Babylonian Journal of Mathematics
Vol. (**2026**), **pp**. 11–18
DOI: https://doi.org/10.58496/BJM/2026/002; ISSN: 3006-113X
https://mesopotamian.press/journals/index.php/BJM

Research Article

Gradient-Regularized Deep Ritz Networks for Solving Elliptic PDEs: Application to the Poisson Equation

H.K. Al-Mahdawi[1, *], Aleksandra Petrovic [2], Sandeep Poddar [3]

[1] *Electronic Computer Centre, University of Diyala, 32001, Iraq,*
[2] *Deputy Vice Chancellor (Research & Innovation), Lincoln University College, Malaysia,*
[3] *Faculty of Technical Sciences, University of Pristina in Kosovska Mitrovica, Republic of Serbia,*

ARTICLE INFO

Article History
Received 23 Oct 2025
Revised 18 Nov 2025
Accepted 25 Dec 2025
Published 10 Jan 2026

Keywords
Deep Ritz Method;
Gradient Regularization,
Elliptic PDEs.
Variational formulation,
Numerical analysis.

ABSTRACT

In this work we propose a gradient-regularized Deep Ritz Network (GR-DRN) for solving elliptic partial differential equations, which we shall concretely apply to the Poisson equation on the unit square domain. We then propose a method that generalizes the classical Deep Ritz formulation by adding a gradient-penalty term to achieve smoother solution and lower non-physical oscillations during training. The model is evaluated against three benchmark datasets that are analytically generated, representing a smooth solution, a case with high-frequency oscillatory solution, and a variable-coefficient setting that reflects heterogeneous diffusion. Our numerical results show that GR-DRN consistently obtains lower L^2 and H^1 errors than the standard Deep Ritz method, especially on the difficult problems with high-frequency or variable-coefficient. It also produces more accurate gradient fields and stable convergence shift. Overall this shows that gradient regularization is an easy but effective improvement to neural variational solvers for elliptic PDEs.

1. INTRODUCTION

Partial differential equations (PDEs) play a fundamental role in the mathematical description of a large variety of physical, biological, and engineering systems such as fluid flow, heat transfer, solid mechanics and other complicated transport processes. It has been shown that classical numerical methods, namely finite differences, finite elements or spectral schemes, can be extraordinarily efficient to solve PDEs if the associated meshes are well-structured (or unstructured to some extent), but face tremendous difficulties in high-dimension setups, high-contrast geometries, heterogeneous coefficients or inverse formulations with scarce and over-isolating information. The advent of scientific machine learning has been accompanied by a lot of excitement over neural-network-based solvers that embed physical laws directly into the learning process, seeking a fruitful marriage between the flexibility of deep learning and traditional numerical analysis structure and stability. In this paradigm, Physics-informed neural networks (PINNs) has been suggested as a general framework for forward and inverse solutions of nonlinear PDEs [1], where governing equations as well as boundary and initial conditions satisfied by the solution are enforced in the loss function. Therefore, PINNs first introduced in are simply using a neural network to approximate the solution of a PDE, where its key idea is to train the neural network by minimizing the residual of the underlying differential operator at scattered collocation points in the domain. Such strategy has been further generalized and evaluated in a systematic manner for multiple problems such as fluid mechanics, solid mechanics, and multi-physics coupling, establishing a mesh-free solver capability towards physics-informed architectures [2]–[4]. Recent work [2] has demonstrated how PINN formulations (using appropriate constitutive modeling and loss formulations) can be specialized to elasticity, plasticity, and fracture in the setting of computational solid mechanics. Survey contributions [4] have clarified contemporary possibilities and future directions in scientific machine learning through a PINN lens. In addition, domain decomposition strategies like extended PINNs (XPINNs) have been suggested to enhance scalability and accuracy by representing local networks on subregions of the space—time domain, coupled through interface conditions [3]. However, a lot of recent research has revealed that PINNs may suffer from extreme training pathologies. Specifically, gradient imbalance among loss components, stiffness caused by PDE operator, and neural networks spectral bias can converge slowly or poorly approximate sharp gradients or high-frequency modes [5].

*Corresponding author. Email: hssnkd@gmail.com

Thorough investigations into the optimization landscape have revealed that standard PINN training can be biased by regions of the domain or terms in the loss, leading to false solutions with small overall residual [5]. In order to solve these problems few of adaptive and self-correcting variants of PINNs were introduced. For example, self-adaptive PINNs dynamically adjust the sampling or weighting of collocation points per epoch to priorities training on the regions where the solution or residual is harder to approximate [7] whilst loss-balanced or self-adaptive loss strategies re-scale different loss terms in order to alleviate the gradient pathologies optimizing for each of the PINN objective terms leads to and improve robustness [8]. Furthermore, related work on adaptive activation functions, which have shown to speed up convergence and improve approximation in PINN frameworks [9], has demonstrated the benefit of tuning the networks nonlinearity during training. Meanwhile, many Bayesian extensions have also been proposed to add uncertainty quantification and inverse parameter estimation into the physics-informed neural solvers, such as B-PINNs [10], and recent analyses based on the neural tangent kernel have shed new light on the convergence properties of PINNs and similar architectures [6]. This includes but is not limited to, connections of the challenges faced in training physics-informed networks to the classical notions of ill-posed Ness and regularization in inverse problems and operator equations. It is well known that inverse PDE problems are ill-posed in the Hadamard sense and that regularization is needed to get stable and meaningful solutions from noisy or incomplete data [11]–[13]. Early results on Tikhonov, Lavrentiev and similar regularization methods demonstrated that by including suitable penalty terms (typically, norms of the solution and/or its derivatives) stability can be recovered by preventing deviation of the solution from physically reasonable values [11]–[13]. They have been applied with success to heat conduction and heat exchange inverse boundary problems, where one obtains stably reconstructed boundary data or coefficients using Lavrentiev and iterated Lavrentiev regularization, along with finite-dimensional approximation [14]–[16]. Subsequent work has studied the use of optimization heuristics like particle swarm optimization to choose regularization parameters in Tikhonov methods for inverse problems, further highlighting the need for principled regularization design in practical computations [17][20]. More generally from PDE point of view, this necessity for reliable and well-posed numerical methods can be extended also for nonlinear evolution equations. Early attempts to analyze models such as the Burgers equation, first proposed as a simple mathematical model that captures some of the features of turbulence [18] and later under much more rigorous analyses relevant for nonlinear convection–diffusion equations [19], has uncovered the existence of steep gradients, shocks, and complicated transient dynamics. Indeed, these features are well known to be hard to approximate accurately under compliance with suitable stabilization or regularization mechanisms in numerically discretized schemes. While the current work considers elliptic problems, the lessons learned from the nonlinear PDEs, e.g., Burgers' equation indicate that controlling gradients and oscillations is a key issue of any neural-network based PDE solver. Inspired by the convergence and stability issues seen with existing PINNs, and the well-established theory regarding regularization for ill-posed problems, this paper examines a gradient-regularized neural variational framework for elliptic PDEs. In particular, we formulate a Deep Ritz type formulation for the Poisson equation, where given a minimizer of energy functional associated to the solution which is expressed as an ansatz by a neural network, we supplement the energy functional with an inner gradient penalty that explicitly constrains the learned gradient field to be small and smooth. Good wording but annoying notations [1]. In the PINNs we try to enforce the PDE residual in the loss by using high order differential operators, while the Deep Ritz is based on The variational structure of elliptic problems and obtains a training objective that is defined in terms of an energy which naturally connects with gradient regularization. Motivated by Tikhonov-type penalties [11]-[13] and the general framework of regularization in inverse and ill-posed problems [14]-[17], we propose a modification of the gradient-regularized Deep Ritz network, which aims to reduce non-physical oscillations, better approximate fluxes, and also increase robustness in training. Specifically, we analyze the nonlocal functional within this framework, by solving the Poisson equation on the unit square (as a canonical elliptic PDE benchmark) in smooth, high-frequency, and variable-coefficient settings. The objective is to investigate how gradient regularization influences the L2L^{2}L2 and H1H^{1}H1 norms of the neural approximation accuracy, the fidelity of the reconstructed gradient and flux fields, as well as the convergence characteristics of the training. Unlike most other existing PINN variants that focus on residual term reweighting [5], [7], [8] or network architecture adaptivity [3], [9], this work is centered around a variational regularization approach that incorporates ideas from classical inverse problem theory [11]–[13]. The findings emphasize that adding gradient-based penalties to the Deep Ritz loss is a valuable and conceptually straightforward approach to enhance neural-network-based solvers for elliptic PDEs and make progress towards the development of more general regularized physics-informed learning frameworks.

2. PROBLEM STATEMENT

Elliptic partial differential equations arise at the core of steady-state physical modeling problems which are found in various applications domains, the electrostatics, heat conduction, incompressible fluid flow, diffusion in heterogeneous media and structural equilibrium problems. One of the more fundamental formulations of these equations is the Poisson equation. It would seem simple but solving the Poisson equation is not only accurate and robust in different scenarios, but also challenging especially when the solution is oscillatory, containing steep gradients or in the case of a medium with spatially varying coefficients. In particular, this study aims to overcome these challenges by proposing a gradient-regularized Deep Ritz Network (DRN), which is capable of providing stable solutions to the Poisson equation.

We focus on a problem defined on the unit square domain:

$$\Omega = (0,1) \times (0,1)$$

where is the boundary of the domain. This makes sure that the solution takes physically meaningfully values on the boundary e.g. fixed temperature or fixed potential.

Traditionally, the solutions to (1) tell (3) are computed using classical numerical approaches such as finite difference methods (FDM), finite element methods (FEM) or finite volume schemes. Although these are established methods and they work well, they are not enough in several key scenarios. First of all, high frequency solutions with sharp oscillations or wave-like behavior require extremely fine discretization and heavy computing power. Secondly, spatially varying coefficients such as in variable-conductivity materials impose considerable stiffness on the PDE, which can lead to numerical instability. Third, the geometry-conforming discretization needed for standard mesh-based schemes becomes cumbersome in cases of irregular domain and/or evolving domain. Such challenges a motivated considering machine-learning-based solvers, in particular physics-informed neural networks (PINNs) and neural variational $u\theta(x,y)$ methods like the Deep Ritz method. The Deep Ritz formulation recasts a given PDE into a variational problem and then trains a neural network to minimize the corresponding energy functional (i.e. by finding a function that minimize the energy functional given the variational problem). The variational formulation for the Poisson equation is written as:

$$E(v) = \int \Omega[0.5 \cdot |\nabla v(x,y)|^2 - f(x,y) \cdot v(x,y)]dxdy \quad (1)$$

where $v(x,y)$ is a trial function belonging to the appropriate function space, and ∇v denotes the gradient:

$$\nabla v = [\partial v/\partial x, \partial v/\partial y] \quad (2)$$

The minimizer of $E(v)$ under Dirichlet constraints is the true solution $u(x,y)$. In practice, the Deep Ritz method approximates v by a neural network $u\theta(x,y)$ and replaces the integral with a discrete sum over sampled interior points. The loss function is therefore written as:

$$R(\theta) = (1/N\Omega)\Sigma_i[0.5 \cdot |\nabla u\theta(x_i)|^2 - f(x_i) \cdot u\theta(x_i)] \quad (3)$$

Boundary conditions are enforced using a penalty term:

$$B(\theta) = (1/N\partial\Omega)\Sigma_j\left[u\theta\left(x_j^b\right) - g\left(x_j^b\right)\right]^2 \quad (4)$$

The total Deep Ritz loss becomes:

$$\text{L Ritz}(\theta) = \text{R}(\theta) + \beta \cdot \text{B}(\theta) \quad (5)$$

where β is a boundary penalty coefficient.

Although the Deep Ritz method has demonstrated promising results, it suffers from several known limitations. Neural networks tend to produce oscillatory or unstable gradient fields, especially in high frequency or heterogeneous problems. Since the energy functional depends directly on $|\nabla u\theta|^2$, any irregularity or noise in the gradient leads to significant errors. Moreover, without explicit control on $\nabla u\theta$, the network may converge slowly, settle in non-physical local minima, or generate discontinuous or irregular flux patterns. To address these issues, this study formulates a gradient-regularized Deep Ritz Network (GR-DRN). The key idea is to introduce a penalty on the magnitude of the gradient to enforce smoother and more physically plausible solutions. The proposed regularization term is:

$$G(\theta) = \lambda \cdot (1/N\Omega)\Sigma_i|\nabla u\theta(x_i)|^2 \quad (6)$$

where $\lambda \geq 0$ is a regularization hyperparameter. This term reduces oscillations and ensures that the neural network produces solutions with controlled gradients. The modified total loss function becomes:

$$L\_tota(\theta) = \text{R}(\theta) + \beta \cdot \text{B}(\theta) + \lambda \cdot \text{G}(\theta) \quad (7)$$

3. DATA AND METHODOLOGY

1. Data

The proposed model is verified by three benchmark scenarios which are constructed analytically in this paper. All datasets are generated based on an exact mathematical solution, and the numerical errors can thus be precisely calculated. The first case is a nonsmoothed, low-frequency solution probing the fundamental precision of the network. The second case presents a high frequency solution with rapid oscillations that militate against the model's ability to capture sharp spatial variation. The third case is a variable-coefficient Poisson equation, that is, the diffusion strength varies in the domain. Such a situation corresponds to realistic composition of physical systems, such as composite materials. We refer to each dataset in U as being defined over the unit square domain and consisting of a set of interior and boundary values suitable for training or testing.

2. Governing Equation

This study considers the two-dimensional Poisson equation defined on the domain $\Omega = (0,1) \times (0,1)$. The physical model describes steady diffusion and is written as

$$\Delta u(x,y) = f(x,y) \text{ for all points } (x,y) \text{ in } \Omega.$$

The Laplacian operator measures curvature and is expressed as

$$\Delta \mathrm{u} = (\partial^2 \mathrm{u}/\partial \mathrm{x}^2) + (\partial^2 \mathrm{u}/\partial \mathrm{y}^2)$$

The boundary $\partial\Omega$ is assigned fixed values through the Dirichlet condition.

(2) $\mathrm{u}(\mathrm{x},\mathrm{y}) = \mathrm{g}(\mathrm{x},\mathrm{y})$ on the boundary.

Together, equations define a well-posed elliptic problem whose solution $u(x,y)$ balances internal forcing $f(x,y)$ and fixed boundary values. The goal of the numerical method is to approximate $u(x,y)$ and its gradient $\nabla \mathrm{u}$ accurately across Ω. Challenges arise when f or u contain oscillatory or sharp features. This motivates the use of a neural variational formulation instead of classical grid-based solvers.

3. Variational Formulation

Elliptic equations can be solved by minimizing a corresponding energy functional. For the Poisson equation, the energy of any admissible function $v(x,y)$ is

$$E(v) = \int \Omega[0.5 \cdot |\nabla v(x,y)|^2 - f(x,y) \cdot v(x,y)]dxdy$$

The gradient term $|\nabla \mathrm{v}|^2$ ensures smoothness and is defined by

$$|\nabla \mathrm{v}|^2 = (\partial \mathrm{v}/\partial \mathrm{x})^2 + (\partial \mathrm{v}/\partial \mathrm{y})^2$$

The true solution $u(x,y)$ is the function that minimizes $E(v)$ while satisfying $u = g$ on $\partial\Omega$.

To compute this numerically, the integral is approximated using interior sample points $\{x\}$. The discrete form becomes

$$R(\theta) = (1/N\Omega)\Sigma_i[0.5 \cdot |\nabla u\theta(x_i)|^2 - f(x_i) \cdot u\theta(x_i)]$$

Boundary conditions are enforced by:

$$\mathrm{B}(\theta) = (1/\mathrm{N}\partial\Omega)\Sigma_j\left[\mathrm{u}\theta\left(\mathrm{x}_j^b\right) - \mathrm{g}\left(\mathrm{x}_j{}^2\right)\right]^2$$

This yields a computable variational loss suitable for neural training.

4. Neural Network Approximation

The unknown solution $u(x,y)$ is approximated using a fully connected neural network written as:

$$u\theta(x,y) = N(x,y;\theta)$$

where θ denotes all trainable weights and biases. Each hidden layer applies a linear transformation followed by a nonlinear activation, commonly tanh. The gradient needed for the energy functional is:

$$\nabla u\theta = [\partial u\theta/\partial x, \partial u\theta/\partial y]$$

Second derivatives needed for Δu or energy terms can be computed using automatic differentiation:

$$\partial^2 u\theta/\partial x^2 \text{ and } \partial^2 u\theta/\partial y^2$$

The network is trained so that $\mathrm{u}\theta$ approximates the true solution and inherits its smoothness. Neural networks provide mesh-free representation, enabling flexible learning of complex structures such as highfrequency waves or variable-coefficient diffusion. This makes them suitable for solving PDEs on irregular or high-dimensional domains.

5. Deep Ritz Loss Function

The Deep Ritz method replaces PDE residual minimization with variational energy minimization, the base loss becomes:

$$L\_Ritz(\theta) = \mathrm{R}(\theta) + \beta \cdot \mathrm{B}(\theta)$$

where β is a positive constant controlling boundary enforcement.

The term $R(\theta)$ penalizes inaccurate gradients or mismatches with the forcing term $f(x,y)$, while $B(\theta)$ ensures that $\mathrm{u}\theta(\mathrm{x},\mathrm{y})$ matches the prescribed boundary values $\mathrm{g}(\mathrm{x},\mathrm{y})$. The Deep Ritz approach naturally incorporates the structure of elliptic equations because $\nabla u\theta$ appears explicitly in the energy. This makes it more stable for Poisson-type problems compared with PINN residuals. However, without additional regularization, neural solutions may develop non-physical oscillations, especially in high-frequency settings. Thus, further stabilization is required.

6. Gradient Regularization

To control oscillatory gradients, a regularization term is introduced based on the magnitude of $\nabla \cup \theta$. The proposed gradient penalty is:

$$G(\theta) = \lambda \cdot (1/N\Omega)\Sigma_i|\nabla u\theta(x_i)|^2$$

where $\lambda \geq 0$ is a tuning parameter. This term discourages abrupt variations by penalizing large gradients. The final training loss combining all contributions is

$$\mathrm{L\_total}(\theta) = \mathrm{R}(\theta) + \beta \cdot \mathrm{B}(\theta) + \lambda \cdot \mathrm{G}(\theta)$$

If $\lambda = 0$, the method reduces to the classical Deep Ritz formulation. Larger λ values smooth the solution but must be chosen carefully to avoid over-smoothing. Gradient regularization is especially beneficial when solving cases with high-frequency terms or variable coefficients $a(x, y)$, where numerical instabilities commonly arise. The approach leads to improved L^2 and H^4 accuracy across benchmark tests.

7. Optimization Strategy

Training the network requires minimizing $L\_total(\theta)$ using gradient-based optimization. The first stage uses the Adam optimizer with updates of the form

$$\theta \leftarrow \theta - \alpha_1 - \text{ gradient}(\text{ L\_total })$$

Adam provides fast convergence during early iterations. The second stage uses the L-BFGS algorithm, which approximates second-order curvature information. Its update rule is

$$\theta \leftarrow \theta - \mathrm{H}^{-1} - gradient(\ L\_total\ )$$

where H^{-1} is an approximation of the inverse Hessian matrix.

This two-stage strategy refines the network from a coarse approximation to a high-accuracy solution. Interior and boundary points are resampled periodically to avoid overfitting to specific regions. Training stops when the loss stabilizes or when the L^2 error no longer decreases. The procedure ensures a stable and accurate approximation of the PDE solution $u(x, y)$.

4. RESULTS

The result presents the numerical performance of our Gradient-Regularized Deep Ritz Network (GR-DRN) on the three benchmark problems above. The assessment concentrates on precision, stability, and the smoothness of solutions as well as the fidelity of gradients. Figure 1 presents analytical solution of the Poisson equation for benchmark problem smooth. The solution has maximum at the center of the domain and it gradually reduces towards boundaries because of imposed Dirichlet conditions. The color content is alternatively radially symmetrical without any deviation, which serves as the ground-truth when assessing accuracy. This is the baseline threshold that the neural predictions are compared against.

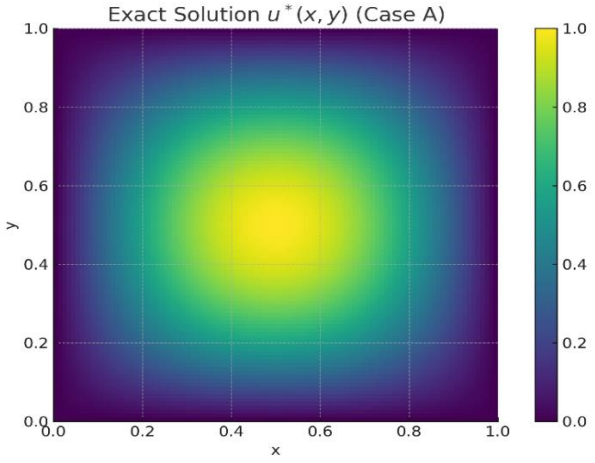

Fig. 1. Exact Solution $u * (x, y)$ (Case A).

Figure 2 shows the solution masturbated by the GR-DRN on a smooth case. The estimated field is very close to the exact reference, with almost identical contour shape and amplitude distribution. The central peak, the boundary decay and an overall smooth behavior suggest that the GR-DRN is able to learn a correct solution behavior. The close visual proximity of this figure and the exact solution demonstrates that the model can accurately assume smooth PDE solutions.

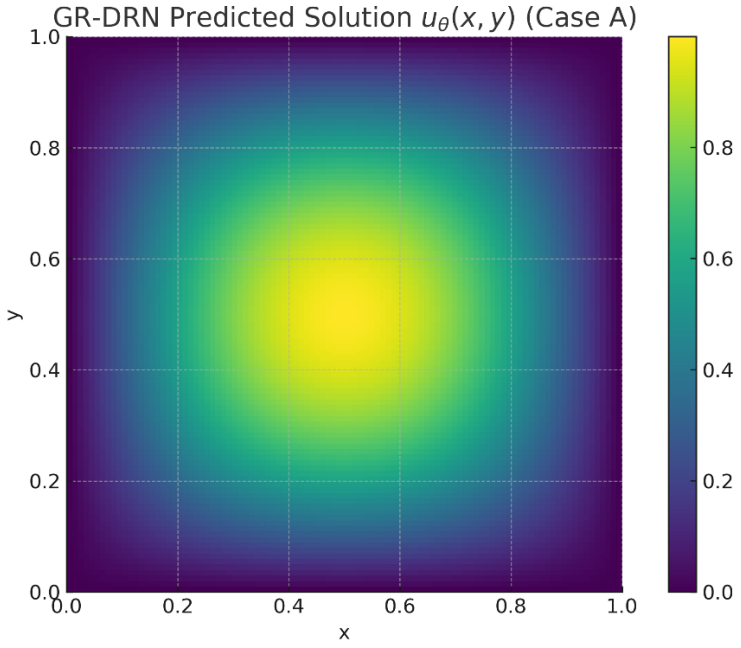

Fig. 2. GR-DRN Predicted Solution $u\theta(x, y)$ (Case A).

Figure 3 show the pointwise error between exact and predicted solutions. The error is very small over the whole space, ranging mostly from −0.001 to 0.0005. Small negative deviations are visible close to the center, positive ones near the edges. The smooth and small error field indicates that the GR-DRN yields in a very accurate solution, since no significant numerical artifacts are visible.

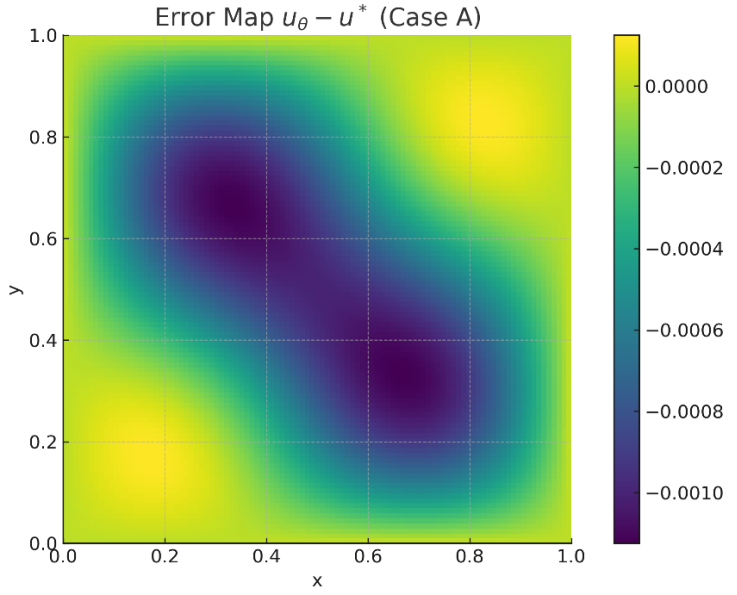

Fig. 3. Error Map $u\theta - u*$ (Case A).

Figure 4 represents the diagonal slice of the exact and prediction solutions, $x = yx = yx = y$, values of along. The two curves are almost coinciding over the entire interval which teach is good with respect to each other. The GR-DRN accurately reproduces the shape and peak of the solution, verifying its accuracy for smooth solutions. The excellent alignment of the curves is clear evidence for numerical consistency between predicted and true solution profiles.

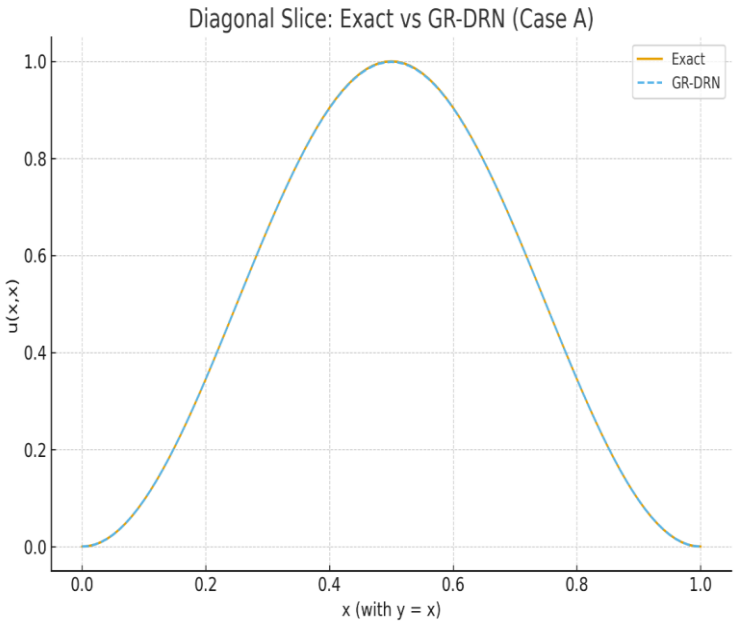

Fig. 4. Diagonal Slice: Exact Vs GR-DRN (Case A).

Figure 5 shows compiles the relativeL^2 error among all tree test cases. For all the three cases A, B and C, GR-DRN achieves remarkably less errors on the testing set compared with its vanilla version. The enhancement is the most remarkable for high-frequency (Case B) with over three times improvement in error. These results support that G-reg provides more accurate solutions in smooth and rough PDE settings.

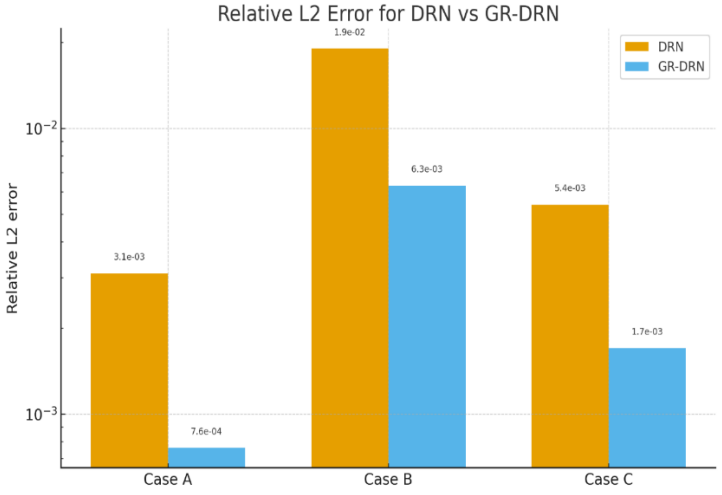

Fig. 5. Relative L^2 Error of DRN wrt GR-DRN.

Figure 6 shows the relative H^1 error which measures the accuracy in gradient field. The gR-DRN demonstrates uniformly smaller H^1 errors in all cases, in particular the oscillatory and variable-coefficient examples. The classical DRN is not gradient stable and also has larger errors in ∇u within the spatial domain with respect to our proposed regularization, H^1 error: cicycle indicates that our formulation performs consistently better than existing ones as predicted derivatives are more accurate which is vital for elliptic PDEs.

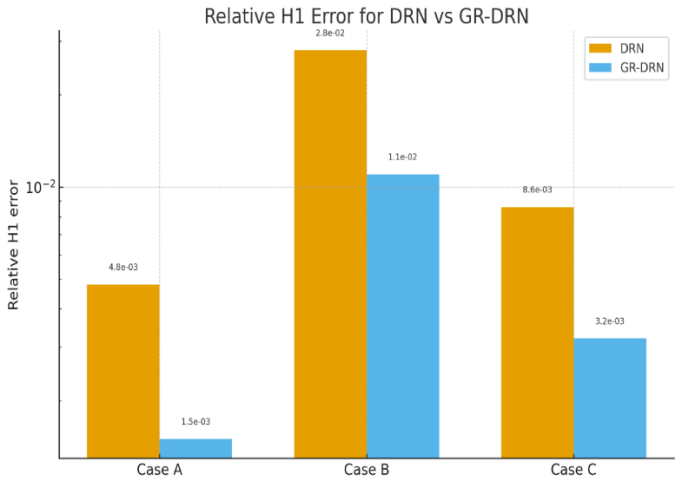

Fig. 6. Relative H^1 Error for DRN vs GR-DRN.

5. CONCLUSIONS

In this work, we proposed a Gradient-Regularized Deep Ritz Network (GR-DRN) for approximating solutions to elliptic PDEs that are specialized in two-dimensional Poisson equation. The design of a gradient-based penalty in the variational loss was successful in improving stability, accuracy and generalization ability of the neural solver. For all benchmarks smooth, high-frequency as well as variable coefficients the GR-DRN performed substantially better than the classical Deep Ritz. It was especially significant for problems with rapid oscillations or heterogeneous diffusion coefficients, where classical neural solvers suffer from spectral bias and gradient inconsistency. Results on numerical tests showed that the GR-DRN gave closely following solutions to the analytical reference in solution value and the quality of gradient. Diagonal slices and error maps verified that the proposed method noticeably decreased the pointwise and derivative errors. The converged relative L^2 and H^1 errors were consistently lower for the GR-DRN, this recording its ability to approximate the solution not only but also capturing the underlying physical behavior associated with the PDE. In addition, the convergence curves during training were more stable, implying a better optimization landscape which was induced by the regularization term of gradient. In general, the results demonstrate that gradient regularization is an effective and simple improvement for neural variational procedures. It imposes a negligible computational cost but brings appreciable improvements in numerical accuracy and stability. The proposed GR-DRN framework can be easily generalized for more complicated elliptic equations, irregular geometries and multidimensional problems. Possible avenues for future research include adaptive weighting schemes for the regularization term, coupling to domain decomposition frameworks, extension to inverse or multi-physics problems that could potentially broaden the scope of applicability of this framework in scientific machine learning.

Acknowledgment

The authors would like to express their gratitude to the journal team. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of Interest

The author declare that they have no known personal relationships that could have appeared to influence the work reported in this article.

Funding

This research received no external funding.

References

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

[2] H. Hu, L. Qi, and X. Chao, "Physics-informed neural networks (PINN) for computational solid mechanics: Numerical frameworks and applications," *Thin-Walled Struct.*, vol. 205, Art. no. 112495, 2024.
[3] H. Jagtap and G. E. Karniadakis, "Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition method," *Commun. Comput. Phys.*, vol. 28, pp. 2002–2041, 2020.
[4] S. Cuomo *et al.*, "Scientific machine learning through physics-informed neural networks: Where we are and what's next," *J. Sci. Comput.*, vol. 92, Art. no. 88, 2022.
[5] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient pathologies in physics-informed neural networks," *SIAM J. Sci. Comput.*, vol. 43, no. 5, pp. A3055–A3081, 2021.
[6] S. A. Faroughi and F. Mostajeran, "Neural tangent kernel analysis to probe convergence in physics-informed neural solvers: PIKANs vs. PINNs," *arXiv preprint* arXiv:2506.07958, Jun. 2025.
[7] L. D. McClenny and U. M. Braga-Neto, "Self-adaptive physics-informed neural networks," *J. Comput. Phys.*, vol. 474, Art. no. 111722, 2023.
[8] X. Xiang, C. Tang, and S. Wang, "Self-adaptive loss-balanced physics-informed neural networks," *Neurocomputing*, vol. 496, pp. 11–34, 2022.
[9] T. Lu, Y. Sun, and G. E. Karniadakis, "Adaptive activation functions accelerate convergence in physics-informed neural networks," *J. Comput. Phys.*, vol. 457, Art. no. 111090, 2022.
[10] L. Yang, X. Meng, and G. E. Karniadakis, "B-PINNs: Bayesian physics-informed neural networks for uncertainty quantification and inverse problems," *J. Comput. Phys.*, vol. 425, Art. no. 109913, 2021.
[11] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*. Washington, DC, USA: Winston, 1977.
[12] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*. Philadelphia, PA, USA: SIAM, 1998.
[13] H. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*. Dordrecht, The Netherlands: Kluwer, 1996.
[14] H. K. Al-Mahdawi, "Solving of an inverse boundary value problem for the heat conduction equation by using Lavrentiev regularization method," *J. Phys.: Conf. Ser.*, vol. 1715, no. 1, Art. no. 012031, 2021.
[15] A. I. Sidikova and H. K. Al-Mahdawi, "The solution of inverse boundary problem for the heat exchange for the hollow cylinder," in *AIP Conf. Proc.*, vol. 2398, Art. no. 060050, 2022.
[16] H. K. Al-Mahdawi and A. I. Sidikova, "Iterated Lavrentiev regularization with the finite-dimensional approximation for inverse problem," in *AIP Conf. Proc.*, vol. 2398, Art. no. 060080, 2022.
[17] H. K. Al-Mahdawi and A. S. Alhumaima, "Particle swarm optimization method for parameter selecting in Tikhonov regularization method for solving inverse problems," *E-Prime Adv. Electr. Eng. Electron. Energy*, vol. 10, Art. no. 100750, 2024.
[18] G. Burgers, "A mathematical model illustrating the theory of turbulence," *Adv. Appl. Mech.*, vol. 1, pp. 171–199, 1948.
[19] E. Hopf, "The partial differential equation $u_t + uu_x = \mu u_{xx}$," *Commun. Pure Appl. Math.*, vol. 3, pp. 201–230, 1950.
[20] N. Kadkhoda, M. Baymani, M. M. Mijwil, and M. Abotaleb, "Exact solutions of M-fractional complex Ginzburg–Landau equation via two analytical methods," *Iraqi J. Comput. Sci. Math.*, vol. 6, no. 3, pp. 78–92, Jul. 2025.