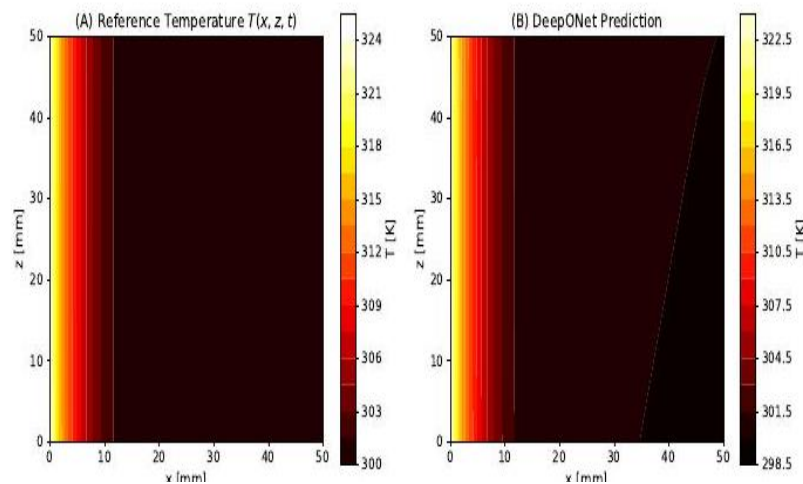


Figure 3: Thermal field distribution comparison at $t = 180$ s.

(A) FDM numerical benchmark $T(x, z, t)$ illustrating heat diffusion from the boundary. (B) DeepONet predicted temperature distribution. Trajectory-averaged relative L^2 error: 3.2×10^{-4} ; final snapshot relative L^2 error: 3.5×10^{-4} .

4.2.3. Liquid Fraction Predictions

Figure 4 displays numerical reference and predicted liquid fraction fields $\phi(x, z, t)$. The reference solution depicts melt-front propagation originating from the left boundary, with the isoline $\phi = 0.5$ representing the phase

change boundary. The DeepONet model captures this structural propagation interface. Recovering liquid fraction values directly from thermal predictions using Eq. (5) ensures thermodynamic consistency across solidus and liquidus boundaries.

At the final snapshot, the relative L^2 error for the liquid fraction field is 7.2%. This error magnitude relative to temperature predictions (0.035%) stems from the piecewise definition of $\phi(T)$, where small temperature variations near transition thresholds T_s and T_l map to larger local phase fraction differences.

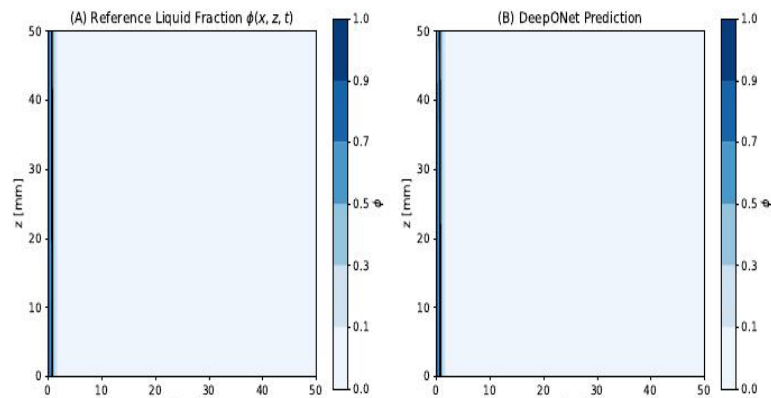


Figure 4: Liquid fraction field and phase change boundary.

(A) Reference liquid fraction field $\phi(x, z, t)$ computed via FDM, with the phase boundary highlighted by the $\phi = 0.5$ contour (solid black line). (B) Recovered liquid fraction field (relative L^2 error: 7.2%), determined via post-processing step Eq. (5).

4.2.4. Reference Thermal Diagnostics

Figure 5 presents heat transfer diagnostics calculated from reference numerical

simulations, including space-averaged Nusselt numbers and wall heat flux distributions. Figure 5A traces the temporal decay of average Nusselt numbers $Nu(t)$, reflecting increased conductive resistance as liquid melt thickness grows. Figure 5B illustrates boundary heat flux profiles $q(z, t)$ at $t = 180$ s. These diagnostic curves describe the physical regime generated by the FDM reference solver.

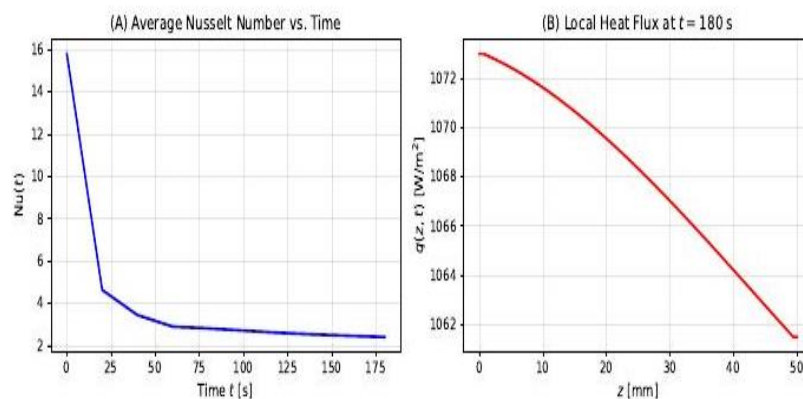


Figure 5: Reference solver thermal diagnostics.

(A) Temporal evolution of space-averaged Nusselt number $Nu(t)$ from FDM simulations. (B) Local wall heat flux distribution $q(z, t)$ evaluated at $t = 180$ s.

4.3. Computational Efficiency and Execution Benchmark

To evaluate model execution speed, execution times for both numerical FDM and DeepONet forward evaluations were measured under identical CPU software environments. Ten complete FDM simulation runs (80×80 spatial grid, $\Delta t = 0.5$ s, $t_{\text{final}} = 200$ s) yielded an

average solve time of 6.90 ± 0.02 s per simulation. DeepONet forward-pass execution was timed across 1,000 passes (evaluating 10,000 spatial query points per pass following 100 warm-up runs), yielding an average execution time of 8.18 ± 0.85 ms. Reported values exclude pre- and post-processing operations.

The resulting execution speedup ratio is calculated as:

$$S = t_{\text{FDM}} / t_{\text{DeepONet}} = 6.90 / 0.00818 \approx 844 \text{ (17)}$$

Propagating measurement variance yields an estimated acceleration interval of $S \approx 844 \pm 87$ ($750\times$ to $930\times$).

Implementation Dependencies. The measured 6.90 s benchmark reflects a non-vectorized Python FDM solver utilizing explicit loop structures. Re-implementing the same FDM algorithm using vectorized NumPy array slicing

reduces execution time to ~ 75 ms per run, which adjusts the speedup ratio to $\sim 100\times$. Thus, speedup ratios reflect comparisons against explicit script implementations rather than absolute limits of numerical schemes. These metrics represent single-forward-pass evaluations and do not incorporate offline data generation or network training costs.

Table 4: Execution timing benchmarks for numerical FDM and DeepONet models.

Metric Parameter	Numerical FDM	DeepONet
Average Solve Time	6.90 s/case	8.18 ms/inference
Standard Deviation	0.02 s	0.85 ms
Evaluated Sample Runs	10	1,000
Warm-up Iterations	—	100
Query Point Evaluation	—	10,000
Grid Resolution	80×80	—
Time Step (Δt)	0.5 s	—
Total Duration (tmax)	200 s	—
Pre/Post-processing	Excluded	Excluded

5. Methodological Limitations

We explicitly outline key physical and computational constraints of this study to clarify scope and guide future extensions:

Limitation 1: Conduction-Only Physical Constraints. All reported metrics evaluate a pure thermal conduction model with effective heat capacity (Eqs. (3)–(7)). Buoyancy-driven natural convection — which alters melt boundary shape in liquid phases — is omitted. Given the Rayleigh number ($Ra \approx 6.8 \times 10^6$), convective transport would play a role in physical enclosures. Retraining network models on coupled Navier–Stokes/enthalpy-porosity datasets remains an important next step.

Limitation 2: Single Domain Geometry and Material Properties. Evaluations are restricted to a single square cavity (50×50 mm) containing paraffin RT82. Performance across alternative domain geometries (e.g., rectangular, cylindrical, or finned cavities) or distinct PCM classes (such as inorganic salt hydrates) requires additional validation.

Limitation 3: Absence of Empirical PI-DeepONet Baselines. The PI-DeepONet formulation (Section 2.3) is presented

conceptually and has not been numerically trained or evaluated. Assessing whether physics-informed loss terms improve data efficiency in phase change systems requires further empirical studies.

Limitation 4: Single Random Seed Evaluations. Reported performance metrics reflect single random seed initializations (seed 42). Model performance can exhibit minor seed-dependent variation; for example, re-running execution passes with altered internal random sequences yielded average trajectory relative errors ranging between 0.021% and 0.032%. Multi-seed evaluations reporting mean and standard deviation metrics are recommended for future benchmarks.

Limitation 5: Absence of Alternative Operator Architecture Baselines. This study does not perform direct empirical comparisons against alternative operator models, such as Fourier Neural Operators (FNOs) [9] or Convolutional Neural Operators (CNOs) [13]. Consequently, comparative architecture performance on phase change problems remains to be established.

Limitation 6: Preliminary Dataset Convergence Scaling. The empirical convergence fit ($MSE \propto N^{-1.80}$, Fig. 2) was derived from preliminary dataset runs and requires formal verification across unified pipelines using multiple random seeds.

Limitation 7: Smooth Boundary Condition Distributions. Input function spaces are constructed exclusively via smooth Gaussian Random Fields ($l = 0.2$). Generalization toward non-smooth, high-frequency, or transient thermal boundaries has not been assessed.

Limitation 8: Solver Implementation Benchmark Dependence. The observed 844 $\times$ speedup compares DeepONet forward passes against an explicit, non-vectorized Python numerical code. Comparing execution against vectorized NumPy algorithms reduces relative acceleration to $\sim 100\times$.

6. Conclusion

This manuscript evaluated Deep Operator Networks (DeepONets) applied to learning non-uniform thermal mappings in 2D Phase Change Material (PCM) energy storage systems governed by effective heat capacity formulations. The unstacked, normalized DeepONet architecture reached a trajectory-averaged relative L^2 temperature error of 3.2×10^{-4} (0.032%) and a final snapshot error of 3.5×10^{-4} (0.035%), improving upon standard baseline estimates. Post-processed liquid fraction fields yielded a relative L^2 error of 7.2%, reflecting sensitivity along piecewise thermal transformation boundaries $\phi(T)$. Furthermore, a sensor interpolation bound scaling as $1/m^2$ was derived mathematically. Primary constraints center on omitting fluid motion and natural convection within reference numerical solvers. While the framework surrogates conduction heat transfer, physical systems operating at $Ra \sim 10^7$ generate buoyancy-driven flows that alter phase interface trajectories. Extending this operator methodology to coupled momentum-energy transport, validating physics-informed residual constraints, and benchmarking performance against alternative operator architectures represent clear directions for future research.

REFERENCES

- G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- T. Chen and H. Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE Trans. Neural Networks*, vol. 6, no. 4, pp. 911–917, 1995.
- H. Chen and T. Chen, "Approximation capability to functions of several variables, nonlinear functionals, and operators by radial basis function neural networks," *IEEE Trans. Neural Networks*, vol. 6, no. 4, pp. 904–910, 1995.
- L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, pp. 218–229, 2021.
- M. M. Farid, A. M. Khudhair, S. A. K. Razack, and S. Al-Hallaj, "A review on phase change energy storage: materials and applications," *Energy Conversion and Management*, vol. 45, no. 9–10, pp. 1597–1615, 2004.
- V. R. Voller and M. Cross, "Accurate solutions of moving boundary problems using the enthalpy method," *Int. J. Heat Mass Transfer*, vol. 24, no. 3, pp. 545–556, 1981.
- A. D. Brent, V. R. Voller, and K. J. Reid, "Enthalpy-porosity technique for modeling convection-diffusion phase change: application to the melting of a pure metal," *Numerical Heat Transfer*, vol. 13, no. 3, pp. 297–318, 1988.

- Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier neural operator for parametric partial differential equations," in *Proc. ICLR*, 2021.
- N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Neural operator: Graph kernel network for partial differential equations," in *Proc. ICLR*, 2023.
- P. Jin, S. Pang, Z. Hou, G. Li, and G. E. Karniadakis, "Multi-fidelity neural operator for learning partial differential equations with incomplete data," *J. Comput. Phys.*, vol. 474, 111838, 2023.
- H. Diab, G. Kevrekidis, and G. E. Karniadakis, "U-DeepONet: Enhanced deep operator network for PDEs with incomplete or noisy data," *J. Comput. Phys.*, vol. 494, 112518, 2023.
- R. Raonic, R. Liu, M. R. P. R. de Oliveira, L. H. G. Ferreira, G. C. P. Silva, R. P. Cardoso, and N. M. Mateus, "Convolutional neural operators for robust and accurate learning of PDEs," *J. Comput. Phys.*, vol. 498, 112697, 2024.
- G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, pp. 422–440, 2021.
- 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," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.
- J. F. A. Torres, D. M. Rivera, and G. E. Karniadakis, "Adaptive physics-informed neural networks: A review," *arXiv preprint arXiv:2501.08543*, 2025.
- G. Priyadarshi, C. Murali, S. Agarwal, and B. K. Kiran Naik, "Parametric investigation and optimization of phase change material-based thermal energy storage integrated desiccant coated energy exchanger through physics informed neural networks oriented deep learning approach," *Journal of Energy Storage*, vol. 80, Article 110231, 2024.
- M. H. Ranjbar, K. Gharali, and A. Ng, "End-to-end deep learning temperature prediction algorithms of a phase change material from experimental photos," *Energy Science & Engineering*, vol. 13, no. 6, pp. 3410–3421, 2025.
- M. Hassanzadeh, E. Ghaderi, F. Fatahi, and M. A. Bijarchi, "An adaptive physics-driven deep learning framework for a two-phase Stefan problem," *arXiv preprint arXiv:2512.01011*, 2025.
- C. Zhang, Z. Gan, Z. Jiang, Z. Xu, and Q. Yang, "An integrated surrogate and material optimization framework for PCM-enhanced energy piles based on EC-DeepONet," *Energy*, vol. 360, Article 141529, 2026.

Appendix A. 1D Integration Operator Benchmark

Consider a linear dynamical ODE system governed by $f(s(x), u(x), x) = u(x)$, which models the linear integration operator mapping:

$$G : u(x) \mapsto s(x) = \int_0^x u(\tau) d\tau \quad (\text{A.1})$$

Standard feedforward networks evaluated across depth (2–4 layers) and width (10–2560 nodes) parameters achieve baseline test MSEs of $\sim 10^{-4}$. Unstacked DeepONets utilizing bias terms reach stable loss convergence over 20,000 optimization steps (Fig. A.6), confirming basic operational efficacy.

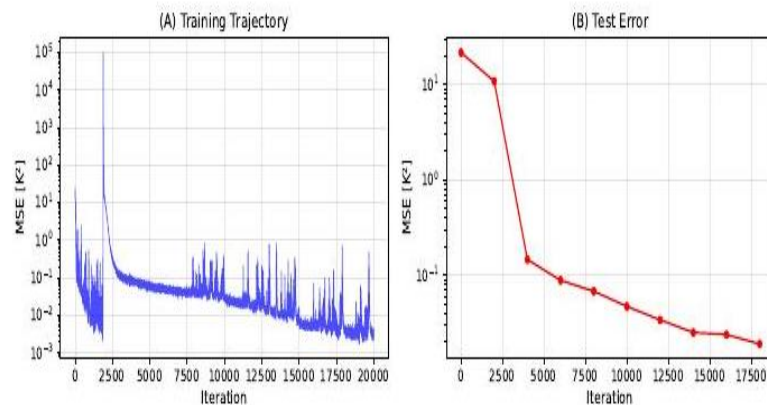


Figure A.6: DeepONet optimization trajectories on 1D integration mappings. (A) Training MSE over 20,000 steps using step-wise learning rate decay. (B) Test set MSE evaluated at 2,000-step intervals. Final test MSE reaches $\sim 10^{-4}$.

Appendix B. 1D Nonlinear ODE System Benchmark

We analyze $f(s(x), u(x), x) = -s^2(x) + u(x)$, which models quadratic damping behavior. Test set error (MSE $\sim 3.3 \times 10^5$) exceeds training loss (MSE $\sim 8.3 \times 10^4$) by approximately four times, demonstrating a distinct generalization gap (Fig. B.7).

Evaluating out-of-distribution test inputs in Fig. B.8 indicates divergence from RK4 numerical solutions as spatial coordinate x advances. Because system dynamics $-s^2 + u$ exhibit strong local trajectory sensitivity, minor errors in operator approximations accumulate downstream. This benchmark highlights performance limits of non-physics-informed operator learning when applied to stiff nonlinear systems outside training distribution boundaries.

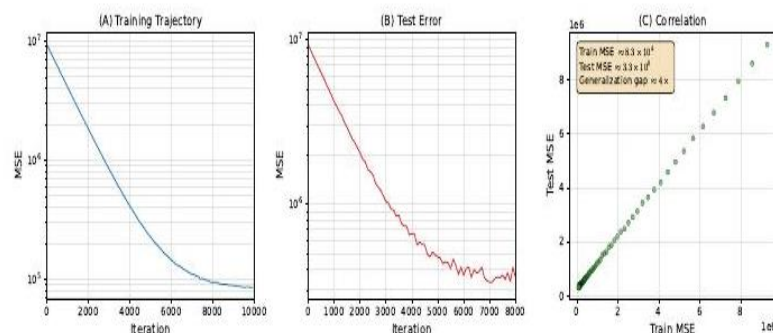


Figure B.7: Nonlinear ODE system error analysis. (A) Training loss trajectory for system $ds/dx = -s^2 + u$. (B) Test set MSE evaluated during training. (C) Correlation mapping between training ($\approx 8.3 \times 10^4$) and test loss ($\approx 3.3 \times 10^5$), showing a $4\times$ generalization gap.

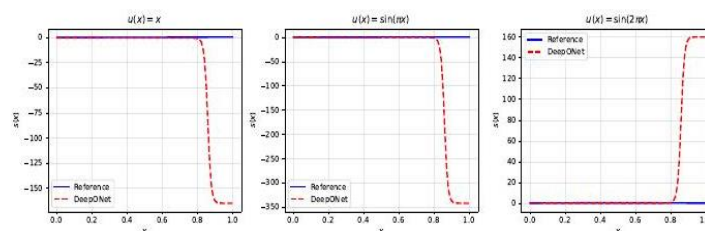


Figure B.8: Nonlinear ODE evaluation profiles. Out-of-distribution evaluation for input functions $u(x) = x$, $u(x) = \sin(\pi x)$, and $u(x) = \sin(2\pi x)$. Solid blue curves: RK4 baseline; dashed red curves: DeepONet predictions.

Appendix C. Proof of Lemma 2

For Gaussian random fields generated using RBF kernels $k_l(z_1, z_2) = \exp(-\|z_1 - z_2\|^2/(2l^2))$, sample paths are smooth and infinitely differentiable. On a uniform domain partition $[a, b]$ with spacing $h = (b - a)/m$, piecewise-linear interpolation errors for twice-differentiable functions satisfy:

$$\max_{z \in [a, b]} |u(z) - u_m(z)| \leq (h^2/8) \max_{z \in [a, b]} |u''(z)|$$

For RBF kernels, second-derivative path magnitudes are bounded in expectation by C/l^2 . Substituting $h = (b - a)/m$ yields:

$$\kappa(m, V) \sim (b - a)^2/(8m^2) \cdot C/l^2 \sim 1/(m^2 l^2)$$

completing the proof.

Appendix D. Proof of Theorem 3

Let $u \in V$ and $u_m = L_m(u) \in U_m$ represent its piecewise linear interpolant. Define the trajectory error as $e(t) = (Gu)(t) - (Gu_m)(t)$. From operator equation (11) and Lipschitz conditions (15):

$$\|e(t)\| = \left\| \int_0^t [f((Gu)(\tau), u(\tau), \tau) - f((Gu_m)(\tau), u_m(\tau), \tau)] d\tau \right\| \leq \int_0^t [c\|e(\tau)\| + c|u(\tau) - u_m(\tau)|] d\tau \leq c \int_0^t \|e(\tau)\| d\tau + c(b - a)\kappa(m, V)$$

Applying Grönwall's Inequality yields:

$$\|e(t)\| \leq c(b - a)\kappa(m, V)e^{ct} \leq c(b - a)\kappa(m, V)e^{c(b-a)}$$

Constraining this error bound below ε yields the condition $c(b-a)\kappa(m, V)e^{c(b-a)} < \varepsilon$.

At coordinate point d , mapping $[u(x_0), \dots, u(x_m)]^T \mapsto (Gu_m)(d)$ represents a continuous function from $\mathbb{R}^{(m+1)}$ to $\mathbb{R}^K$. By neural network universal approximation theorems [1], weight matrices W_1, W_2 and biases b_1, b_2 exist such that mapping errors remain bounded within $\varepsilon - c(b-a)\kappa(m, V)e^{c(b-a)} > 0$. Combining this result with interpolation bounds completes the proof.