Engineering Applications of Artificial Intelligence 179 (2026) 115179

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Research paper

Boundary Region Reinforcement Physics-Informed Neural Networks for solving partial differential equations

Shengtai Yao [a,b,c], Weifeng Huang [a,b], Yang Hu [d], Qiang He [a,b,*]

[a] Department of Mechanical Engineering, Tsinghua University, Beijing, 100084, China
[b] State Key Laboratory of Tribology in Advanced Equipment, Tsinghua University, Beijing, 100084, China
[c] Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, 21218, MD, United States
[d] School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing, 100044, China

ARTICLE INFO

Keywords:
Partial differential equations
Physics-Informed Neural Networks
Boundary Region Reinforcement
Computational accuracy
Deep learning
Thermo-elastic coupling system

ABSTRACT

Physics-Informed Neural Networks (PINNs) have emerged as a deep learning framework for solving partial differential equations (PDEs). However, standard approaches often struggle to strictly enforce boundary conditions, limiting their computational accuracy, especially in complex multiphysics coupled systems. To address this challenge, we introduce Boundary Region Reinforcement Physics-Informed Neural Networks (BRR-PINNs), a novel framework designed to strengthen boundary constraint enforcement and significantly improve solution accuracy. The method converts non-Dirichlet conditions into exact Dirichlet boundaries using intermediate variables to reduce redundant loss terms, and applies a spatial reinforcement function to prioritize training near boundaries. We validate the effectiveness of BRR-PINNs on a two-dimensional axisymmetric thermo-elastic coupling problem, involving a circular ring subjected to both thermal and mechanical loads. Experimental results show that our method consistently outperforms several existing approaches, achieving approximately one order of magnitude improvement in accuracy.

1. Introduction

With the continuous advancement of deep learning, its extensive applications in areas such as image recognition, semantic understanding, and autonomous driving rely largely on models trained with vast amounts of data. However, in fields such as engineering applications, there are some research problems in which obtaining a large amount of data is infeasible or prohibitively costly. Raissi et al. (2019) proposed Physics-Informed Neural Networks (PINNs), which integrate the physical laws and experimental data into the loss function of the neural network. The entire training process involves minimizing the loss function, ensuring that the resulting model satisfies the given equations and conditions. The advantage is that it requires a small amount of, or even no, labeled experimental data to obtain solutions to partial differential equations (PDEs), due to the incorporation of physical laws into the network. The core of PINNs computational capability lies in automatic differentiation (AD) (Baydin et al., 2018), which enables it to incorporate physical equations into the loss function. AD can be conveniently accessed in commonly used frameworks such as PyTorch and TensorFlow. Raissi et al. (2019) proposed two types of problem: the data-driven solution and the data-driven discovery of partial differential equations. They validated the effectiveness of the PINNs method on equations involving the Schrödinger equation, Allen-Cahn equation, and Navier–Stokes equations. Since it has been proposed, PINNs have been applied in various fields, including heat transfer (Zobeiry and Humfeld, 2021; Cai et al., 2021) chemical kinetics (Ji et al., 2021; De Florio et al., 2022), fluid dynamics (Jin et al., 2021; Almajid and Abu-Al-Saud, 2022), solid mechanics (Haghighat et al., 2021; Bai et al., 2023; Le-Duc et al., 2024), systems biology (Yazdani et al., 2020), geophysics (Kadeethum et al., 2020) etc.

After the proposal of PINNs, various variants have been proposed to enhance the model performance and extend its application to different problems. These include B-PINNs (Yang et al., 2021), based on a Bayesian framework to mitigate over fitting and achieve higher prediction accuracy in noisy data; CAN-PINN (Chiu et al., 2022), which couple AD with numerical differentiation (ND) to obtain greater robustness and efficiency; g-PINNs (Yu et al., 2022), a gradient-enhanced method, improves the smoothness of the obtained solution by adding derivative terms to the loss function, thus enhancing the precision; cPINN (Jagtap et al., 2020), designed to solve nonlinear conservation laws in discrete domains, can efficiently lend itself to parallelized computation, etc. The

* Corresponding author.
E-mail addresses: syao31@jh.edu (S. Yao), huangwf@mail.tsinghua.edu.cn (W. Huang), yanghu@bjtu.edu.cn (Y. Hu), heqiang@tsinghua.mail.edu.cn (Q. He).

https://doi.org/10.1016/j.engappai.2026.115179
Received 8 October 2024; Received in revised form 22 April 2026; Accepted 19 May 2026
Available online 30 May 2026
0952-1976/© 2026 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

Algorithm 1: Boundary Region Reinforcement Physics-Informed Neural Networks (BRR-PINNs)

Data: Collocation points from the solution domain: $\boldsymbol{x}$

Step 1: Formulate the governing PDEs and identify the *primitive output variables* $\boldsymbol{u}_p$.

Step 2: Handle boundary conditions as follows:

  for each primitive output variable u_p^i in $\boldsymbol{u}_p$ **do**

    for each Dirichlet boundary condition for u_p^i **do**

      Compute the distance function ϕ_j;

    end

    Calculate the approximate distance function (ADF): $\phi(\boldsymbol{x}) = \left(\sum_{j=1}^{k} \frac{1}{\phi_j^m} \right)^{-\frac{1}{m}}$;

    Give a particular solution u_{par}^i for u_p^i;

    Impose Dirichlet boundary conditions for u_p^i: $u_p^i = u_{\text{par}}^i + \phi\, u_p^i$;

  end

  Define the set of *intermediate output variables* $\boldsymbol{q}$ for non-Dirichlet boundary conditions;

  Construct the corresponding set of *connecting equations* f_{cnc}.

  for each intermediate output variable q_i in $\boldsymbol{q}$ **do**

    Impose Dirichlet boundary conditions for q_i using the "hard" method;

  end

Step 3: Reinforce the boundary region

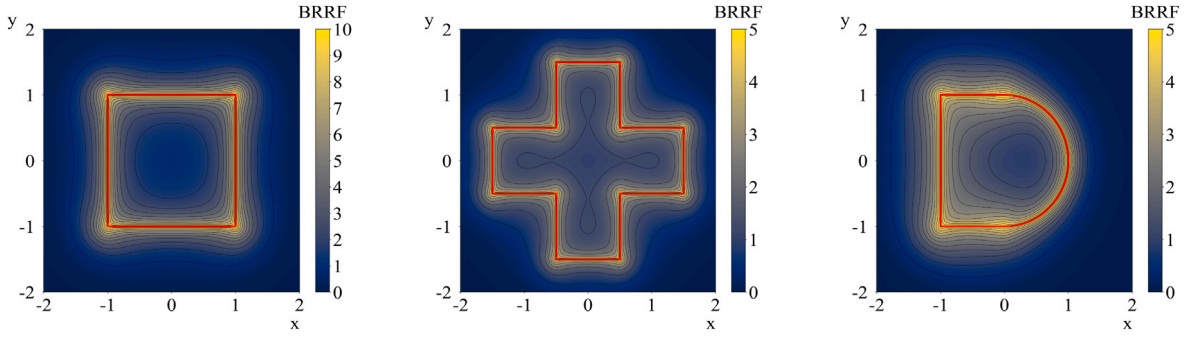
$$\mathcal{L}_{\text{cnc}}^i = \omega_{\text{cnc}}^i \ \text{mean}\left(\ (f_{\text{cnc}}^i(\boldsymbol{x}) \cdot \text{BRRF}(\boldsymbol{x}))^2 \ \right),$$

Compute the total loss function: $\mathcal{L} = \sum \mathcal{L}_{\text{PDE}}^j + \sum \mathcal{L}_{\text{cnc}}^i$;

Train the network parameters.

return θ

Result: Trained network parameters θ (i.e., the solution)



(a) The BRRF distribution of a convex polygon domain (square), reinforcement coefficient $\beta = 10$

(b) The BRRF distribution of a concave polygon domain, reinforcement coefficient $\beta = 5$

(c) The BRRF function distribution over a domain with curved boundaries, $\beta = 5$

Fig. 2. In two-dimensional problems, the BRRF distribution over various-shaped domains with red lines indicating the boundaries, including convex polygon domain, concave polygon domain and domain with curved boundaries.

2. Prioritize interior points near the boundaries by reinforcing the boundary region, leading to more accurate satisfaction of the boundary conditions.

As shown in Table 1, we compare the proposed BRR-PINNs with several existing approaches, including conventional PINNs, the "hard" method, and g-PINNs, in terms of boundary enforcement, the number of loss terms. Suppose that the number of governing PDEs is M_p; the total number of boundary conditions is M_b, among which M_d are Dirichlet-type constraints. Let M_c denote the number of distinct connecting equations. Clearly, $M_c \leq M_b - M_d$. Let N denote the number of sample points. Regarding boundary enforcement, the proposed BRR-PINNs incorporate intermediate-variable transformation and Dirichlet imposition (IVD), together with boundary region reinforcement (BRR). Moreover, the additional computational overhead introduced by BRR per training epoch is $\mathcal{O}(M_c N)$.

Table 1
Comparison of the boundary enforcement and the number of loss terms of different methods.

Method	Boundary enforcement	# Loss terms
Conventional PINNs	"Soft" enforce	$M_p + M_b$
"Hard" method PINNs	"Hard" impose (Dirichlet-type)	$M_p + M_b - M_d$
g-PINNs	"Soft" enforce	$2M_p + M_b$
BRR-PINNs	**IVD+BRR**	$\boldsymbol{M_p + M_c}$

3. Numerical experiment

This section uses a two-dimensional axisymmetric thermo-elastic coupling system of a circular ring as an example to demonstrate the capabilities of BRR-PINNs. The problem is derived from the sealing ring in a non-contact mechanical seal, with boundary conditions that closely replicate practical operating environments, highlighting the method's applicability to real-world engineering problems. This problem can be

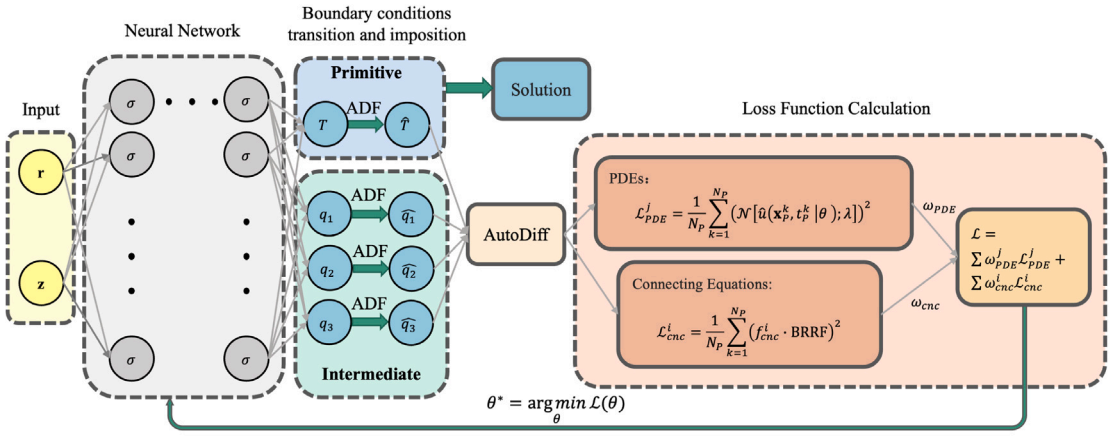


Fig. 3. BRR-PINNs computational framework.

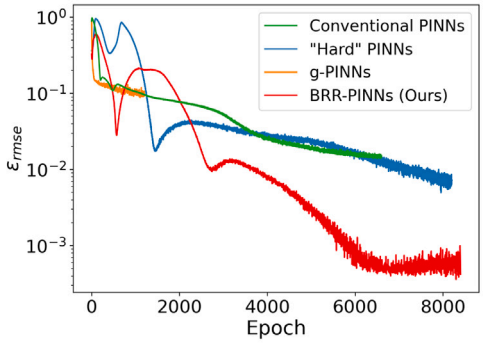


Fig. 4. Accuracy comparison of different methods on heat transfer problem.

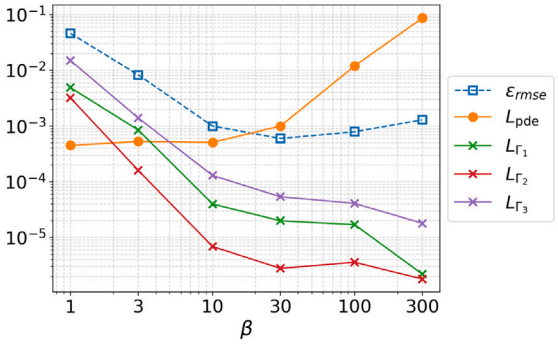


Fig. 5. The effect of boundary reinforcement coefficient β in the heat transfer problem.

divided into three parts: heat transfer, elastic deformation, and thermo-elastic coupling deformation. The detailed description and setup can be found in Appendix A. In this section, we will compare the proposed BRR-PINNs method with several existing methods, including the conventional PINNs, "Hard" method PINNs, and g-PINNs. To evaluate the computational result, a L_2 evaluation metric, as shown in (17),

$$\varepsilon_{rmse} = \sqrt{\frac{\text{mean}(|\text{ Predict} - \text{Exact }|^2)}{\text{mean}(|\text{ Exact }|^2)}}, \tag{17}$$

where, "Predict" is the result of PINNs, and "Exact" is the result of FEM. They are both vectors composed of the outputs at the collocation points in the solution domain.

For each method, we follow the hyperparameter-tuning procedure and the stopping criteria described in Appendix C.1. The details of the tuning process and the parameters selected for each method will be provided in Appendix C.2.

3.1. Heat transfer problem

The heat transfer problem that we considered is described in detail in Appendix A.1. We compare the memory usage, computational time per epoch, floating point operations (FLOPs) per epoch, and the computational accuracy of the proposed BRR-PINNs with several existing methods in Table 2, using the experimental setup in Table C.6. As shown in the results, BRR-PINNs require memory usage and computational time comparable to conventional PINNs and "hard" methods, while clearly outperforming g-PINNs, whose higher-order derivatives lead to substantially higher computational cost. More importantly, BRR-PINNs achieve a computational accuracy on the order of $\mathcal{O}(10^{-4})$, performing an improvement of approximately one order of magnitude over the other methods.

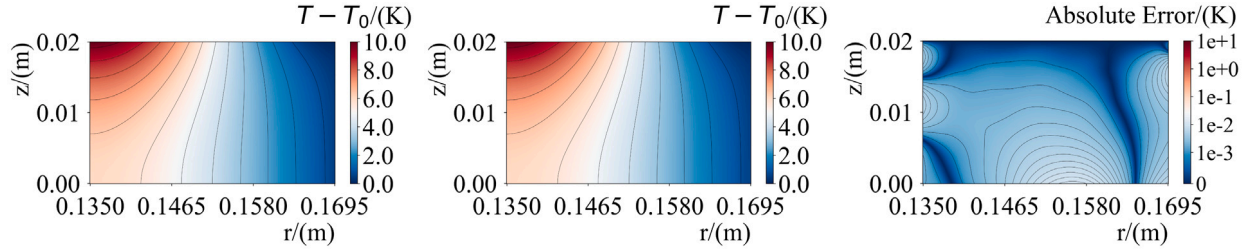
As shown in Fig. 4, we present the accuracy of the different methods throughout the training process. We can clearly observe the superiority of our proposed method, BRR-PINNs, which achieves an accuracy of the order of $\mathcal{O}(10^{-4})$. The resulting temperature-rise distributions from BRR-PINNs and the absolute error with the FEM result can be found in Fig. 6. More details of the hyperparameter tuning process, selected parameters, the training process, and the resulting temperature-rise distributions for each method are provided in Appendix C.2.1.

We further analyze the effect of the boundary reinforcement coefficient β on each loss term and the overall accuracy. As shown in Fig. 5, increasing β leads to a clear decreasing trend in all loss of boundary conditions, again demonstrating that our algorithm effectively enforces the boundary conditions. However, as β increases, the loss terms of the governing PDEs get less priority during training, resulting in an observable increase in residuals. This reveals a trade-off between boundary satisfaction and the governing equations. Consequently, the overall error ε_{rmse} first decreases and then increases as β increases. The choice of β should therefore balance these two factors: it should be large enough to significantly reduce the boundary losses, yet not so large that it severely affects the PDEs residuals.

Moreover, when $\beta = 1$, the model simply incorporates the intermediate-variable transformation and Dirichlet imposition without applying boundary region reinforcement. This serves as an ablation case for BRR-PINNs and highlights that the boundary region reinforcement is the core component of the proposed architecture.

Table 2
Comparison of different methods on the heat transfer problem.

Method	GPU memory	Time/epoch	FLOPs/epoch	ε_{rmse}	Stop epoch
Conventional PINNs	1.3 GB	0.46 s	2.797e10	1.5e−2	6600
"Hard" method PINNs	1.3 GB	0.56 s	2.801e10	7.1e−3	8200
g-PINNs	6.4 GB	1.33 s	1.113e11	9.9e−2	1200
BRR-PINNs (Ours)	**1.6 GB**	**0.72 s**	**3.428e10**	**6.0e−4**	**8400**



(a) FEM (b) BRR-PINNs (c) Absolute Error

Fig. 6. The result from BRR-PINNs.

Table 3
Comparison of different methods on elastic deformation problem.

Method	GPU memory	Time/epoch	FLOPs/epoch	$\varepsilon_{rmse}(u_r)$	$\varepsilon_{rmse}(u_z)$	Stop epoch
Conventional PINNs	12.13 GB	0.82 s	3.460e11	3.9e−3	1.1e−2	118k
"Hard" method PINNs	12.49 GB	0.88 s	3.464e11	3.2e−3	6.0e−3	160k
BRR-PINNs (Ours)	**13.59 GB**	**1.04 s**	**3.831e11**	**1.1e−4**	**1.7e−3**	**161k**

Additionally, we ran BRR-PINNs ten times, and the results were stable. The mean accuracy, measured by ε_{rmse}, is 6.5e−4, and the standard deviation is 4.3e−5.

3.2. Elastic deformation problem

We then consider the elastic deformation problem, described in detail in Appendix A.2. We compare our BRR-PINNs with conventional PINNs and the "Hard" method PINNs. Note that g-PINNs require a significantly larger computational graph for this problem and consequently require far more GPUs memory (larger than 40 GB), beyond what is typically available on standard GPUs. Therefore, we do not include g-PINNs in our comparison.

Using the experimental setup in Table C.8, together with the hyperparameter tuning strategy and stopping criteria described in Appendix C.1, the comparison results are presented in Table 3. It is evident that our proposed BRR-PINNs outperform other methods, achieving approximately one order of magnitude improvement in accuracy without increasing too much training complexity, including GPU memory usage or training time per epoch. The accuracy curves during training for displacement in the r-direction and z-direction are shown in Figs. 7 and 8, respectively. The predicted displacement maps obtained using the BRR-PINNs, along with the corresponding absolute error distributions, are shown in Fig. 9. Additional details including the selected parameters and visualizations of the results from other methods are provided in Appendix C.2.2.

Similarly to the heat transfer problem, the boundary coefficient β exhibits a parallel behavior in the elastic deformation case. As shown in Figs. 10 and 11, increasing β leads to higher loss residuals of governing PDEs while significantly reducing boundary condition errors. The overall computational errors first decrease and then increase as β grows, indicating the trade-off involved in selecting β.

3.3. Thermo-elastic deformation problem

Finally, we consider the thermo-elastic coupling deformation problem, described in detail in Appendix A.3. The comparison results are presented in Table 4 and Figs. 12 and 13. The predicted displacement maps obtained using the BRR-PINNs are shown in Fig. 14. More details can be found in Appendix C.2.3. It is clear that our proposed method is capable of accurately solving multiphysics coupling systems and performs much better.

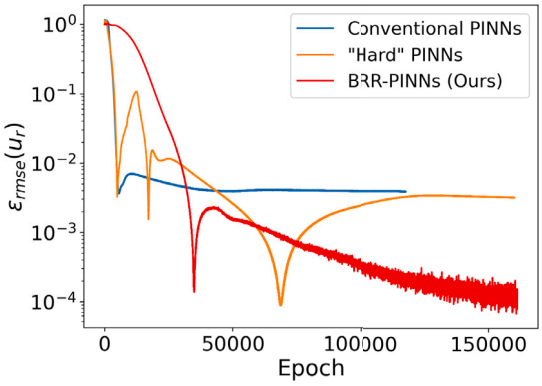


Fig. 7. Accuracy comparison of different methods of z-direction displacement on elastic deformation problem.

4. Discussion and future work

In summary, we proposed a novel framework, BRR-PINNs, and demonstrated its ability to significantly enhance computational accuracy across three representative problems: the heat transfer problem, the elastic deformation problem, and the thermo-elastic deformation problem. Compared to several existing approaches, including conventional PINNs, "hard" method PINNs, and g-PINNs, our method achieves an improvement of approximately one order of magnitude in precision.

Table 4
Comparison of different methods on thermo-elastic deformation problem.

Method	GPU memory	Time/epoch	FLOPs/epoch	$\varepsilon_{rmse}(u_r)$	$\varepsilon_{rmse}(u_z)$	Stop epoch
Conventional PINNs	14.38 GB	0.94 s	4.120e11	2.7e−3	5.3e−2	95k
"Hard" method PINNs	14.55 GB	1.00 s	4.124e11	1.4e−3	1.2e−2	69k
BRR-PINNs (Ours)	**14.26 GB**	**1.17 s**	**4.701e11**	**1.2e−4**	**1.4e−3**	**102k**

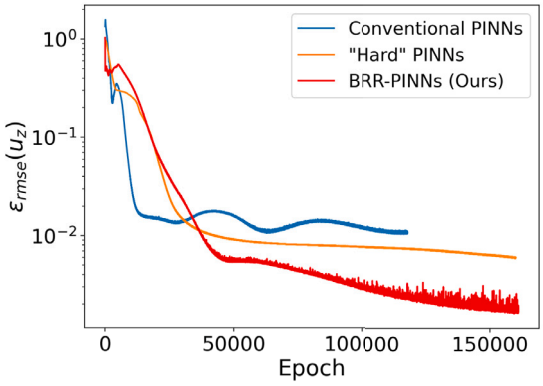


Fig. 8. Accuracy comparison of different methods of r-direction displacement on elastic deformation problem.

Table A.5
Physical parameters for the two-dimensional axisymmetric circular ring.

Physical parameters	
Inner diameter r_1	0.135 m
External diameter r_2	0.1695 m
Height h_1	0.02 m
Poisson ratio μ	0.14
Convective heat transfer coefficient h	2.06e4 W/(m^2 K)
Heat conductivity coefficient k	60 W/(m K)
External temperature T_0	303.15 K
Maximum temperature rise T_m	10 K
Elasticity modulus E	420 GPa
Low pressure P_l	1 MPa
High pressure P_h	10 MPa
Linear thermal expansion coefficient	4e−6 K^{-1}

Furthermore, we conducted a detailed analysis of the boundary reinforcement coefficient and highlighted the trade-off between reducing the governing PDE residuals and the satisfaction of the boundary conditions.

Intuitively, the failure of conventional PINNs can be attributed to the fact that, in the early stages of training, the model lacks sufficient guidance on where to converge and may become trapped in local minima, ultimately failing to satisfy all governing equations and boundary conditions. Our method mitigates this issue by introducing a framework that prioritizes boundary regions over the interior domain. Experimental results demonstrate that this strategy effectively guides the model toward more accurate solutions. Moreover, the framework naturally extends to complex geometries, as illustrated in Fig. 2. These observations indicate that BRR-PINNs hold strong potential for addressing more challenging systems, including complex geometries, higher-dimensional PDEs, and nonlinear PDEs. However, further validation of BRR-PINNs on these sophisticated problems remains an important direction for future work.

Our future work will proceed in several directions. First, as mentioned above, we aim to validate the proposed framework on more complex and challenging problems. Second, we plan to develop a more rigorous mathematical explanation for the phenomena observed in this study and further improve the efficiency and accuracy of the framework. Finally, the current implementation still relies on empirically tuned parameters. An additional research direction is to incorporate multi-task learning techniques or other self-adjusting weighting strategies into the framework, while ensuring the model performance.

CRediT authorship contribution statement

Shengtai Yao: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Data curation. **Weifeng Huang:** Writing – review & editing, Validation, Project administration. **Yang Hu:** Writing – review & editing, Investigation, Data curation. **Qiang He:** Writing – review & editing, Supervision, Project administration, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 52575218, 52275201, U2541266) and the Beijing Outstanding Young Scientist Program, China (JWZQ20240102006).

Appendix A. Problem setup

In this paper, we consider the heat transfer and deformation problem of a two-dimensional axisymmetric circular ring, with symmetry in the angular direction θ. The physical parameters of the ring are listed in Table A.5.

A.1. Heat transfer problem

The heat transfer equation in cylindrical coordinates (r, z, θ) is formulated as shown in (A.1),

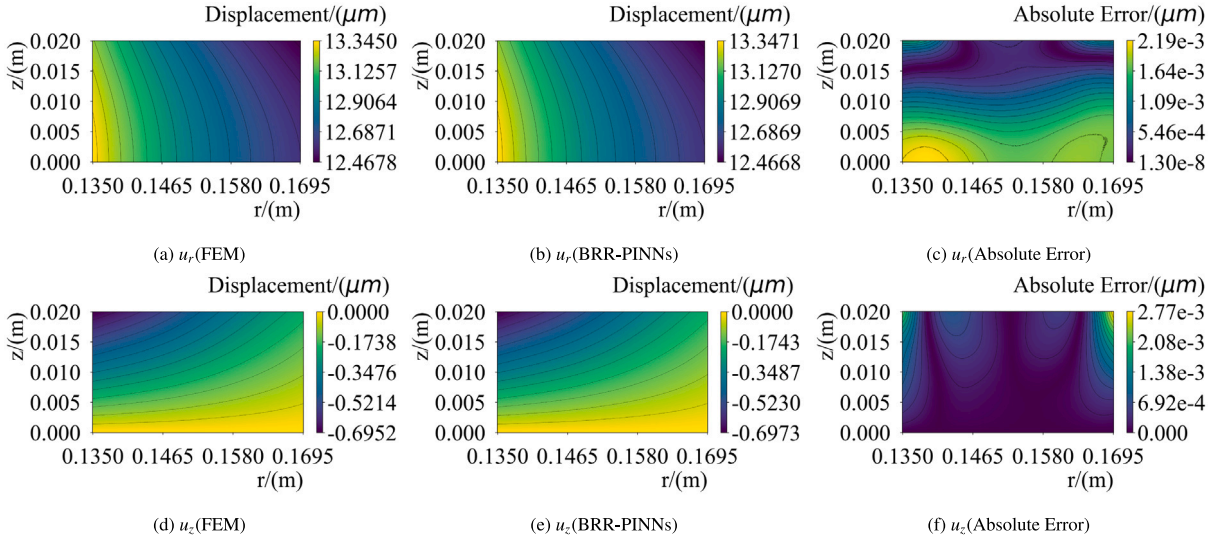
$$\frac{\partial T}{\partial t} = k\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 T}{\partial \theta^2} + \frac{\partial^2 T}{\partial z^2}\right), \tag{A.1}$$

where, T represents temperature; t denotes the time variable; and k is the thermal conductivity coefficient. With the inclusion of steady-state and symmetry conditions, the heat transfer equation is formulated as shown in (A.2)

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right) + \frac{\partial^2 T}{\partial z^2} = 0. \tag{A.2}$$

The boundary conditions for the heat transfer problem are shown in Fig. A.15. The shaded area represents a cross-section of the circular ring, which is exposed to an environment at a temperature of T_0. The inner diameter boundary Γ_1 and the bottom boundary Γ_2 are adiabatic boundaries, the outer diameter boundary Γ_3 is subject to convective heat transfer and the upper boundary Γ_4, an isothermal boundary, is maintained at a constant temperature. The boundary conditions for these four boundaries are given by (A.3)–(A.6),

$$\frac{\partial T}{\partial r} = 0, \ (r, z) \in \Gamma_1 \tag{A.3}$$



(a) u_r(FEM) (b) u_r(BRR-PINNs) (c) u_r(Absolute Error)

(d) u_z(FEM) (e) u_z(BRR-PINNs) (f) u_z(Absolute Error)

Fig. 9. FEM solution and BRR-PINNs computed result for the elastic deformation problem (displacement field $[u_r, u_z]$), along with the absolute error between them.

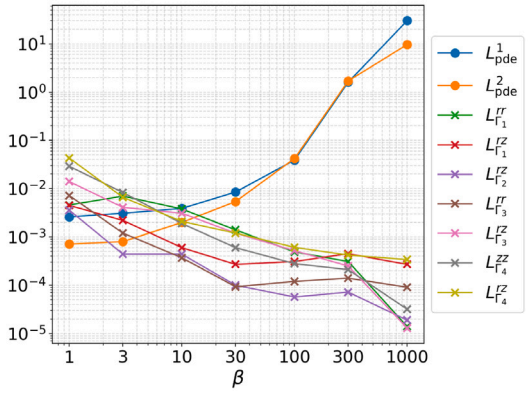


Fig. 10. The effect of boundary reinforcement coefficient β on loss terms in the elastic deformation problem.

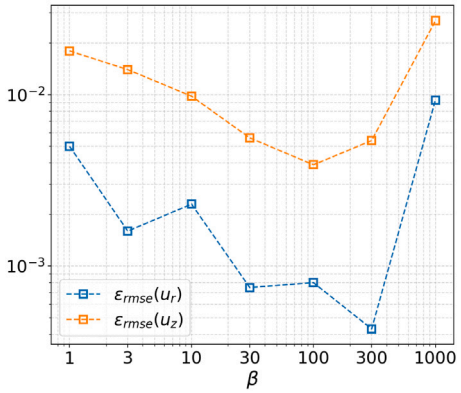


Fig. 11. The effect of boundary reinforcement coefficient β on accuracy in the elastic deformation problem.

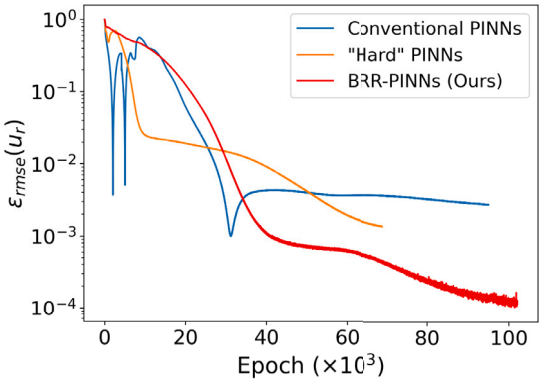


Fig. 12. Accuracy comparison of different methods of r-direction displacement on thermo-elastic deformation problem.

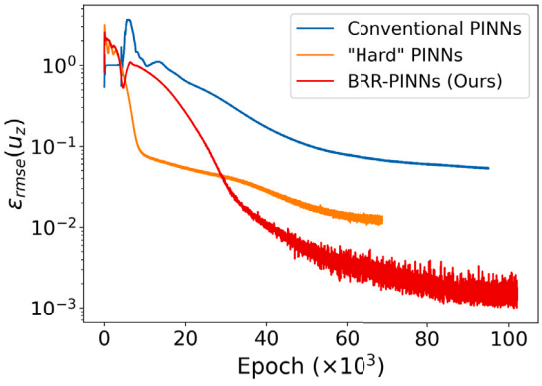


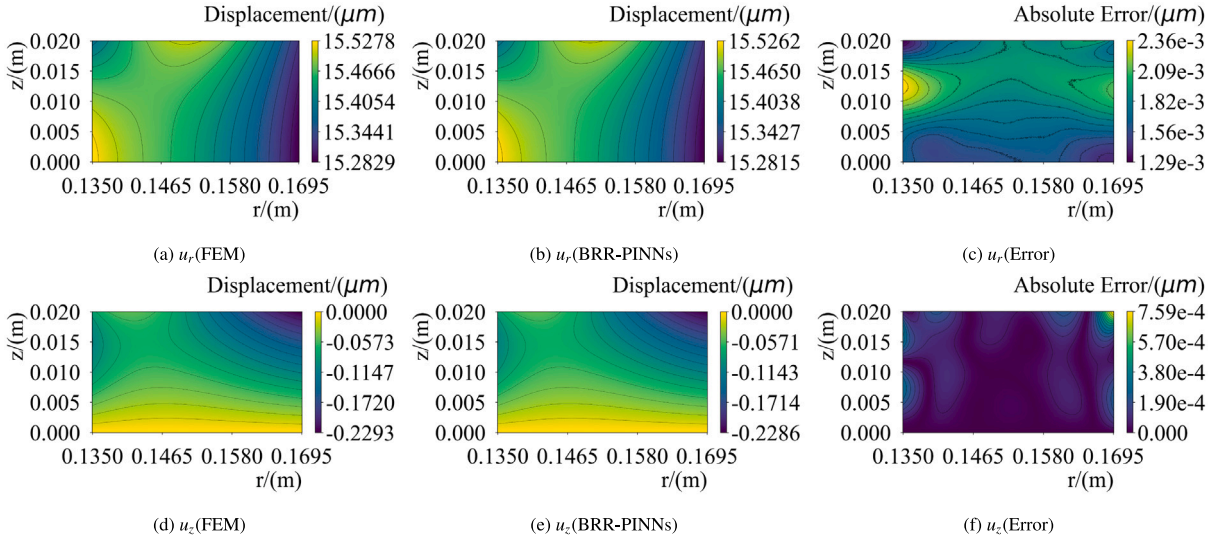
Fig. 13. Accuracy comparison of different methods of z-direction displacement on thermo-elastic deformation problem.

$$\frac{\partial T}{\partial z} = 0, \ (r, z) \in \Gamma_2 \tag{A.4}$$

$$h\left(T - T_0\right) + k\frac{\partial T}{\partial r} = 0, \ (r, z) \in \Gamma_3 \tag{A.5}$$

$$T - T_0 - T_{\text{iso}}(r) = 0, \ (r, z) \in \Gamma_4 \tag{A.6}$$

where, h is the coefficient of convective heat transfer, k is the thermal conductivity coefficient, and T_{iso} is the distribution of constant



(a) u_r(FEM) (b) u_r(BRR-PINNs) (c) u_r(Error)

(d) u_z(FEM) (e) u_z(BRR-PINNs) (f) u_z(Error)

Fig. 14. FEM solution and BRR-PINNs computed result for the thermo-elastic coupling deformation problem (displacement field $[u_r, u_z]$), along with the absolute error between them.

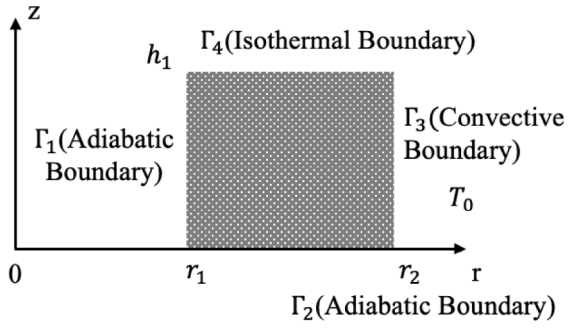


Fig. A.15. The boundary conditions of the heat transfer problem.

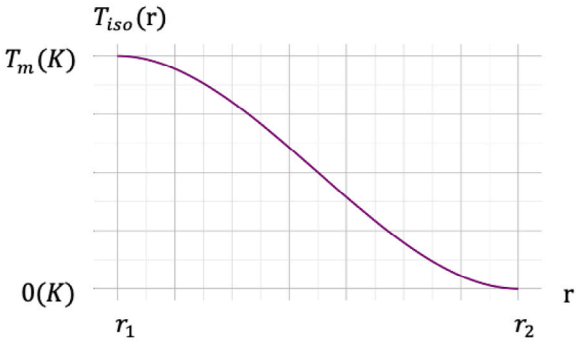


Fig. A.16. The temperature rise distribution of the isothermal boundary.

temperature rise on the boundary Γ_4, and is shown in Fig. A.16. In this paper, T_{iso} is constructed by a function such that it satisfies the boundary conditions of the inner and outer diameters simultaneously at both ends. The construction method is shown in (A.7),

$$T_{\text{iso}}(r) = T_m\left(n\left(\frac{r-r_1}{r_2-r_1}\right)^{n+1} - (n+1)\left(\frac{r-r_1}{r_2-r_1}\right)^{n} + 1\right), \tag{A.7}$$

where, T_m is the maximum temperature rise; order n is used to control the shape of the curve. In this work, we choose $T_m = 10K$ and $n = 2$. Other physical parameters can be found in Table A.5.

A.2. Elastic deformation problem

Next, we consider the elastic deformation for the two-dimensional axisymmetric circular ring. The problem exhibits symmetry in the angular direction, with shear stresses in the angular direction being zero, i.e., $\sigma_{\theta r} = \sigma_{\theta z} = \sigma_{r\theta} = \sigma_{z\theta} = 0$. The equilibrium equations for the infinitesimal element in cylindrical coordinates (r, z) are given by (A.8)–(A.9),

$$r\frac{\partial \sigma_{rr}}{\partial r} + r\frac{\partial \sigma_{zr}}{\partial z} + \sigma_{rr} - \sigma_{\theta\theta} = 0, \tag{A.8}$$

$$r\frac{\partial \sigma_{zz}}{\partial z} + r\frac{\partial \sigma_{rz}}{\partial r} + \sigma_{rz} = 0, \tag{A.9}$$

where, according to the reciprocal theorem of shear stress, $\sigma_{rz} = \sigma_{zr}$. The relationship between the stress tensor and displacements for linear elastic isotropic materials is given by (A.10)–(A.13),

$$\sigma_{rr} = 2G\left[\frac{1-\mu}{1-2\mu}\frac{\partial u_r}{\partial r} + \frac{\mu}{1-2\mu}\left(\frac{u_r}{r} + \frac{\partial u_z}{\partial z}\right)\right], \tag{A.10}$$

$$\sigma_{\theta\theta} = 2G\left[\frac{1-\mu}{1-2\mu}\frac{u_r}{r} + \frac{\mu}{1-2\mu}\left(\frac{\partial u_z}{\partial z} + \frac{\partial u_r}{\partial r}\right)\right], \tag{A.11}$$

$$\sigma_{zz} = 2G\left[\frac{1-\mu}{1-2\mu}\frac{\partial u_z}{\partial z} + \frac{\mu}{1-2\mu}\left(\frac{\partial u_r}{\partial r} + \frac{u_r}{r}\right)\right], \tag{A.12}$$

$$\sigma_{rz} = G\left(\frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z}\right), \tag{A.13}$$

where, u_r represents the displacement in r direction; u_z is the displacement in z direction; μ is the Poisson's ratio of the material; G is the shear modulus, given by $G = \frac{E}{2(1+\mu)}$, where E is the Young's modulus of the material. These parameters value can be found in Table A.5. Eqs. (A.8)–(A.13) can fully describe the elastic deformation, and the output variables include displacement field $[u_r, u_z]^T$ and stress tensor $[\sigma_{rr}, \sigma_{\theta\theta}, \sigma_{zz}, \sigma_{rz}]^T$. The boundary conditions for elastic deformation problem are shown in Fig. A.17. Boundaries Γ_1, Γ_3 and Γ_4 are imposed pressure perpendicular to their surfaces. Boundary Γ_1 has high pressure, $P_h = 10$ MPa; boundary Γ_3 has low pressure, $P_l = 1$ MPa; and boundary Γ_4 has a varying pressure distribution as shown in Fig. A.18. Since the pressure acts perpendicularly to the surface, there are no components along the surface direction. Boundary Γ_2 is imposed a displacement constraint in z-direction, and similarly, there are no stress components along the surface. On these four boundaries, the stress component $\sigma_{rz} = 0$. The boundary condition equations for the four boundaries are shown in (A.14)–(A.17),

$$\sigma_{rr} = P_h;\ \sigma_{rz} = 0,\quad (r, z) \in \Gamma_1, \tag{A.14}$$

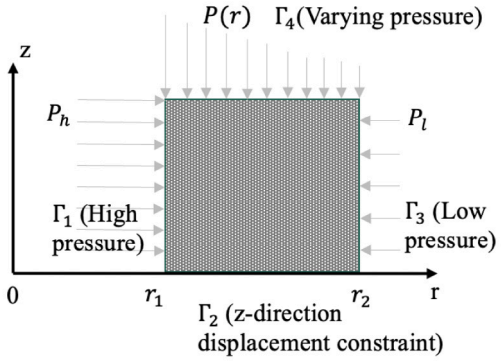


Fig. A.17. The boundary conditions of the elastic deformation problem.

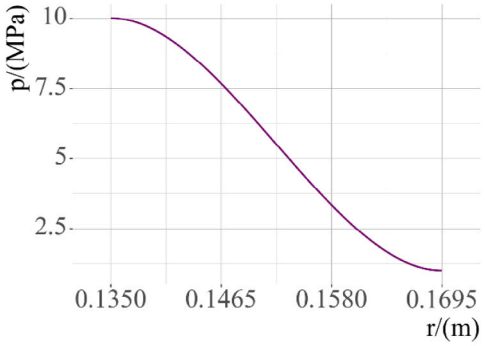


Fig. A.18. The varying pressure distribution for the Γ_4 boundary.

$$u_z = 0;\ \sigma_{rz} = 0, \quad (r,z) \in \Gamma_2, \tag{A.15}$$

$$\sigma_{rr} = P_l;\ \sigma_{rz} = 0, \quad (r,z) \in \Gamma_3, \tag{A.16}$$

$$\sigma_{zz} = P(r);\ \sigma_{rz} = 0, \quad (r,z) \in \Gamma_4. \tag{A.17}$$

A.3. Thermo-elastic deformation problem

The thermo-elastic deformation equations for a two-dimensional axisymmetric circular ring are analogous to the elastic deformation equations. The equilibrium equations are same as those of elastic deformation, as shown in (A.8)–(A.9). The expression for the stress tensor needs to incorporate the thermal expansion effect, as illustrated in (A.18)–(A.21),

$$\sigma_{rr} = 2G\left[\frac{1-\mu}{1-2\mu}\frac{\partial u_r}{\partial r} + \frac{\mu}{1-2\mu}\left(\frac{u_r}{r} + \frac{\partial u_z}{\partial z}\right)\right] - \gamma\Delta T, \tag{A.18}$$

$$\sigma_{\theta\theta} = 2G\left[\frac{1-\mu}{1-2\mu}\frac{u_r}{r} + \frac{\mu}{1-2\mu}\left(\frac{\partial u_z}{\partial z} + \frac{\partial u_r}{\partial r}\right)\right] - \gamma\Delta T, \tag{A.19}$$

$$\sigma_{zz} = 2G\left[\frac{1-\mu}{1-2\mu}\frac{\partial u_z}{\partial z} + \frac{\mu}{1-2\mu}\left(\frac{\partial u_r}{\partial r} + \frac{u_r}{r}\right)\right] - \gamma\Delta T, \tag{A.20}$$

$$\sigma_{rz} = G\left(\frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z}\right), \tag{A.21}$$

where, γ is the thermal stress coefficient, $\gamma = \frac{\alpha E}{1-2\mu}$, where α is the linear thermal expansion coefficient; ΔT is the temperature rise. It should be noted that ΔT is not a constant; it varies with coordinates (r,z). Therefore, when substituting it into the equilibrium equation, $\frac{\partial \Delta T}{\partial r} \neq 0$. The boundary conditions imposed for thermo-elastic coupling deformation problem are totally the same as elastic deformation, shown in Appendix A.2.

Appendix B. BRR-PINNs implementation details

In this section, we provide details on how to implement the BRR-PINNs method, using the example problems defined in Appendix A.

B.1. Heat transfer problem implementation

The first step of the BRR-PINNs algorithm is to formulate the governing PDEs and determine the primitive output variables. As shown in (A.2), there is only one equation and the single primitive variable is the temperature T.

Step 2 addresses the boundary conditions given in (A.3)–(A.6). Among them, only (A.6) is a Dirichlet condition, which can be directly enforced using the "hard" method described in Section 2.1. For the other three boundary conditions, we introduce three intermediate output variables:

$$q_1 = \frac{\partial T}{\partial r}; \quad q_2 = \frac{\partial T}{\partial z}; \quad q_3 = h\left(T - T_0\right) + k\frac{\partial T}{\partial r}. \tag{B.1}$$

These intermediate variables are defined over the entire domain (r,z), and their corresponding connecting equations are given as follows:

$$f^1_{\text{cnc}}(r,z) = q_1(r,z) - \frac{\partial T(r,z)}{\partial r},$$

$$f^2_{\text{cnc}}(r,z) = q_2(r,z) - \frac{\partial T(r,z)}{\partial z},$$

$$f^3_{\text{cnc}}(r,z) = q_3(r,z) - h\left(T(r,z) - T_0\right) - k\frac{\partial T(r,z)}{\partial r}.$$

Using these intermediate variables and connecting equations, the boundary conditions can be reformulated as:

$$q_1(r,z) = 0, \ (r,z) \in \Gamma_1, \tag{B.2}$$

$$q_2(r,z) = 0, \ (r,z) \in \Gamma_2, \tag{B.3}$$

$$q_3(r,z) = 0, \ (r,z) \in \Gamma_3, \tag{B.4}$$

which are Dirichlet boundaries with respect to the intermediate variables $[q_1, q_2, q_3]^\top$, and can therefore be directly imposed using the "hard" method.

Step 3 is to reinforce the boundary regions. The loss term corresponding to each connecting equation can be expressed as:

$$\mathcal{L}^i_{\text{cnc}} = \omega^i_{\text{cnc}}\ \text{mean}\left(\left(f^i_{\text{cnc}} \cdot \text{BRRF}\right)^2\right). \tag{B.5}$$

The output variables of the neural networks is $[T, q_1, q_2, q_3]^\top$, and the loss function is $\mathcal{L} = \mathcal{L}_{\text{PDE}} + \sum_{i=1}^{3}\mathcal{L}^i_{\text{cnc}}$. The network is then trained to minimize this total loss function.

In summary, the entire process described above is illustrated in Fig. 3.

B.2. Elastic deformation problem implementation

The formulation of the elastic deformation problem is presented in (A.8)–(A.13). It is important to emphasize that only (A.8) and (A.9) are the governing PDEs, representing the force balance of the differential elements. In contrast, (A.10)–(A.13) are the constitutive relations that describe the relationship between stress and displacement in specific materials and are distinct from the governing equilibrium equations. Therefore, the primitive variables are u_r and u_z.

The next step addresses the eight boundary conditions presented in (A.14)–(A.17). The condition $u_z = 0$ on the boundary Γ_2 is a standard Dirichlet condition which can be "hard" imposed. The remaining conditions are not Dirichlet-type with respect to primitive variables are u_r and u_z. It is natural to define four intermediate variables, σ_{rr}, $\sigma_{\theta\theta}$, σ_{zz}, σ_{rz}, and (A.10)–(A.13) will serve as the connecting equations. Compared to conventional PINNs, whose loss function consists of 2+8 = 10 terms, the BRR-PINNs method requires only 6 loss terms.

Finally, we proceed the boundary region reinforcement, which is the same as Step 3 of the heat transfer case in Appendix B.1.

Table C.6
Experimental setup for the heat transfer problem.

Experimental setup	
Neural network	5 × 64 MLP
Activation function	tanh
Optimizer	Adam
Sampling method	Random per epoch
Collocation points in domain	128 × 128
Collocation points per boundary	64
Batch size	1024
Checkpoint window size W	100
Stopping threshold ε	0.01
GPU	NVDIA RTX 4090

Appendix C. Missing details for Section 3

C.1. Hyperparameter selection and stopping criteria

Hyperparameter selection. The success of PINN-based methods relies on careful hyperparameter tuning, including the learning rate, the weights assigned to each loss term, the boundary reinforcement coefficient, and others. In this paper, we adopt an empirical tuning strategy. For each method-conventional PINNs, "hard" method PINNs, g-PINNs, and BRR-PINNs-we follow the tuning procedure outlined below:

- Step 1: Select a learning rate that ensures a stable and consistent decrease of the loss function.
- Step 2 (if applicable): Adjust the boundary reinforcement coefficient β.
- Step 3: Empirically adjust the weights of the individual loss terms according to their training behaviors.

Stopping criteria. Our stopping criteria is to monitor the whole loss function L, and denote by $L(i)$ the loss value at ith epoch. We define checkpoints at epochs

$$kW, \quad k = 1, 2, 3, \ldots$$

For each checkpoint kW with $k \geq 2$, compute the average loss over current and previous windows of length W:

$$L_{\text{cur}}(k) := \frac{1}{W} \sum_{i=(k-1)W+1}^{kW} L(i),$$

$$L_{\text{pre}}(k) := \frac{1}{W} \sum_{i=(k-2)W+1}^{(k-1)W} L(i).$$

The relative improvement between two successive windows is defined as

$$\Delta(k) := \frac{L_{\text{pre}}(k) - L_{\text{cur}}(k)}{L_{\text{cur}}(k)}.$$

Using $L_{\text{cur}}(k)$ as the denominator makes the measure sensitive to orders of magnitude reductions. For example, when the loss decreases from 0.01 to 0.001, the metric yields $\Delta(k) = 9$, while the decrease from 0.11 to 0.1 results in $\Delta(k) = 0.1$. We then terminate training once the relative improvement falls below a threshold, i.e., $\Delta(k) \leq \varepsilon$. In this work, we set the stopping threshold to $\varepsilon = 0.01$.

C.2. Numerical experiment details

C.2.1. Details for heat transfer problem

We follow the hyperparameter tuning strategy and the stopping criteria described in Appendix C.1. We first set all loss term weights to 1 and the boundary reinforcement coefficient β to 10, then tune the learning rate for each method. As shown in Fig. C.19, we select a learning rate of 1e-5 for all methods. We also note that the oscillations observed in g-PINNs arise from higher-order differentiation rather than from the choice of learning rate.

We then tune the other parameters including the loss term weights and boundary reinforce coefficient. We summarize the tuning process in the following Table C.7. The loss value we report is the average of the last $W = 100$ epochs. We note that parameter tuning and selection are based on the behavior of each individual loss term; the errors with respect to the "true" FEM solution are provided solely for reference. The entries in bold in Table C.7 are the set of selected parameters, and we provide the training process for each loss term in Fig. C.20.

For a more intuitive comparison, Fig. C.21 presents the predicted temperature-rise distributions alongside the absolute error with respect to the "true" FEM solution. It is clearly observed that BRR-PINNs outperform the other methods.

C.2.2. Details for elastic deformation problem

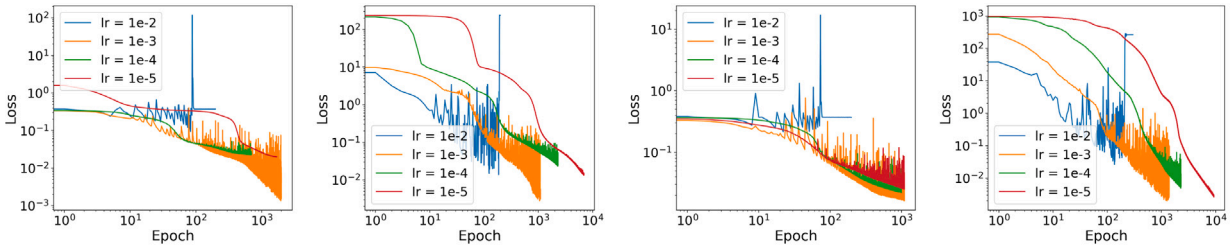
The predicted displacement maps alongside the absolute error with respect to the "true" FEM solution for each method are shown in Fig. C.22.

C.2.3. Details for thermo-elastic deformation problem

We use the same experimental setup with the case of elastic deformation problem, as shown in Table C.8. The predicted displacement and error maps are shown in Fig. C.23 (see Tables C.9–C.12).

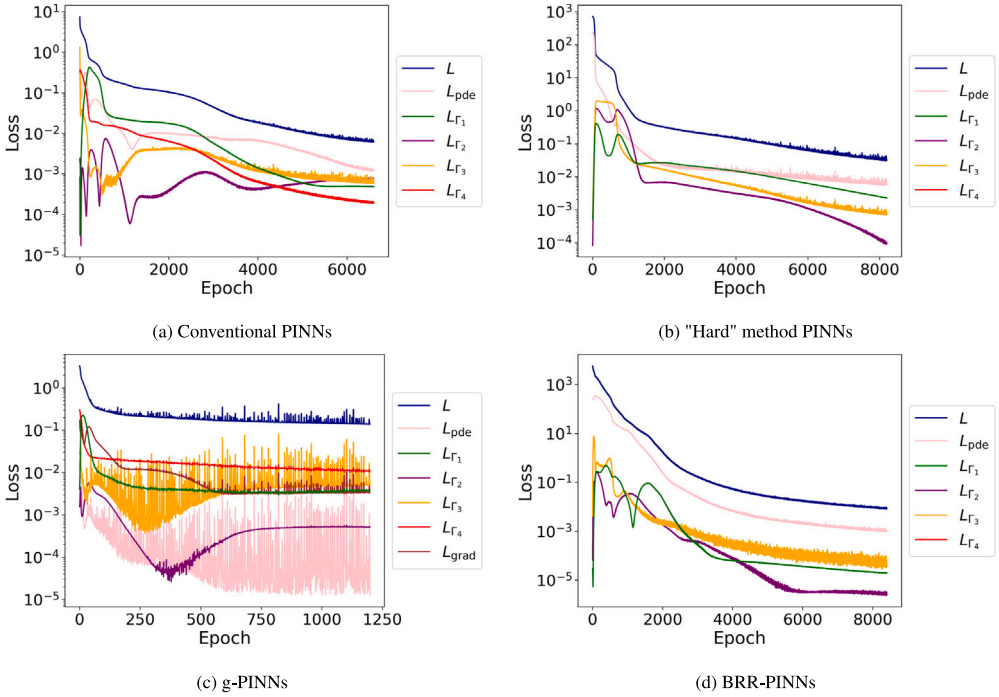
Table C.7
Comparison of memory usage, computational time, and training complexity. For the first three methods, the loss-term weights follow the order $[L_{\text{pde}}, L_{\Gamma_1}, \ldots, L_{\Gamma_4}, L_{\text{grad}}]$. For BRR-PINNs, the weights are ordered as L_{pde} and the connecting equation losses generated by boundaries $\Gamma_1, \Gamma_2, \Gamma_3$, respectively. Entries in bold highlight the chosen parameter sets along with their corresponding outcomes.

Method	Loss term weights	β	L	L_{pde}	L_{Γ_1}	L_{Γ_2}	L_{Γ_3}	L_{Γ_4}	L_{grad}	Stop epoch	$\varepsilon_{\text{rmse}}$
Conventional PINNs	[1,1,1,1,1,–]	–	1.9e−2	7.8e−4	1.2e−3	1.5e−4	9.5e−5	1.7e−2	–	1500	1.2e−1
	[1,1,1,1,10,–]	–	5.5e−3	1.3e−3	5.1e−4	8.1e−4	1.5e−3	1.4e−4	–	7600	1.4e−2
	[1,1,1,3,10,–]	**–**	**6.4e−3**	**1.2e−3**	**4.9e−4**	**7.9e−4**	**6.3e−4**	**2.0e−4**	**–**	**6600**	**1.5e−2**
"Hard" PINNs	[1,1,1,1,–,–]	–	2.4e−2	6.2e−3	8.9e−3	9.4e−4	7.9e−3	0	–	7400	1.2e−2
	[1,3,1,3,–,–]	–	8.9e−2	2.7e−2	8.9e−3	2.7e−3	1.1e−2	0	–	4100	1.5e−2
	[3,3,1,10,–,–]	**–**	**3.3e−2**	**6.2e−3**	**2.3e−3**	**1.0e−4**	**7.8e−4**	**0**	**–**	**8200**	**7.1e−3**
g-PINNs	[1,1,1,1,1,1]	–	2.6e−2	2.1e−4	1.9e−3	3.7e−4	3.4e−3	2.0e−2	8.9e−4	1200	1.3e−1
	[1,1,1,1,10,1]	–	1.1e−1	2.8e−4	2.3e−2	4.9e−4	2.4e−2	5.6e−3	5.9e−3	1300	7.0e−2
	[1,3,1,1,10,1]	–	1.3e−1	5.1e−4	4.2e−3	3.3e−3	2.7e−2	7.9e−3	3.5e−3	1400	8.0e−2
	[1,3,1,3,10,1]	**–**	**1.5e−1**	**3.3e−4**	**3.7e−3**	**5.2e−4**	**8.3e−3**	**1.1e−2**	**3.7e−3**	**1200**	**9.9e−2**
BRR-PINNs	[1,1,1,1]	1	2.9e−3	4.5e−4	4.9e−3	3.2e−3	1.5e−2	0	–	7700	4.6e−2
	[1,1,1,1]	10	2.8e−3	5.1e−4	4.0e−5	6.9e−6	1.3e−4	0	–	9300	1.0e−3
	[1,1,1,1]	**30**	**8.7e−3**	**1.0e−3**	**2.0e−5**	**2.8e−6**	**5.4e−5**	**0**	**–**	**8400**	**6.0e−4**
	[1,1,1,1]	100	8.5e−2	1.2e−2	1.7e−5	3.6e−6	4.1e−5	0	–	8400	7.9e−4
	[1,1,1,1]	300	2.9e−1	8.7e−2	2.2e−6	1.8e−6	1.8e−5	0	–	9700	1.3e−3
	[1,3,1,3]	**30**	1.3e−2	9.2e−4	6.3e−6	1.7e−5	3.5e−5	0	–	11700	2.2e−3
	[3,1,1,1]	**30**	1.8e−2	1.2e−3	7.7e−6	1.8e−5	1.2e−4	0	–	6100	2.0e−3
	[10,1,1,3]	**30**	5.1e−2	1.1e−3	1.5e−4	4.9e−5	8.5e−5	0	–	8900	3.2e−3



(a) Conventional PINNs (b) "Hard" method PINNs (c) g-PINNs (d) BRR-PINNs

Fig. C.19. Learning rate tuning for each method.



(a) Conventional PINNs (b) "Hard" method PINNs

(c) g-PINNs (d) BRR-PINNs

Fig. C.20. The training process of the selected model for each method.

Table C.8
Experimental setup for the elastic deformation problem.

Experimental setup	
Neural network	5 × 128 MLP
Activation function	tanh
Optimizer	Adam
Sampling method	Random per epoch
Collocation points in domain	128 × 128
Collocation points per boundary	128
Batch size	4096
Checkpoint window size W	1000
Stopping threshold ϵ	0.01
GPU	NVDIA RTX 4090

Table C.9
The selected weight parameters for BRR-PINNs method in the elastic deformation problem.

Learning rate	L^1_{PDE}	L^2_{PDE}	$L_{\sigma_{rr}}$	$L_{\sigma_{\theta\theta}}$	$L_{\sigma_{zz}}$	$L_{\sigma_{rz}}$	β
1e–5	1	1	3	1	3	10	100

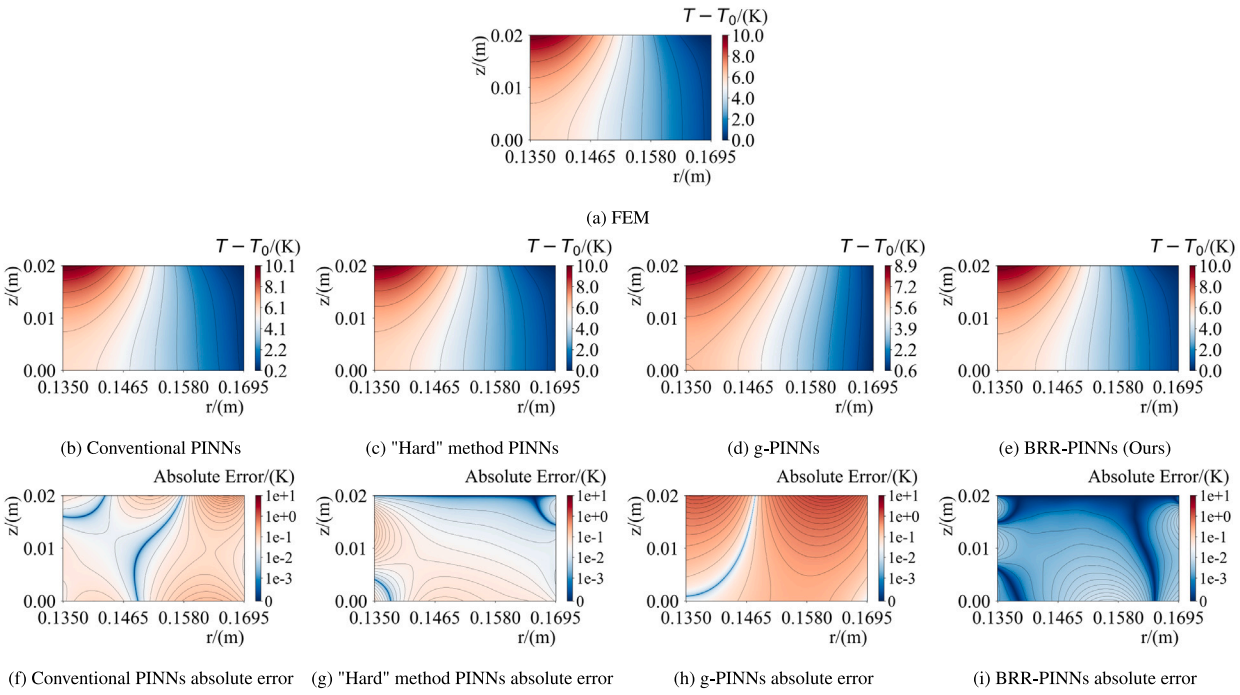


Fig. C.21. Comparison of the resulting temperature-rise distributions from several existing methods and BRR-PINNs.

Table C.10
The selected weight parameters for conventional PINNs and "hard" method PINNs in the elastic deformation problem.

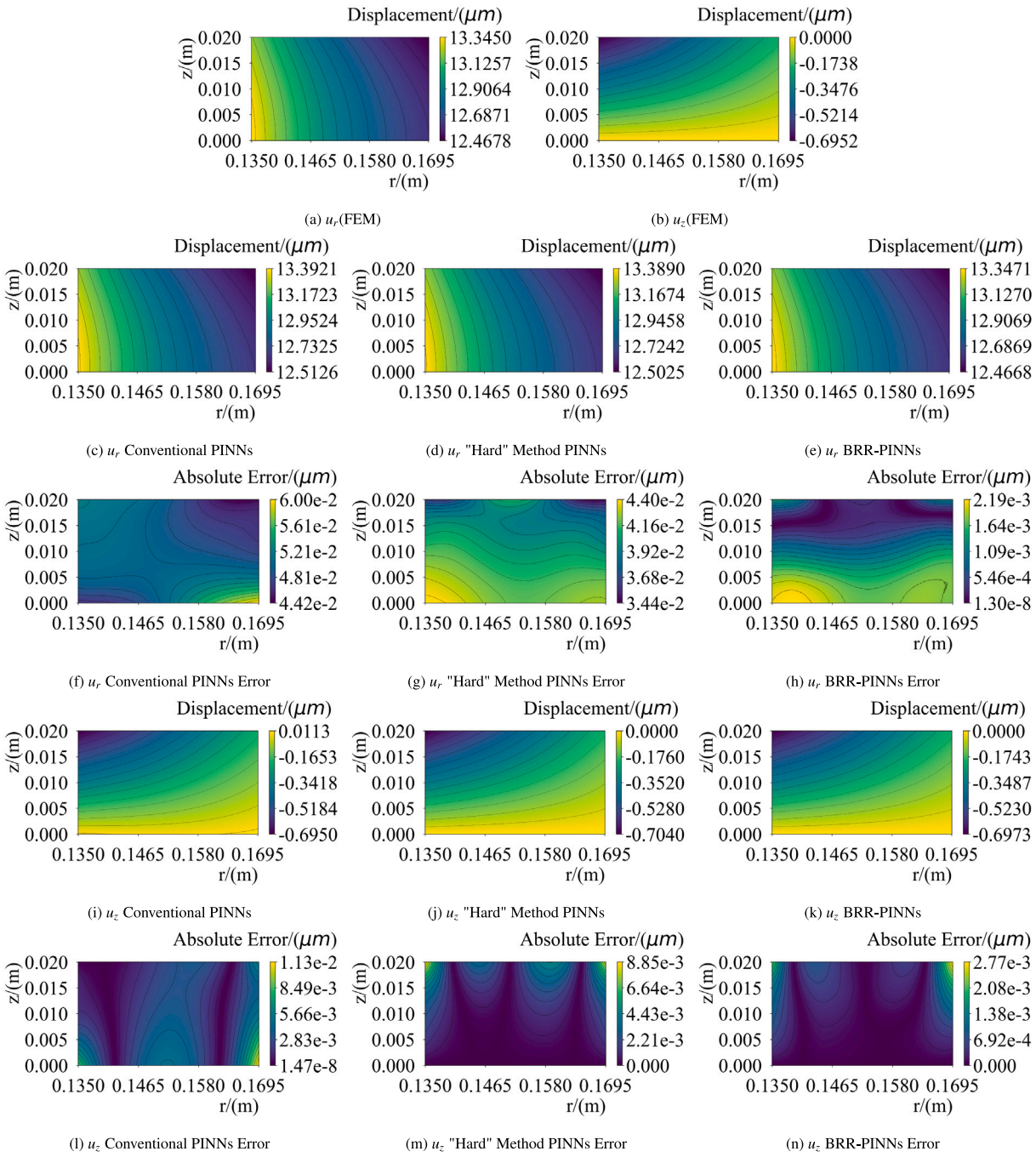
Method	Learning rate	L^1_{PDE}	L^2_{PDE}	$L^{\sigma_{rr}}_{\Gamma_1}$	$L^{\sigma_{rz}}_{\Gamma_1}$	$L^{u_z}_{\Gamma_2}$	$L^{\sigma_{rz}}_{\Gamma_2}$	$L^{\sigma_{rr}}_{\Gamma_3}$	$L^{\sigma_{rz}}_{\Gamma_3}$	$L^{\sigma_{zz}}_{\Gamma_4}$	$L^{\sigma_{rz}}_{\Gamma_4}$
Conventional PINNs	1e–5	1	1	3	10	3	1	1	10	30	1
"Hard" Method PINNs	1e–5	1	1	10	3	–	1	1	1	30	1

Table C.11
The selected weight parameters for BRR-PINNs method in the thermal-elastic deformation problem.

Learning rate	L^1_{PDE}	L^2_{PDE}	$L_{\sigma_{rr}}$	$L_{\sigma_{\theta\theta}}$	$L_{\sigma_{zz}}$	$L_{\sigma_{rz}}$	β
1e–5	1	1	3	1	3	10	100

Table C.12
The selected weight parameters for conventional PINNs and "hard" method PINNs in the thermo-elastic deformation problem.

Method	Learning rate	L^1_{PDE}	L^2_{PDE}	$L^{\sigma_{rr}}_{\Gamma_1}$	$L^{\sigma_{rz}}_{\Gamma_1}$	$L^{u_z}_{\Gamma_2}$	$L^{\sigma_{rz}}_{\Gamma_2}$	$L^{\sigma_{rr}}_{\Gamma_3}$	$L^{\sigma_{rz}}_{\Gamma_3}$	$L^{\sigma_{zz}}_{\Gamma_4}$	$L^{\sigma_{rz}}_{\Gamma_4}$
Conventional PINNs	1e–5	1	1	1	3	30	1	1	1	10	10
"Hard" Method PINNs	1e–5	1	1	1	10	–	1	1	3	30	10



(a) u_r(FEM) (b) u_z(FEM)

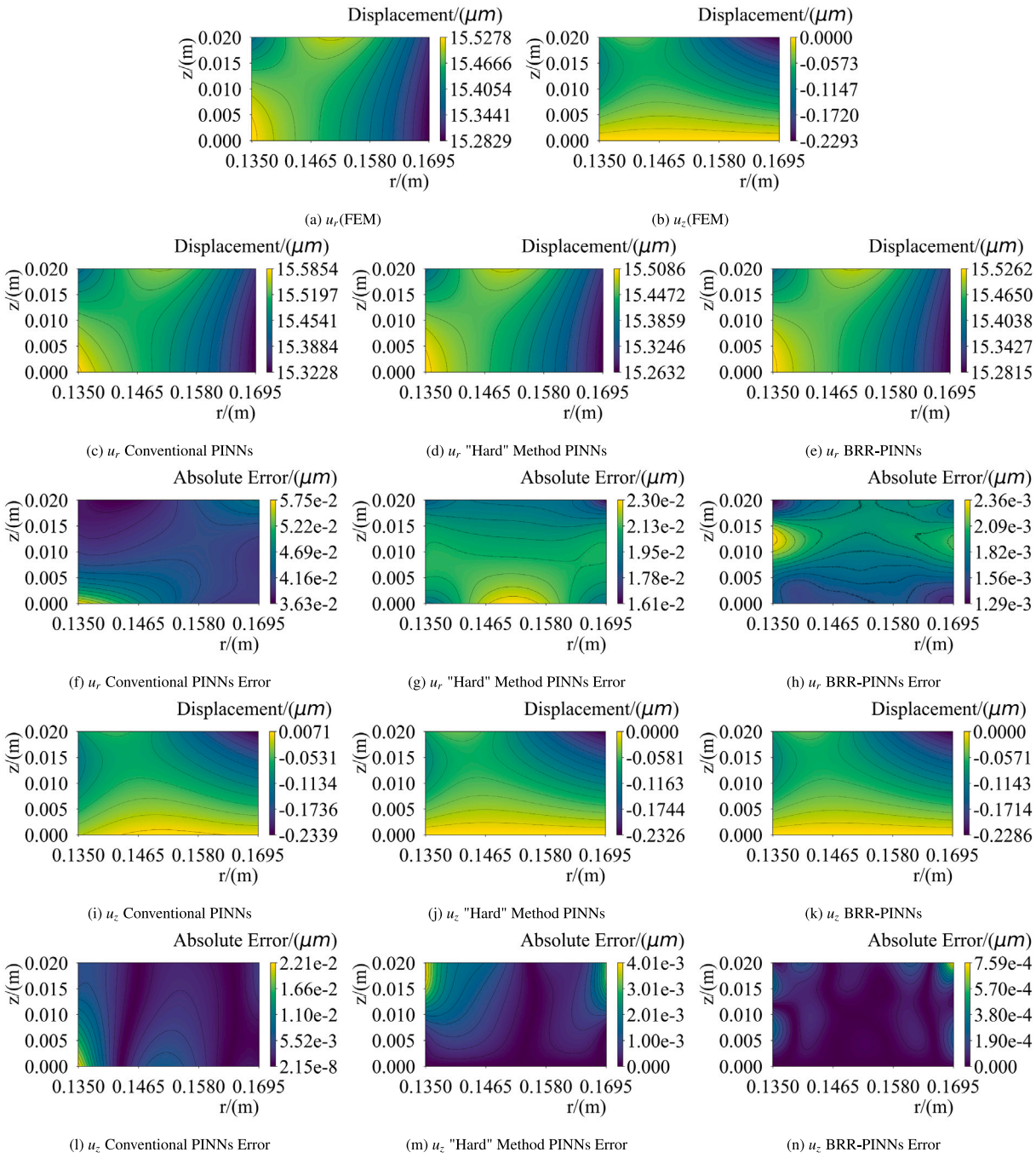
(c) u_r Conventional PINNs (d) u_r "Hard" Method PINNs (e) u_r BRR-PINNs

(f) u_r Conventional PINNs Error (g) u_r "Hard" Method PINNs Error (h) u_r BRR-PINNs Error

(i) u_z Conventional PINNs (j) u_z "Hard" Method PINNs (k) u_z BRR-PINNs

(l) u_z Conventional PINNs Error (m) u_z "Hard" Method PINNs Error (n) u_z BRR-PINNs Error

Fig. C.22. Comparison of the resulting displacement maps from several existing methods and BRR-PINNs in the elastic deformation problem.



(a) u_r(FEM) (b) u_z(FEM)

(c) u_r Conventional PINNs (d) u_r "Hard" Method PINNs (e) u_r BRR-PINNs

(f) u_r Conventional PINNs Error (g) u_r "Hard" Method PINNs Error (h) u_r BRR-PINNs Error

(i) u_z Conventional PINNs (j) u_z "Hard" Method PINNs (k) u_z BRR-PINNs

(l) u_z Conventional PINNs Error (m) u_z "Hard" Method PINNs Error (n) u_z BRR-PINNs Error

Fig. C.23. Comparison of the resulting displacement maps from several existing methods and BRR-PINNs in the thermo-elastic deformation problem.

Data availability

Data will be made available on request.

References

Agmon, S., Douglis, A., Nirenberg, L., 1959. Estimates near the boundary for solutions of elliptic partial differential equations satisfying general boundary conditions. I. Comm. Pure Appl. Math. 12 (4), 623–727.

Almajid, M.M., Abu-Al-Saud, M.O., 2022. Prediction of porous media fluid flow using physics informed neural networks. J. Pet. Sci. Eng. 208, 109205.

Anagnostopoulos, S.J., Toscano, J.D., Stergiopulos, N., Karniadakis, G.E., 2024. Residual-based attention in physics-informed neural networks. Comput. Methods Appl. Mech. Engrg. 421, 116805.

Aygun, A., Maulik, R., Karakus, A., 2023. Physics-informed neural networks for mesh deformation with exact boundary enforcement. Eng. Appl. Artif. Intell. 125, 106660.

Babuška, I., Banerjee, U., Osborn, J.E., 2003. Survey of meshless and generalized finite element methods: a unified approach. Acta Numer. 12, 1–125.

Bai, J., Rabczuk, T., Gupta, A., Alzubaidi, L., Gu, Y., 2023. A physics-informed neural network technique based on a modified loss function for computational 2D and 3D solid mechanics. Comput. Mech. 71 (3), 543–562.

Baldan, M., Di Barba, P., 2024. Energy-based PINNs for solving coupled field problems: Concepts and application to the multi-objective optimal design of an induction heater. IET Sci. Meas. Technol.

Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M., 2018. Automatic differentiation in machine learning: a survey. J. Mach. Learn. Res. 18 (153), 1–43.

Berrone, S., Canuto, C., Pintore, M., Sukumar, N., 2023. Enforcing Dirichlet boundary conditions in physics-informed neural networks and variational physics-informed neural networks. Heliyon 9 (8).

Cai, S., Wang, Z., Wang, S., Perdikaris, P., Karniadakis, G.E., 2021. Physics-informed neural networks for heat transfer problems. J. Heat Transf. 143 (6), 060801.

Chen, J., Du, R., Wu, K., 2020. A comparison study of deep Galerkin method and deep Ritz method for elliptic problems with different boundary conditions. arXiv preprint arXiv:2005.04554.

Chiu, P.-H., Wong, J.C., Ooi, C., Dao, M.H., Ong, Y.-S., 2022. CAN-PINN: A fast physics-informed neural network based on coupled-automatic–numerical differentiation method. Comput. Methods Appl. Mech. Engrg. 395, 114909.

De Florio, M., Schiassi, E., Furfaro, R., 2022. Physics-informed neural networks and functional interpolation for stiff chemical kinetics. Chaos: Interdiscip. J. Nonlinear Sci. 32 (6).

Eshkofti, K., Hosseini, S.M., 2023. A gradient-enhanced physics-informed neural network (gPINN) scheme for the coupled non-fickian/non-fourierian diffusion-thermoelasticity analysis: A novel gPINN structure. Eng. Appl. Artif. Intell. 126, 106908.

Goswami, S., Anitescu, C., Chakraborty, S., Rabczuk, T., 2020. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. Theor. Appl. Fract. Mech. 106, 102447.

Haghighat, E., Raissi, M., Moure, A., Gomez, H., Juanes, R., 2021. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. Comput. Methods Appl. Mech. Engrg. 379, 113741.

Harandi, A., Moeineddin, A., Kaliske, M., Reese, S., Rezaei, S., 2024. Mixed formulation of physics-informed neural networks for thermo-mechanically coupled systems and heterogeneous domains. Internat. J. Numer. Methods Engrg. 125 (4), e7388.

Jagtap, A.D., Kharazmi, E., Karniadakis, G.E., 2020. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. Comput. Methods Appl. Mech. Engrg. 365, 113028.

Ji, W., Qiu, W., Shi, Z., Pan, S., Deng, S., 2021. Stiff-pinn: Physics-informed neural network for stiff chemical kinetics. J. Phys. Chem. A 125 (36), 8098–8106.

Jin, X., Cai, S., Li, H., Karniadakis, G.E., 2021. (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. J. Comput. Phys. 426, 109951.

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.

Kathane, S., Karagadde, S., 2024. A physics informed neural network (PINN) methodology for coupled moving boundary PDEs. arXiv preprint arXiv:2409.10910.

Khadijeh, M., Cerqueglini, V., Kasbergen, C., Erkens, S., Varveri, A., 2025. Multistage physics informed neural network for solving coupled multiphysics problems in material degradation and fluid dynamics. Eng. Comput. 1–31.

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., Mahoney, M.W., 2021. Characterizing possible failure modes in physics-informed neural networks. Adv. Neural Inf. Process. Syst. 34, 26548–26560.

Laubscher, R., 2021. Simulation of multi-species flow and heat transfer using physics-informed neural networks. Phys. Fluids 33 (8).

Laubscher, R., Rousseau, P., 2022. Application of a mixed variable physics-informed neural network to solve the incompressible steady-state and transient mass, momentum, and energy conservation equations for flow over in-line heated tubes. Appl. Soft Comput. 114, 108050.

Le-Duc, T., Lee, S., Nguyen-Xuan, H., Lee, J., 2024. A hierarchically normalized physics-informed neural network for solving differential equations: Application for solid mechanics problems. Eng. Appl. Artif. Intell. 133, 108400.

Lu, L., Pestourie, R., Yao, W., Wang, Z., Verdugo, F., Johnson, S.G., 2021. Physics-informed neural networks with hard constraints for inverse design. SIAM J. Sci. Comput. 43 (6), B1105–B1132.

Luong, K.A., Le-Duc, T., Lee, J., 2023. Deep reduced-order least-square method—A parallel neural network structure for solving beam problems. Thin-Walled Struct. 191, 111044.

Lyu, L., Wu, K., Du, R., Chen, J., 2020. Enforcing exact boundary and initial conditions in the deep mixed residual method. arXiv preprint arXiv:2008.01491.

Ma, Y., Xu, X., Yan, S., Ren, Z., 2022. A preliminary study on the resolution of electro-thermal multi-physics coupling problem using physics-informed neural network (PINN). Algorithms 15 (2), 53.

Masclans, N., Vázquez-Novoa, F., Bernades, M., Badia, R.M., Jofre, L., 2023. Thermodynamics-informed neural network for recovering supercritical fluid thermophysical information from turbulent velocity data. Int. J. Thermofluids 20, 100448.

New, A., Eng, B., Timm, A.C., Gearhart, A.S., 2023. Tunable complexity benchmarks for evaluating physics-informed neural networks on coupled ordinary differential equations. In: 2023 57th Annual Conference on Information Sciences and Systems. CISS, IEEE, pp. 1–8.

Niaki, S.A., Haghighat, E., Campbell, T., Poursartip, A., Vaziri, R., 2021. Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture. Comput. Methods Appl. Mech. Engrg. 384, 113959.

Praditia, T., Walser, T., Oladyshkin, S., Nowak, W., 2020. Improving thermochemical energy storage dynamics forecast with physics-inspired neural network architecture. Energies 13 (15), 3873.

Pu, R., Feng, X., 2022. Physics-informed neural networks for solving coupled Stokes–Darcy equation. Entropy 24 (8), 1106.

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. J. Comput. Phys. 378, 686–707.

Raj, M., Kumbhar, P., Annabattula, R.K., 2021. Physics-informed neural networks for solving thermo-mechanics problems of functionally graded material. arXiv preprint arXiv:2111.10751.

Rao, C., Sun, H., Liu, Y., 2021. Physics-informed deep learning for computational elastodynamics without labeled data. J. Eng. Mech. 147 (8), 04021043.

Rezaei, S., Harandi, A., Moeineddin, A., Xu, B.-X., Reese, S., 2022. A mixed formulation for physics-informed neural networks as a potential solver for engineering problems in heterogeneous domains: Comparison with finite element method. Comput. Methods Appl. Mech. Engrg. 401, 115616.

Rvachev, V.L., Sheiko, T.I., 1995. R-functions in boundary value problems in mechanics.

Sheng, H., Yang, C., 2021. PFNN: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries. J. Comput. Phys. 428, 110085.

Shi, S., Liu, D., Huo, Z., 2023. Simulation of thermoelastic coupling in silicon single crystal growth based on alternate two-stage physics-informed neural network. Eng. Appl. Artif. Intell. 123, 106468.

Sukumar, N., Srivastava, A., 2022. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. Comput. Methods Appl. Mech. Engrg. 389, 114333.

Sun, L., Gao, H., Pan, S., Wang, J.-X., 2020. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. Comput. Methods Appl. Mech. Engrg. 361, 112732.

Wang, J., Mo, Y., Izzuddin, B., Kim, C.-W., 2023. Exact Dirichlet boundary physics-informed neural network EPINN for solid mechanics. Comput. Methods Appl. Mech. Engrg. 414, 116184.

Wang, S., Teng, Y., Perdikaris, P., 2021. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM J. Sci. Comput. 43 (5), A3055–A3081.

Wang, S., Yu, X., Perdikaris, P., 2022. When and why PINNs fail to train: A neural tangent kernel perspective. J. Comput. Phys. 449, 110768.

Xie, J., Chai, Z., Xu, L., Ren, X., Liu, S., Chen, X., 2022. 3D temperature field prediction in direct energy deposition of metals using physics informed neural network. Int. J. Adv. Manuf. Technol. 119 (5), 3449–3468.

Yang, L., Meng, X., Karniadakis, G.E., 2021. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. J. Comput. Phys. 425, 109913.

Yazdani, A., Lu, L., Raissi, M., Karniadakis, G.E., 2020. Systems biology informed deep learning for inferring parameters and hidden dynamics. PLoS Comput. Biol. 16 (11), e1007575.

Yu, J., Lu, L., Meng, X., Karniadakis, G.E., 2022. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. Comput. Methods Appl. Mech. Engrg. 393, 114823.

Zobeiry, N., Humfeld, K.D., 2021. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. Eng. Appl. Artif. Intell. 101, 104232.