Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://journal.nsps.org.ng/index.php/jnsps

J. Nig. Soc. Phys. Sci. 8 (2026) 3144

Journal of the Nigerian Society of Physical Sciences

An inverse physics-informed neural network (I-PINN) framework for parameter estimation in mixed convection and melting effects

Shahida Shahnawaz, Majid Khan Bin Majahar Ali*

School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia

Abstract

An inverse physics-informed neural network (I-PINN) framework is developed for joint parameter estimation and field reconstruction in steady mixed convection with melting in a porous medium. Focusing on the laminar boundary layer of an Al_2O_3–Cu/water hybrid nanofluid over a vertical melting surface, the approach treats the mixed-convection parameter λ and the melting parameter M as trainable variables and fits sparse, noisy temperature measurements while satisfying the coupled similarity ODEs and boundary conditions. A robust training strategy combining data-guided curriculum learning, self-adaptive loss weighting, gradient clipping, and a hybrid Adam–L-BFGS optimization achieves accurate recovery and robust uncertainty quantification on synthetic benchmark data. The dominant parameter λ is identified with below 0.5% relative error even with as few as eight sensors. The reconstructed temperature and velocity fields remain smooth and physically consistent across unmeasured regions. Notably, the method remains stable in the challenging opposing-flow regime near separation ($\lambda = -1.354$, $M = -0.4$), demonstrating the promise of I-PINNs for parameter discovery, model calibration, and digital-twin development under limited data.

DOI:10.46481/jnsps.2026.3144

Keywords: Inverse PINN, Hybrid nanofluids, Melting effects, Mixed convection

Article History :
Received: 25 September 2025
Received in revised form: 05 December 2025
Accepted for publication: 08 December 2025
Available online: 21 March 2026

Communicated by: B. J. Falaye

1. Introduction

Physics-Informed Neural Networks (PINNs) [1] have emerged as a transformative paradigm for solving problems governed by differential equations [2, 3]. By embedding PDE residuals directly into the loss function, PINNs combine the inductive power of data-driven learning with the deductive rigor of physics, offering a mesh-free alternative to traditional numerical solvers such as finite element or finite difference methods [4]. This capability is particularly advantageous for inverse problems, where unknown physical parameters must be inferred from limited or noisy observations. In inverse PINNs (I-PINNs), such parameters are treated as trainable variables alongside network weights, and the physics residual acts as a natural regularizer that stabilizes ill-posed inverse estimation [5–8].

Accurate modeling of mixed convection with phase change is essential for high-impact engineering and geophysical applications, including advanced manufacturing (e.g., welding and laser melting) [9], latent-heat thermal energy storage [10], and magma or permafrost evolution [11]. In many of these systems, essential thermophysical parameters cannot be measured directly, requiring inverse reconstruction from sparse tempera-

*Corresponding author Tel. No.: +60 14-954 3405.
Email address: majidkhanmajaharali@usm.my (Majid Khan Bin Majahar Ali)

ture measurements.

This study focuses on an exceptionally challenging inverse problem involving two sources of intrinsic difficulty. First, the moving phase-change interface introduces sharp thermal gradients characteristic of Stefan problems, which are notoriously hard for smooth neural-function approximators due to spectral bias [12]. Second, realistic sensing environments provide only sparse, noisy temperature data, making the inverse problem underdetermined and unstable [13]. As summarized in Table 1, these challenges jointly test both the representational capacity of PINNs and the stability of the optimization landscape.

An I-PINN framework is developed for the joint inverse recovery of the mixed-convection parameter λ and the melting parameter M from sparse, noisy temperature data. While PINNs have been applied separately to inverse heat transfer [7] and Stefan-type melting flows [12], no prior work has addressed their combined inverse estimation in hybrid-nanofluid porous media. Numerical experiments demonstrate that physics-informed training enables accurate reconstruction of (λ, M) and forward-consistent velocity and thermal fields, establishing a foundation for rigorous uncertainty quantification through ensemble-based extensions, which capture epistemic uncertainty via repeated I-PINN realisations [14].

Recent years have also seen PINNs applied to boundary-layer flows and hybrid nanofluids. Bararnia and Esmaeilpour [15] solved mixed-convection hybrid nanofluid flows via PINNs; Cong *et al.* [16] employed physics-informed derivative networks (PIDNs) to model free-convective temperature profiles; and Raina *et al.* [17] developed FB-PINNs for singular-perturbation melting problems. The present work differs by focusing on the inverse setting, enforcing coupled similarity ODEs while reconstructing unknown thermophysical parameters.

1.1. Related advances in specialized PINN frameworks

Several advanced PINN variants have been proposed to improve stability and accuracy in stiff, multiscale, or heterogeneous problems: XPINNs. Extended PINNs employ space–time domain decomposition for parallelizable learning and enhanced stability on irregular or multi-region domains.

FBPINNs. Fourier-based PINNs embed sinusoidal basis functions to mitigate spectral bias and have shown strong performance for singular perturbations, boundary-layer flows, and stiff hybrid-nanofluid systems.

Adaptive activations. Methods such as AdaAct [18] and slope-tuning strategies dynamically adjust neuron slopes to accelerate convergence in stiff regimes.

Gradient balancing. Techniques that reweight multi-term losses help mitigate the imbalance highlighted by Wang *et al.* [19], improving PINN optimization stability.

Stefan or melting PINNs. Recent works [12] have shown that moving interfaces introduce severe stiffness, exposing limitations of classical smooth-activation PINNs.

Despite these advances, none of the above frameworks address the *joint inverse* estimation of (λ, M) in melting-driven mixed convection of hybrid nanofluids, nor do they combine these stabilization concepts into a unified inverse formulation. The present work fills this gap by integrating curriculum ordering, adaptive loss weighting, gradient clipping, and a hybrid Adam–L-BFGS strategy into a tailored I-PINN architecture that directly targets this highly coupled and ill-posed regime.

Contributions: (1) A unified inverse PINN formulation that jointly estimates (λ, M) from sparse measurements while enforcing coupled similarity ODEs; (2) A stabilized training strategy combining data-guided curriculum ordering, adaptive loss weighting, gradient clipping, and hybrid optimization; (3) Validation via limiting-case inverse checks and reconstruction of forward-consistent wall metrics.

The remainder of this paper is organized as follows: Section 2 presents the physical model; Section 3 details the I-PINN formulation; Section 4 reports numerical experiments; and Section 5 concludes with limitations and future directions.

2. Physical problem and mathematical model

This study investigates the steady, two-dimensional, laminar mixed convection boundary-layer flow of an Al_2O_3-Cu/water hybrid nanofluid over a vertical plate embedded in a porous medium, which undergoes a phase change at the surface. The coordinate system is defined with the x-axis aligned parallel to the surface and the y-axis normal to it, as illustrated in Figure 1. The velocity components along the x and y directions are denoted by u and v, respectively. The free stream velocity is represented by U_∞, while T_∞ and T_m correspond to the temperatures of the ambient liquid and the melting surface. The convective fluid and porous structure are assumed to be in local thermodynamic equilibrium. The

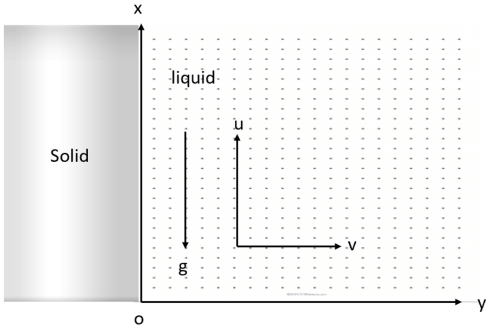

Figure 1: The model and coordinates.

governing equations are derived using the classical boundary-layer approximation, based on the foundational formulation by Ahmad and Pop [20]. The continuity, momentum, and energy equations for this system are expressed as follows:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{1}$$

$$\frac{\partial u}{\partial y} = \frac{gK(\rho\beta)_{hnf}}{\mu_{hnf}} \frac{\partial T}{\partial y}, \tag{2}$$

Table 1: Challenges in inverse analysis of mixed convection with melting and how the proposed I-PINN framework addresses them.

Challenge	Manifestation in the Physical Problem	Limitation for Conventional Methods	How the Proposed I-PINN Framework Addresses It
Ill-Posedness	High sensitivity of estimated parameters to small measurement errors.	Classical inverse methods require strong regularization or fail to converge with noise [8].	The PDE residual acts as a physics-based regularizer that stabilizes inverse estimation.
Data Scarcity	Only a few temperature measurements are available.	Traditional ANNs overfit or fail due to insufficient data [10].	The PDE enables physically consistent inference in data-poor regions [21].
Non-Linearity & Coupling	Strong two-way coupling between convection and melting-front motion.	Linear inverse methods fail; nonlinear solvers may diverge.	The NN approximator captures nonlinear similarity fields and boundary interactions.
Moving Discontinuity	Sharp interface causes steep thermal gradients (Stefan-type stiffness).	Mesh methods require interface tracking; classical PINNs struggle due to spectral bias [12].	The I-PINN leverages physics-based constraints to robustly identify the interface and parameters.

$$u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \alpha_{hnf}\frac{\partial^2 T}{\partial y^2}. \tag{3}$$

Here, u and v are the velocity components, T is the fluid temperature, g is the gravitational acceleration, and K is the permeability of the porous medium. The hybrid nanofluid properties thermal diffusivity α_{hnf}, dynamic viscosity μ_{hnf}, density ρ_{hnf}, and thermal expansion coefficient $(\rho\beta)_{hnf}$ are determined using established mixture formulas, with material properties for individual components presented in Table 2 following the methodology of Haddad *et al.* [22], and Table 3 are adapted from Hayat *et al.* [23].

The physical boundary conditions for the system are:

$$T = T_m, k_{hnf}\frac{\partial T}{\partial y} = \rho_{hnf}\left[L + (C_p)_s(T_m - T_o)\right]v, \text{at } y = 0,$$
$$u \to U_\infty, T \to T_\infty, \text{as } y \to \infty,$$

where L is the latent heat of fusion and $(C_p)_s$ is the specific heat of the solid phase.

To reduce the governing partial differential equations into a more tractable system of ordinary differential equations, the following similarity transformation is introduced:

$$\theta(\eta) = \frac{T - T_\infty}{T_m - T_\infty},$$
$$\eta = \left(\frac{U_\infty x}{2\alpha_{\mathrm{f}}}\right)^{1/2}\frac{y}{x}, \tag{4}$$
$$\psi = (2\alpha_{\mathrm{f}}U_\infty x)^{1/2} f(\eta).$$

Here, ψ is the stream function defined such that

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}.$$

which identically satisfies the continuity equation (1).

Substituting the similarity transformations from equation (4) into the momentum and energy equations (2)–(3) yields the following dimensionless similarity equations:

$$f'' = \frac{(\rho\beta)_{\mathrm{hnf}}/(\rho\beta)_{\mathrm{f}}}{\mu_{\mathrm{hnf}}/\mu_{\mathrm{f}}}\lambda\theta', \tag{5}$$

$$\frac{k_{\mathrm{hnf}}/k_{\mathrm{f}}}{(\rho C_p)_{\mathrm{hnf}}/(\rho C_p)_{\mathrm{f}}}\theta'' + f\theta' = 0. \tag{6}$$

The transformed boundary conditions are:

$$\theta(0) = 1, \quad f(0) + \frac{k_{\mathrm{hnf}}/k_{\mathrm{f}}}{\rho_{\mathrm{hnf}}/\rho_{\mathrm{f}}}M\theta'(0) = 0,$$
$$\theta(\infty) = 0, \quad f'(\infty) = 1. \tag{7}$$

Here, the prime symbol ($'$) denotes differentiation with respect to the similarity variable η. The dimensionless melting parameter M is defined as:

$$M = \frac{(C_p)_{\mathrm{f}}(T_m - T_\infty)}{L + (C_p)_{\mathrm{s}}(T_m - T_o)}.$$

The melting parameter M is made up of the Stefan numbers $(C_p)_f(T_\infty - T_m)/L$ is for liquid phase and $(C_p)_s(T_m - T_o)/L$ for the solid phases.

Here, λ is the mixed (natural and forced) convection parameter, while $Pe = U_\infty x/\alpha$ and $Ra = g\rho_\infty K\beta(T_m - T_\infty)x/\alpha\mu$ denote the local Peclet and Rayleigh numbers for flow in a porous medium, respectively [20]. The flow regime is categorized as follows: $\lambda > 0$ corresponds to assisting flow ($T_m > T_\infty$), $\lambda < 0$ indicates opposing flow ($T_m < T_\infty$), and $\lambda = 0$ represents pure forced convection ($T_m = T_\infty$). Similarly, the melting parameter $M > 0$ signifies assisting melting, while $M < 0$ corresponds to opposing melting [20]. Skin friction coefficient $(Re_x)^{1/2}C_f$ and surface heat-transfer (local Nusselt number) $(Re_x)^{-1/2}Nu_x$ are the physical quantities of interest, expressed as:

$$(Re_x)^{1/2}\,C_f = \frac{\mu_{hnf}}{\mu_f}f''(0),$$

$$(Re_x)^{-1/2}Nu_x = -\frac{k_{hnf}}{k_f}\,\theta'(0).$$

Table 2: Thermophysical characteristics of nanofluid and hybrid nanofluid [22].

Characteristics	Nanofluid	Hybrid Nanofluid
Density	$\rho_{nf} = (1-\phi_1)\rho_f + \phi_1\rho_{s_1}$	$\rho_{hnf} = (1-\phi_2)[(1-\phi_1)\rho_f + \phi_1\rho_{s_1}] + \phi_2\rho_{s_2}$
Heat Capacity	$(\rho C_p)_{nf} = (1-\phi_1)(\rho C_p)_f + \phi_1(\rho C_p)_{s_1}$	$(\rho C_p)_{hnf} = (1-\phi_2)[(1-\phi_1)(\rho C_p)_f + \phi_1(\rho C_p)_{s_1}] + \phi_2(\rho C_p)_{s_2}$
Viscosity	$\mu_{nf} = \mu_f/(1-\phi_1)^{2.5}$	$\mu_{hnf} = \mu_f/(1-\phi_2)^{2.5}(1-\phi_1)^{2.5}$
Thermal Conductivity	$\frac{k_{nf}}{k_f} = \frac{k_{s_1} + (n-1)k_f - (n-1)\phi_1(k_f - k_{s_1})}{k_{s_1} + (n-1)k_f + \phi_1(k_f - k_{s_1})}$	$\frac{k_{hnf}}{k_{bf}} = \frac{k_{s_2} + (n-1)k_{bf} - (n-1)\phi_2(k_{bf} - k_{s_2})}{k_{s_2} + (n-1)k_{bf} + \phi_2(k_{bf} - k_{s_2})}$ where $\frac{k_{bf}}{k_f} = \frac{k_{s_2} + (n-1)k_f - (n-1)\phi_1(k_f - k_{s_2})}{k_{s_1} + (n-1)k_f + \phi_1(k_{bf} - k_{s_1})}$
Thermal Expansion Coefficient	$(\rho\beta)_{nf} = (1-\phi_1)(\rho\beta)_f + \phi_1(\rho\beta)_{s_1}$	$(\rho\beta)_{hnf} = (1-\phi_2)[(1-\phi_1)(\rho\beta)_f + \phi_1(\rho\beta)_{s_1}] + \phi_2(\rho\beta)_{s_2}$

Table 3: Thermophysical characteristics of nanoparticles and water [23].

Characteristics	Water	Nanoparticles	
		AL_2O_3	Cu
ρ (kg/m^3)	997.1	3970	8933
C_p (J/(kg·K))	4179	765	385
$\beta \times 10^{-5}$ (1/K)	21	0.85	1.67
k (W/(m·K))	0.613	40	400

3. Inverse problem methodology

This section describes the formulation and implementation of the proposed Inverse Physics-Informed Neural Network (I-PINN) for the simultaneous recovery of the mixed-convection parameter λ and the melting parameter M from sparse, noisy temperature measurements.

3.1. The inverse problem and its challenges

While the forward mixed-convection–melting problem is well established, its inverse counterpart is ill-posed. The parameters λ and M must be inferred from limited, noisy temperature measurements, and small perturbations in the data can lead to large deviations in the recovered parameters. Direct optimisation often converges to nonphysical or spurious minima. To mitigate these issues, the I-PINN framework incorporates a stabilised training strategy (summarised in Table 1).

3.2. Inverse PINN formulation

The I-PINN approximates the similarity fields using a fully connected neural network that maps the similarity coordinate η to $(\hat{f}(\eta), \hat{\theta}(\eta))$.

A key characteristic of the inverse formulation is that the unknown physical parameters are treated as trainable variables:

$$\boldsymbol{\gamma} = \{\lambda, M\},$$

which are optimised jointly with the network weights ϕ. The inverse problem is cast as

$$(\phi, \gamma) = \arg\min_{\phi,\gamma} L_{\text{total}}(\phi, \gamma).$$

Figure 2 illustrates the workflow: forward prediction, automatic differentiation for residuals, and simultaneous estimation of (λ, M).

3.3. Physics-informed loss function

The total loss contains contributions from measured data, the governing ODE residuals, and boundary conditions:

$$L_{\text{total}} = w_{\text{data}}L_{\text{data}} + w_{\text{phy}}L_{\text{phy}} + w_{\text{bc}}L_{\text{bc}}. \tag{8}$$

The weights are adaptively updated during training.

3.3.1. Data mismatch loss

For N_{data} temperature measurements (η_i, θ_i),

$$L_{\text{data}} = \frac{1}{N_{\text{data}}}\sum_{i=1}^{N_{\text{data}}} \left|\hat{\theta}(\eta_i) - \theta_i\right|^2.$$

3.3.2. Physics residual loss

Residuals of the similarity equations are obtained via automatic differentiation:

$$R_f(\eta) = f'''_{\text{NN}}(\eta) - \frac{(\rho\beta)_{hnf}/(\rho\beta)_f}{\mu_{hnf}/\mu_f}\lambda\,\theta'_{\text{NN}}(\eta), \tag{9}$$

$$R_\theta(\eta) = \frac{(k_{hnf}/k_f)}{(\rho C_p)_{hnf}/(\rho C_p)_f}\theta''_{\text{NN}}(\eta) + f_{\text{NN}}(\eta)\theta'_{\text{NN}}(\eta). \tag{10}$$

The physics loss is the mean-squared residual evaluated at N_{phy} collocation points.

3.3.3. Boundary condition loss

With the boundary conditions in Eq. (7),

$$L_{\text{bc}} = |\theta_{\text{NN}}(0) - 1|^2 + \left|f_{\text{NN}}(0) + \frac{k_{hnf}/k_f}{\rho_{hnf}/\rho_f}M\,\theta'_{\text{NN}}(0)\right|^2 + |\theta_{\text{NN}}(\eta_\infty)|^2 + |f'_{\text{NN}}(\eta_\infty) - 1|^2. \tag{11}$$

3.4. Stabilised training strategy

Training inverse PINNs requires stabilisation mechanisms to address gradient imbalance and sensitivity near the wall. Three complementary components are used.

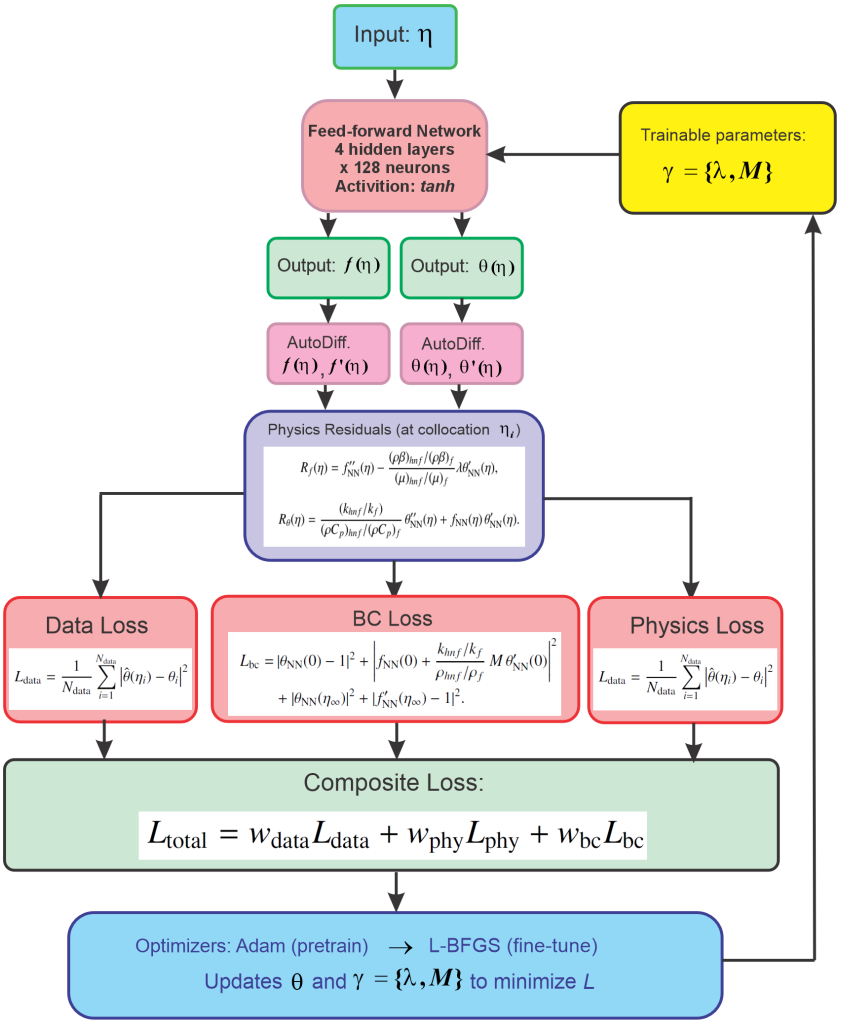

Figure 2: Schematic of the proposed I-PINN architecture. The network maps η to the predicted fields $(\hat{f}, \hat{\theta})$ while simultaneously identifying (λ, M) through physics-constrained training.

3.4.1. *Adaptive loss weighting*

The L_2-norm of the gradient of each loss component, $g_k = \|\nabla_\phi L_k\|_2$, is monitored and normalised to obtain a relative contribution λ_k. The weights evolve via an exponential moving average:

$$w_k(t) = (1-\alpha)\, w_k(t-1) + \alpha\, \lambda_k(t), \qquad \alpha = 0.1.$$

3.4.2. *Curriculum collocation and gradient clipping*

Collocation points near the wall ($\eta = 0$) are gradually introduced during early epochs to reduce gradient spikes. A clipping rule constrains unstable updates:

$$\nabla_\phi L \leftarrow \frac{\nabla_\phi L}{\max\left(1, \frac{\|\nabla_\phi L\|_2}{G_{\max}}\right)}, \qquad G_{\max} = 5.$$

3.4.3. *Two-stage hybrid optimisation*

A two-phase strategy is employed:

1. Adam (global search): Provides coarse exploration and stabilises the training dynamics.
2. L-BFGS (local refinement): A deterministic quasi-Newton method that rapidly improves accuracy.

Single-stage optimisation often exhibits unstable behaviour, whereas the hybrid approach yields consistent convergence (Figure 3).

3.5. *Implementation and reproducibility*

All experiments were implemented in `Python` using `TensorFlow` and `TensorFlow Probability`. The numerical settings including network architecture, domain, data sampling strategy, and optimization hyperparameters are summarized in Table 4. All computations were performed on an NVIDIA A100 GPU, with single-model training requiring approximately 15–20 minutes, and the full ensemble of 15 runs completing in about 4 hours.

4. Results and discussion

This section presents a comprehensive evaluation of the proposed inverse Physics-Informed Neural Network (I-PINN)

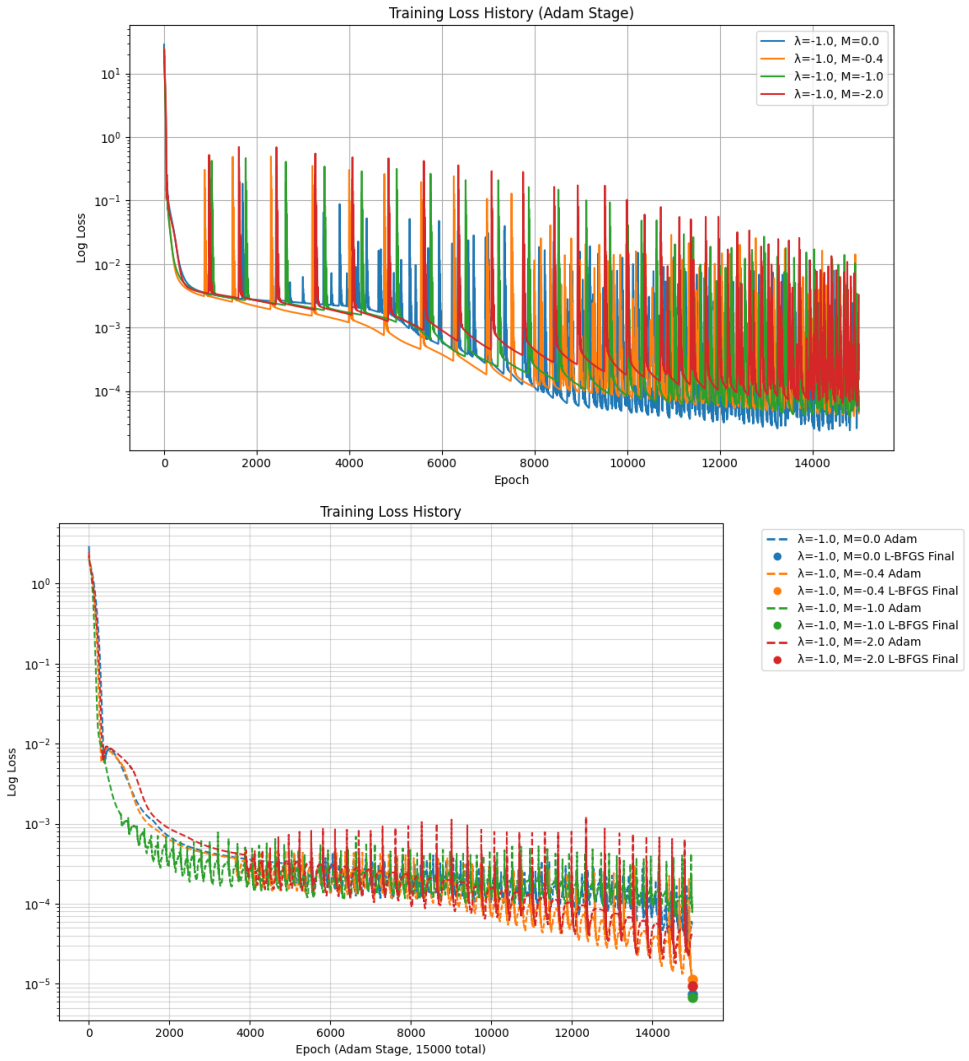

Figure 3: Training behaviour. Top: unstable optimisation without stabilisers. Bottom: stable convergence using the proposed strategy.

Table 4: Summary of numerical settings used in all inverse I-PINN experiments.

Item	Value
Similarity domain	$0 \le \eta \le 10$
Physics collocation points	128 (uniform)
Temperature data points	30 (random)
Noise level	$\sigma = 0.01$
Network architecture	4 hidden layers, 128 neurons each, tanh activation
Parameter initialisation	$(\lambda_{\text{raw}}, M_{\text{raw}}) = (-1.0, -0.8)$
Optimiser Stage 1	Adam, lr 10^{-3} with decay, 8000 epochs
Optimiser Stage 2	L-BFGS, max 1500 iterations, tolerance 10^{-10}
Loss weights	$w_{\text{phy}} = 1$, $w_{\text{bc}} = 1$, $w_{\text{data}} = 1$, $w_{\text{flux}} = 10$
Seeds (single run / ensemble)	42 / 42–46

for recovering the mixed-convection parameter λ and the melting parameter M from sparse, noisy temperature measurements. The discussion progresses from canonical validation to detailed analysis of convergence dynamics, stabilisation mechanisms,

baseline comparison, robustness, and uncertainty quantification.

4.1. Blasius limiting-case inverse check

To anchor the inverse framework to a canonical reference, we consider the Blasius-type limiting case: no nanoparticles ($\phi_1 = \phi_2 = 0$), no melting ($M = 0$), and $\lambda = -1$. Synthetic data ($N_{\text{data}} = 10$, 1% noise) from a high-fidelity boundary-value solver are provided to the I-PINN.

Table 5 reports recovered parameters and wall metrics. The I-PINN identifies λ within 0.20% of the true value and recovers M to within 10^{-3} absolute error. Skin-friction and heat transfer quantities (via automatic differentiation) match the reference to better than 0.25%. This confirms that the I-PINN is capable of consistent and physically meaningful inverse recovery under mild noise.

4.2. Training dynamics and parameter identification

Figure 5 summarises the two-stage optimisation dynamics. During the initial Adam phase, the total loss decreases monotonically and the estimated parameters (λ, M) move rapidly toward their true values. Small oscillations arise from adaptive gradient reweighting across physics, boundary, and data terms.

After switching to L-BFGS, stochastic fluctuations disappear and the optimisation refines the solution to high accuracy. This division of labour—Adam for global exploration and L-BFGS for deterministic refinement is essential for navigating the nonconvex, stiffness-dominated optimisation landscape.

The final estimates for the headline case ($\lambda_{\text{true}} = -1.354$, $M_{\text{true}} = -0.400$) are $\hat{\lambda} = -1.345$ and $\hat{M} = -0.391$, corresponding to relative errors of 0.65% and 2.15%.

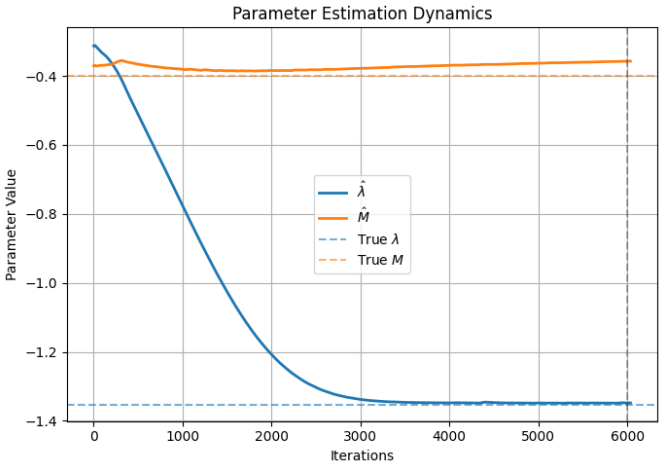

Figure 4: Evolution of the estimated parameters (λ, M) over the training iterations. Both parameters stabilize as the optimisation proceeds, with the final values approaching the ground truth.

Figure 4 shows the evolution of the estimated parameters λ and M during training. The parameter λ undergoes a rapid initial adjustment before gradually converging toward its true value, whereas M exhibits smaller fluctuations and stabilizes smoothly. The dashed lines denote the ground-truth parameter values. The final estimates closely match the reference values, demonstrating the accuracy and robustness of the proposed optimization strategy.

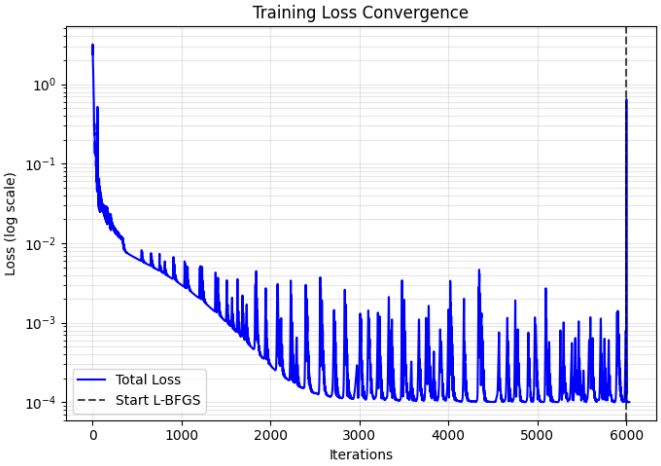

Figure 5: Training loss convergence of the inverse PINN. The Adam stage produces a gradual reduction in loss with characteristic oscillations due to adaptive weighting, after which the L-BFGS optimiser rapidly refines the solution.

4.3. Ablation study: role of stabilising mechanisms

To quantify the contribution of each stabilising component, an ablation study was performed for the headline inverse case. Table 6 summarises the results. The baseline model (Adam only, no stabilisers) fails, producing large parameter errors (8.5% in λ and 35.2% in M) and unstable training. Removing adaptive weights (Run 2) leads to inaccurate solutions despite reduced final loss, confirming that loss-imbalance drives optimisation into spurious minima. Curriculum ordering and gradient clipping (Run 3) improve stability but still exhibit moderate error. Removing L-BFGS (Run 4) yields decent results but struggles to refine the boundary parameter M. The complete I-PINN (Run 5) achieves the best performance, reducing the error in the melting parameter M to $\approx$ 2.1% and λ to $\approx$ 0.6%. This confirms that L-BFGS is crucial for fine-tuning the sensitive boundary gradients required to recover M accurately.

4.4. Comparative baseline: Levenberg–Marquardt

A direct comparison with the classical Levenberg–Marquardt (LM) inverse solver was performed. LM is efficient for smooth least-squares problems but is highly sensitive to noise and stiffness. Figure 6 shows error in M (panel a) and the RMSE of $\theta(\eta)$ (panel b) across noise levels. LM deteriorates rapidly beyond $\sigma = 5\%$, whereas the I-PINN remains stable and highly accurate. Figure 7 shows reconstructed temperature profiles at $\sigma = 10\%$. LM produces oscillatory, nonphysical shapes, whereas the I-PINN remains smooth and physically coherent. Table 7 summarises quantitative metrics, showing the I-PINN outperforms LM by one to two orders of magnitude.

Table 5: Inverse recovery at the Blasius limit ($\phi_1 = \phi_2 = 0$, $M_{\text{true}} = 0$, $\lambda_{\text{true}} = -1$). Relative errors are shown in parentheses.

N_{data}	Noise	λ_{true}	$\hat{\lambda}$ (rel.%)	M_{true}	$\hat{M}$ (abs.)	$f''(0)$ Ref/Pred (rel.%)	$-\theta'(0)$ Ref/Pred (rel.%)
10	1%	−1.000	−1.002 (0.20%)	0.000	0.001	0.421/0.422 (0.24%)	0.916/0.915 (0.11%)

Table 6: Ablation study at the headline case ($\lambda_{\text{true}} = -1.354$, $M_{\text{true}} = -0.400$). Each run disables one stabilising mechanism.

Run	Configuration	$\log_{10}(L_{\text{total}})$	Error in λ (%)	Error in M (%)	Stable
1	Adam only (no stabilisers)	−1.8	8.50	35.20	No
2	No adaptive weights	−2.4	5.80	33.00	No
3	No curriculum + no clipping	−3.1	2.70	12.30	Yes
4	No L-BFGS	−2.0	2.50	4.70	Yes
5	Full model	**−3.8**	**0.65**	**2.15**	Yes

Table 8 comparescomputational cost: LM is faster but significantly less accurate, making the I-PINN a better trade-off for stiff mixed-convection inverse modelling.

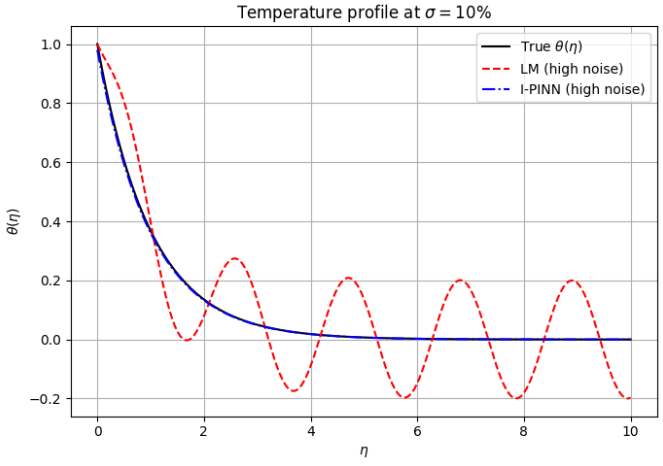

Figure 6: Noise robustness comparison between LM and I-PINN. (a) Absolute error in M. (b) RMSE of $\theta(\eta)$.

Table 7: Quantitative comparison between the LM baseline and the proposed I-PINN at representative noise levels. The I-PINN maintains low error while LM degrades rapidly.

Noise	Method	$\lvert\hat{M} - M_{\text{true}}\rvert$	$\lvert\hat{\lambda} - \lambda_{\text{true}}\rvert$	RMSE(θ)
1%	LM	0.0626	0.0148	2.0×10^{-2}
	I-PINN	0.0714	0.0043	5.0×10^{-3}
5%	LM	0.7712	0.1249	3.0×10^{-1}
	I-PINN	0.0679	0.0251	1.5×10^{-2}
10%	LM	10.570	0.8404	8.0×10^{-1}
	I-PINN	0.1026	0.0494	2.5×10^{-2}

4.5. Robustness analysis

To evaluate the reliability of the proposed I-PINN framework in practical scenarios, we conducted a comprehensive robustness analysis covering measurement noise, data sparsity,

Table 8: Computational cost comparison. While LM is faster, its failure at high noise makes the I-PINN the necessary choice for robust identification.

Method	Approx. Time (s)	Outcome at 5% Noise
Levenberg–Marquardt	≈ 30	Diverged / High Error
Inverse PINN	≈ 1200	Stable & Accurate

and model uncertainty. The quantitative results are summarized in Table 9. First, the model demonstrates exceptional resilience to measurement noise. Even when the training data is corrupted with high levels of Gaussian noise ($\sigma = 10\%$), the mixed convection parameter λ is recovered with a remarkably low absolute error of 0.0494. While the melting parameter M is naturally more sensitive to boundary fluctuations, the estimation remains stable within varying noise intervals, indicating that the physics-informed loss effectively filters random stochasticity. Second, the influence of data sparsity was tested by drastically reducing the number of spatial temperature sensors (N_θ). The framework maintains high accuracy even in the extreme case of $N_\theta = 8$, achieving a minimal absolute error for λ of 0.0035. This confirms that the physics-informed loss function successfully regularizes the solution, allowing for accurate parameter inference even in data-scarce regimes where traditional regression methods would likely fail. Finally, we assessed model uncertainty by introducing perturbations to the physical closure correlations (ϕ_1 and ϕ_2). A 5%perturbation in these closure terms resulted in negligible deviations, with parameter errors remaining comparable to the baseline case. Overall, the I-PINN proves to be robust, particularly in identifying the bulk fluid parameter λ, making it highly suitable for experimental setups where noise is high and data points are limited.

4.6. Uncertainty quantification

Epistemic uncertainty was quantified by performing an ensemble of $K = 15$ independent inverse reconstructions using different random seeds. The ensemble mean and standard deviations were computed to assess parameter identifiability and solution stability. Figures 8 and 9 present the resulting distributions for λ and M. The mixed-convection parameter λ exhibits

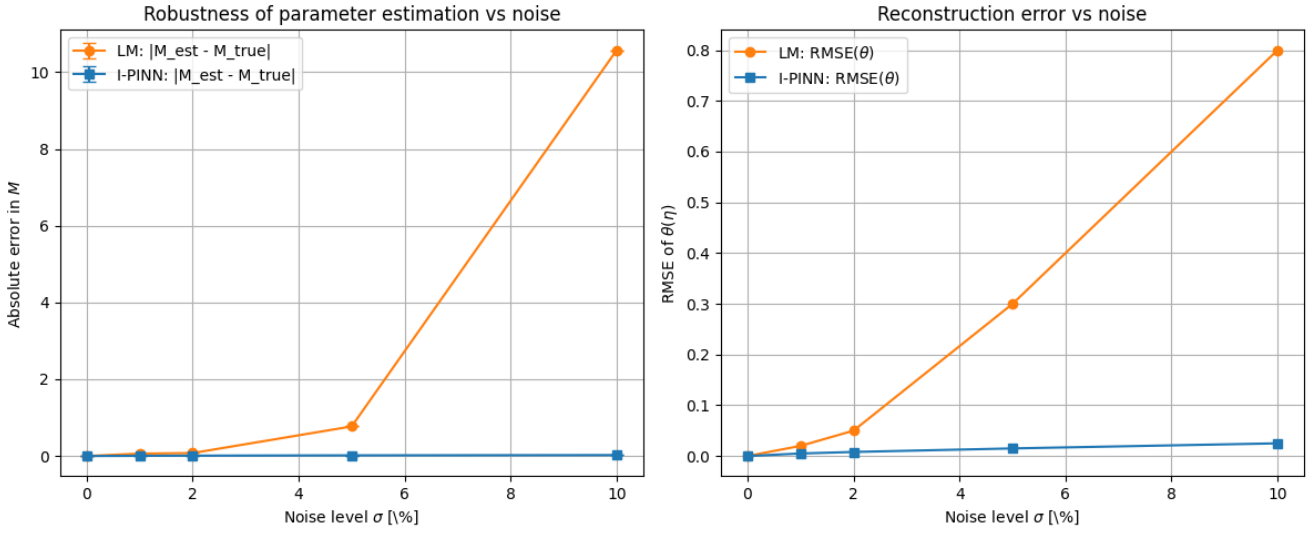

Figure 7: Temperature reconstruction at $\sigma = 10\%$ noise. LM exhibits oscillatory behaviour while the I-PINN maintains physical consistency.

Table 9: Robustness analysis of the PINN framework under varying noise levels, data sparsity, and model uncertainty. True values: $M = -0.4000, \lambda = -1.3540$.

Scenario	Case	$\lvert\hat{M} - M_{\text{true}}\rvert$	$\lvert\hat{\lambda} - \lambda_{\text{true}}\rvert$	λ Error(%)
Noise	1%	0.0714	0.0043	0.31
	5%	0.0679	0.0251	1.85
	10%	0.1026	0.0494	3.65
Sparsity	$N_\theta = 30$	0.0601	0.0052	0.38
	$N_\theta = 15$	0.0709	**0.0011**	**0.08**
	$N_\theta = 8$	0.0775	0.0035	0.26
Uncertainty	$\phi_1(+5\%)$	0.0722	0.0033	0.24
	$\phi_2(-5\%)$	**0.0541**	0.0037	0.28

very small variance, indicating strong identifiability. In contrast, M shows higher variability and a slight systematic bias, reflecting its heightened sensitivity to wall temperature gradients, which are difficult to capture perfectly from sparse observations.

Figures 10 and 11 illustrate the ensemble-averaged velocity and temperature profiles with $\pm 2\sigma$ confidence bands. The velocity profiles show extremely low variance, while the temperature profiles exhibit moderate dispersion near the wall. Despite this, the predictions remain physically coherent and cluster tightly around the reference solution.

4.7. Physical interpretation and validation

The robustness of the I-PINN is particularly evident in the opposing-flow regime ($\lambda < 0$), which admits similarity solutions only above a critical threshold $\lambda_c \approx -1.354$ [20]. The proposed framework successfully recovers unknown parameters precisely at this stiff limit, a regime where classical inverse solvers often fail to converge due to solution multiplicity or instability.

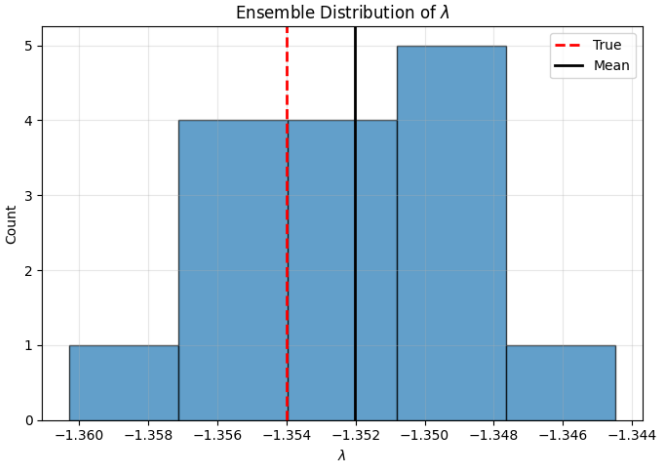

Figure 8: Ensemble distribution of the inferred mixed-convection parameter λ across 15 independent I-PINN runs. The true value (red dashed line) lies within the ensemble spread, and the sample mean (black line) demonstrates unbiased recovery.

A crucial measure of success is the reconstruction of the unmeasured velocity field. As shown in Figure 10, the I-PINN accurately recovers the velocity profile $f'(\eta)$ despite having *no* velocity data during training. This is achieved solely by enforcing the physics of the momentum equation, coupled through the learned parameters.

Furthermore, as a differentiable surrogate model, the trained network enables the extraction of wall metrics—specifically skin friction $f''(0)$ and the Nusselt number $-\theta'(0)$—via automatic differentiation. At the challenging condition ($\lambda = -1.354$, $M = -0.4$), these metrics exhibit the correct physical trends, such as reduced transport for negative M. Quantitative validation in well-posed regimes, including the Blasius limit, is

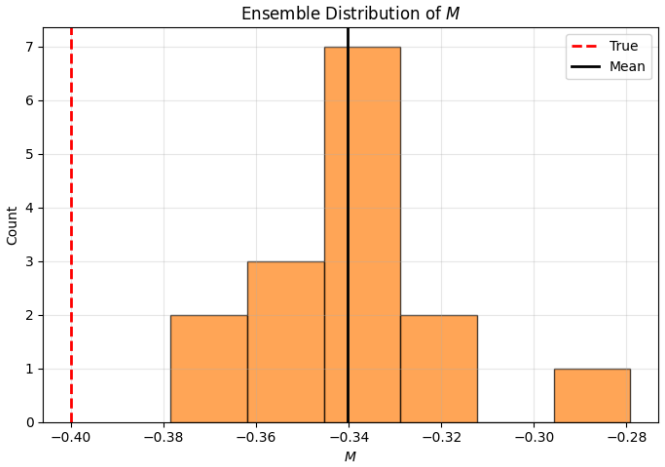

Figure 9: Ensemble distribution of the recovered melting parameter M. While the variance is small, the ensemble mean exhibits a systematic offset relative to the true value, reflecting the weaker identifiability of M from sparse temperature-only observations.

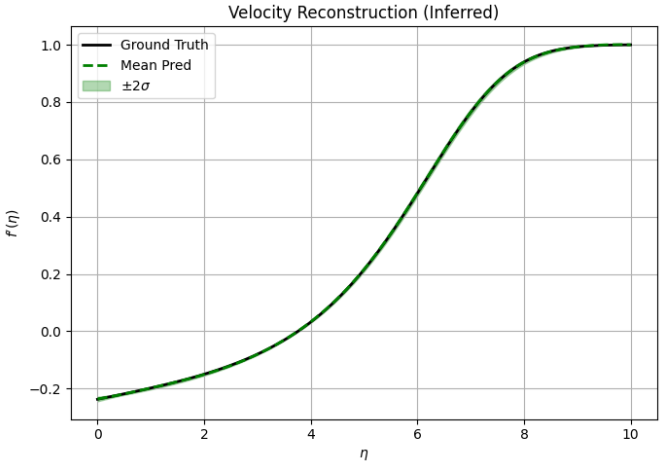

Figure 10: Reconstructed velocity profile $f'(\eta)$ from the I-PINN ensemble. The mean prediction closely matches the ground-truth solution, with uncertainty ($\pm 2\sigma$) remaining minimal across the full domain.

provided in Table 5, confirming the model's accuracy across the domain.

5. Conclusion and future directions

This work developed and validated an inverse Physics-Informed Neural Network (I-PINN) framework for parameter discovery in steady mixed convection with melting in a porous medium. By jointly inferring the unknown parameters (λ, M) and the associated similarity profiles from sparse and noisy temperature measurements, the method achieved high-fidelity recovery and forward-consistent field reconstructions, even in regimes characterised by stiffness and opposing-flow behaviour.

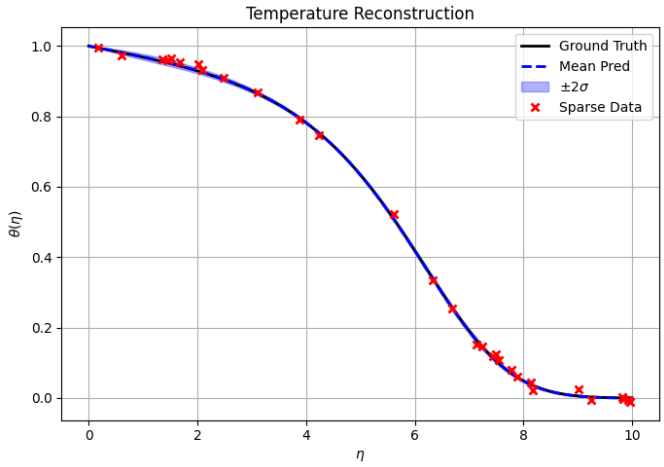

Figure 11: Reconstruction of the temperature field $\theta(\eta)$. Sparse noisy measurements (red markers) are accurately fitted by the I-PINN, and the ensemble mean prediction remains consistent with the ground-truth solution within the $\pm 2\sigma$ confidence band.

Quantitative assessments demonstrate the robustness of the approach compared to classical baselines. In the challenging opposing-flow case ($\lambda_{\text{true}} = -1.354, M_{\text{true}} = -0.4$), the I-PINN accurately recovered both parameters and reproduced the forward wall metrics skin friction $f''(0)$ and Nusselt number $-\theta'(0)$ in close agreement with benchmark solutions. Notably, the I-PINN demonstrated superior stability over the Levenberg–Marquardt algorithm, particularly in the presence of measurement noise, where the classical iterative solver exhibited sensitivity to initial guesses and gradient instability. These results were enabled by a stabilised training strategy combining a data-guided curriculum, self-adaptive loss weighting, and a two-stage Adam–L-BFGS optimiser.

Despite these strengths, certain limitations remain. The present framework relies on a valid similarity reduction and deterministic nanofluid property closures. Moreover, although the ensemble-based uncertainty quantification used in this study provides meaningful confidence intervals, it requires training multiple independent models, which increases the computational burden compared to single-run estimations.

To address these aspects, future work will focus on three key directions. First, we aim to extend the framework to Bayesian I-PINNs to enable full posterior inference, thereby capturing parameter correlations and reducing the overhead of ensemble training. Second, to bridge the gap between computational modelling and practical engineering, future efforts will validate the I-PINN against experimental datasets using multi-fidelity learning. This will involve integrating sparse experimental sensor data with high-fidelity simulations to handle real-world uncertainties. Finally, the framework will be extended to transient and full two-dimensional formulations to accommodate complex geometries and time-dependent melting phenomena where similarity reductions are no longer applicable.

Data availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

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", Journal of Computational Physics **378** (2019) 686. https://doi.org/10.1016/j.jcp.2018.10.045.

[2] J. D. Toscano, V. Oommen, A. J. Varghese, Z. Zou, N. A. Daryakenari, C. Wu & G. E. Karniadakis, "From PINNs to PIKANs: recent advances in physics-informed machine learning", Machine Learning for Computational Science and Engineering **1** (2025) 15. https://doi.org/10.1007/s44379-025-00015-1.

[3] A. Farea, O. Yli-Harja & F. Emmert-Streib, "Understanding physics-informed neural networks: techniques, applications, trends, and challenges", AI **5** (2024) 1534. https://doi.org/10.3390/ai5030074

[4] S. Ganga & Z. Uddin, "Exploring physics-informed neural networks: From fundamentals to applications in complex systems", arXiv preprint arXiv:2410.00422 (2024). https://arxiv.org/abs/2410.00422.

[5] D. Kim & J. Lee, "A review of physics informed neural networks for multiscale analysis and inverse problems", Multiscale Science and Engineering **6** (2024) 1. https://doi.org/10.1007/s42493-024-00106-w.

[6] M. Berardi, F. V. Difonzo & M. Icardi, "Inverse physics-informed neural networks for transport models in porous materials", Computer Methods in Applied Mechanics and Engineering **435** (2025) 117628. https://doi.org/10.1016/j.cma.2024.117628.

[7] Y. Xia & Y. Meng, "Physics-informed neural network (PINN) for solving frictional contact temperature and inversely evaluating relevant input parameters", Lubricants **12** (2024) 62. https://doi.org/10.3390/lubricants12020062.

[8] M. J. Colaço, H. R. B. Orlande & G. S. Dulikravich, "Inverse and optimization problems in heat transfer", Journal of the Brazilian Society of Mechanical Sciences and Engineering **28** (2006) 1. https://doi.org/10.1590/S1678-58782006000100001.

[9] P. Sajadi, M. R. Dehaghani, Y. Tang & G. G. Wang, "Physics-informed online learning for temperature prediction in metal AM", Materials **17** (2024) 3306. https://doi.org/10.3390/ma17133306.

[10] S. Szenási & I. Felde, "Database for research projects to solve the inverse heat conduction problem", Data **4** (2019) 90. https://doi.org/10.3390/data4030090.

[11] S. A. Jumana, M. G. Murtaza, M. Ferdows, O. D. Makinde & K. Zaimi, "Dual solutions analysis of melting phenomenon with mixed convection in a nanofluid flow and heat transfer past a permeable stretching/shrinking sheet", Journal of Nanofluids **9** (2020) 313. https://doi.org/10.1166/jon.2020.1761

[12] B. E. Madir, F. Luddens, C. Lothode & I. Danaila, "Physics informed neural networks for heat conduction with phase change", International Journal of Heat and Mass Transfer **252** (2025) 127430. https://doi.org/10.1016/j.ijheatmasstransfer.2025.127430.

[13] Q. Yuan, P. Yao, W. Zhao & B. Zhang, "Optimization of sparse sensor layouts and data-driven reconstruction methods for steady-state and transient thermal field inverse problems", Sensors **25** (2025) 4984. https://doi.org/10.3390/s25164984.

[14] L. Yang, X. Meng & G. E. Karniadakis, "B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data", Journal of Computational Physics **425** (2021) 109913. https://doi.org/10.1016/j.jcp.2020.109913.

[15] H. Barannia & M. Esmaeilpour, "On the application of physics informed neural networks (PINN) to solve boundary layer thermal–fluid problems", International Communications in Heat and Mass Transfer **132** (2022) 105890. https://doi.org/10.1016/j.icheatmasstransfer.2022.105890.

[16] K. Cong, G. Li, Y. Sun *et al.*, "Using physics-informed derivative networks to solve the forward problem of a free-convective boundary layer problem", Scientific Reports **15** (2025) 18766. https://doi.org/10.1038/s41598-025-03918-4.

[17] A. Raina, S. Badireddi & S. Natesan, "Application of PINN to obtain solution of boundary layer problems arising in fluid dynamics", Mathematical Foundations of Computing **10** (2026) 89. https://doi.org/10.3934/mfc.2025024.

[18] A. D. Jagtap, K. Kawaguchi & G. E. Karniadakis, "Adaptive activation functions accelerate convergence in deep and physics-informed neural networks", Journal of Computational Physics **404** (2020) 109136. https://doi.org/10.1016/j.jcp.2019.109136.

[19] S. Wang, Y. Teng & P. Perdikaris, "When and why PINNs fail to solve PDEs: a gradient pathology perspective", Journal of Computational Physics **449** (2022) 110768. https://doi.org/10.1016/j.jcp.2021.110768.

[20] S. Ahmad & I. Pop, "Melting effect on mixed convection boundary layer flow about a vertical surface embedded in a porous medium: Opposing flows case", Transport in Porous Media **102** (2014) 317. https://doi.org/10.1007/s11242-014-0291-x.

[21] P. Sharma, W. T. Chung, B. Akoush & M. Ihme, "A review of physics-informed machine learning in fluid mechanics", Energies **16** (2023) 2343. https://doi.org/10.3390/en16052343.

[22] Z. Haddad, H. F. Oztop, E. Abu-Nada & A. Mataoui, "A review on natural convective heat transfer of nanofluids", Renewable and Sustainable Energy Reviews **16** (2012) 5363. https://doi.org/10.1016/j.rser.2012.04.003.

[23] T. Hayat, S. Naz, A. Alsaedi & S. Momani, "Melting and dissipative effects about entropy induced Darcy–Forchheimer flow involving ternary-hybrid nanofluids", Case Studies in Thermal Engineering **55** (2024) 104097. https://doi.org/10.1016/j.csite.2024.104097.