

Domain Decomposition-Based Architectures for Randomized Neural Networks

Yong Shang^[0009-0004-5671-7155], Alexander Heinlein^[0000-0003-1578-8104],
Siddhartha Mishra^[0000-0002-2665-5385], and Fei Wang^[0000-0002-9745-1195]

1 Introduction

Domain decomposition methods (DDMs) have shown to be very powerful numerical methods for a wide range of practical problems and are well-suited for implementation on parallel computers. By partitioning the computational domain into either overlapping or non-overlapping subdomains, local problems can be concurrently solved on various processors, thereby enhancing convergence properties and computational efficiency.

Recently, the integration of DDMs with neural networks to solve partial differential equations (PDEs) has become increasingly popular for accelerating computations and improving accuracy across a variety of deep learning models. These approaches can be categorized based on their utilization of machine learning algorithms [9].

Physics-informed neural networks (PINNs) [19] have been widely used to solve PDEs by using a loss function based on the residual terms. Building on PINNs, con-

Yong Shang
Inspur Group Co. Ltd, Shanghai Yunxi Technology Co. Ltd, Shanghai, China,
e-mail: shangyong01@inspur.com

Alexander Heinlein
Delft Institute of Applied Mathematics, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628 CD Delft, Netherlands,
e-mail: a.heinlein@tudelft.nl

Siddhartha Mishra
Seminar for Applied Mathematics, D-MATH, and ETH AI Center, ETH Zürich, Rämistrasse 101,
8092 Zürich, Switzerland. e-mail: smishra@sam.math.ethz.ch

Fei Wang
School of Mathematics and Statistics & State Key Laboratory of Multiphase Flow in Power Engineering, Xi'an Jiaotong University, Xi'an 710049, China. e-mail: feiwang.xjtu@xjtu.edu.cn. The work of this author was partially supported by National Natural Science Foundation of China (Grant No.12375175), Tianyuan Fund for Mathematics of the National Natural Science Foundation of China (Grant No. 12426105), and the Innovative Scientific Program of CNNC.

servative PINNs [11] employ a non-overlapping domain decomposition approach for conservation laws. Deep domain decomposition method (DeepDDM) [15] decomposes the problem using a classical fixed point Schwarz iteration. The idea of finite basis PINNs (FBPINNs) was proposed in [18], where a global function is formed based on an overlapping Schwarz domain decomposition. This approach was further developed in [5] by introducing additive, multiplicative, and hybrid iteration methods for training FBPINNs. Additionally, multilevel FBPINNs were developed in [6] by incorporating multiple levels of domain decompositions to address multiscale problems. Furthermore, [8] improved performance for time-dependent problems by combining multifidelity stacking PINNs with FBPINNs.

An alternative NN-based discretization method that leverages the variational formulation is the Deep Ritz method [23], which converts the problem into a minimization problem. Variational physics-informed neural networks (VPINNs) [12] use neural networks as their trial function space and Legendre polynomials as their test space. Then hp-VPINNs [13] improve accuracy by selecting non-overlapping polynomials as test functions on each subdomain.

Apart from the wide application of DDMs for PINNs and the deep Ritz method, domain decomposition has been adopted to other machine learning algorithms like randomized neural networks (RaNNs). For instance, Extreme Learning Machines (ELMs) [10] train single-hidden layer feedforward networks where the hidden layer parameters are randomly initialized and fixed, and only the output layer parameters are solved. ELMs can maintain universal approximation capability [17]. Combining local extreme learning machines with DDMs was introduced in [7], which improved accuracy and efficiency. For problems with complex geometry, a similar method was proposed in [2]. A combination of RaNNs and the discontinuous Galerkin approach was demonstrated in [21, 22]. Furthermore, the integration of RaNNs and the hybrid discontinuous Petrov-Galerkin method was proposed in [3].

In this study, RaNNs are generated on overlapping subdomains, and local window functions are utilized to construct a global solution. By solving a least-squares problem, we can obtain the solution of PDEs without additional penalty terms. Moreover, two-level extension is introduced to capture diverse scaling information. The paper is structured as follows: Section 2 introduces the basic ideas of RaNNs. Section 3 establishes the Schwarz domain decomposition-based RaNNs to solve PDEs. Section 4 extends this approach to a two-level domain decomposition. Section 5 presents numerical examples to demonstrate the accuracy of the proposed method. The last section concludes the paper with discussions.

2 Randomized neural networks with constraining operator

Let us consider a stationary diffusion problem on the computational domain $\Omega \subset \mathbb{R}^2$,

$$-\Delta u = f \quad \text{in } \Omega, \tag{1}$$

$$u = g \quad \text{on } \partial\Omega, \tag{2}$$

where u represents the solution, f is a given right-hand side function, and g is the function for Dirichlet boundary condition.

To solve (1) using a randomized neural network (RaNN), we aim to find $u_W(x)$ to approximate the exact solution $u(x)$. For instance, with a single-hidden layer feedforward neural network, $u_W(x)$ is given by:

$$u_W(x) = W \cdot \sigma(Ax + b), \quad (3)$$

where $A \in \mathbb{R}^{m \times 2}$ and $b \in \mathbb{R}^m$ are randomly generated elementwise from a given uniform distribution and fixed thereafter, with only $W \in \mathbb{R}^m$ requiring determination. The function σ acts as the activation function.

To enforce the boundary condition, instead of using the penalty method as [7, 2], we integrate a constraining operator C into the neural network (3), resulting in a solution ansatz.

$$\hat{u} = Cu_W = CW \cdot \sigma(Ax + b). \quad (4)$$

Here, C is designed to analytically satisfy the boundary condition (2). A thorough investigation into the exact satisfaction of boundary constraints across diverse problems was conducted in [14].

By selecting a set of collocation points $\{x_i\}_{i=1}^N \subset \Omega$, we need to find W such that

$$-W \cdot \Delta(C \cdot \sigma(Ax_i + b)) = f(x_i) \quad \forall i \in \{1, \dots, N\}. \quad (5)$$

The above equation gives

$$HW = F, \quad (6)$$

where $H \in \mathbb{R}^{N \times m}$ with $H_{i,k} = -\Delta(C \cdot \sigma(A_k x_i + b_k))$, and $F \in \mathbb{R}^N$ with $F_i = f(x_i)$.

In general, the matrix H is not square and symmetric, and therefore, it is solved using least-squares method.

3 Schwarz domain decomposition-based RaNNs

In overlapping Schwarz domain decomposition methods, the problem domain Ω is divided into J overlapping subdomains Ω_j . Here, we employ this idea to introduce an architecture for RaNNs following [18]. Therefore, on each subdomain, we define a RaNN

$$u_{W_j}(x) = W_j \cdot \sigma(A_j x + b_j). \quad (7)$$

The global solution ansatz $\hat{u}(x)$ is a combination of contributions from all networks,

$$\hat{u}(x) = C \sum_{j=1}^J \omega_j u_{W_j}(x) = C \sum_{j=1}^J \omega_j W_j \cdot \sigma(A_j x + b_j), \quad (8)$$

where C is the constraining operator enforcing the boundary condition, and $\{\omega_j\}_{j=1}^J$ are smooth window functions forming a partition of unity and confining the neural

networks to their respective subdomains

$$\text{supp}(\omega_j) \subset \Omega_j \quad \text{and} \quad \sum_{j=1}^J \omega_j \equiv 1 \quad \text{on } \Omega. \quad (9)$$

Thus, by considering collocation points $\{x_i\}_{i=1}^N \subset \Omega$ and substituting (8) into (1), we obtain a least-squares problem that can be solved to determine W_j for each neural network $u_{W_j}(x)$.

4 Two-level extension

Inspired by the multilevel approach in [6], we extend our architecture by adding a second level. In particular, coarse and fine levels of domain decomposition are constructed, with each level $1 \leq l \leq L$ defining a Schwarz domain decomposition of Ω into $J^{(l)}$ subdomains and L being the number of levels.

RaNNs are initialized with different parameters to capture solutions with varying scales. The approximated solution $\hat{u}(x)$ is expressed as:

$$\hat{u}(x) = \frac{C}{L} \sum_{l=1}^L \sum_{j=1}^{J^{(l)}} \omega_j^{(l)} u_{W_j}^{(l)}(x) = \frac{C}{L} \sum_{l=1}^L \sum_{j=1}^{J^{(l)}} \omega_j^{(l)} W_j^{(l)} \cdot \sigma(A_j^{(l)} x + b_j^{(l)}), \quad (10)$$

where we consider the case $L = 2$, C is the constraining operator, and $\{\omega_j^{(l)}\}$ are smooth window functions satisfying the partition of unity within each level l .

Similar to (8), the solution can be obtained by solving a least-squares problem. The above method can be easily extended to a multilevel approach ($L \geq 2$).

5 Numerical examples

Example 1 Let us consider a multi-scale Laplacian problem in the domain $\Omega = [0, 1]^2$ with the exact solution given by

$$u(x, y) = \frac{1}{n} \sum_{i=1}^n \sin(2^i \pi x) \sin(2^i \pi y)$$

where $n = 1, \dots, 5$ indicates the complexity of the problem.

First, we use one-level Schwarz domain decomposition RaNNs to solve the problem. Inputs for each network are normalized within their respective subdomains to the range $[-1, 1]$, and a uniform distribution $\mathcal{U}(-1, 1)$ is used to initialize the weights and bias parameters. The resulting least-squares problems is directly solved

via QR decomposition. These results are compared with those obtained using the multilevel FBPINN proposed in [6].

A uniform domain decomposition is established as follows:

$$\Omega_j^{(l)} = \begin{cases} [0.5 - \delta/2, 0.5 + \delta/2] & l = 1, \\ \left[\frac{(j-1) - \delta/2}{J^{(l)} - 1}, \frac{(j-1) + \delta/2}{J^{(l)} - 1} \right] & l \geq 1, \end{cases}$$

where $\delta > 1$ represents the overlap ratio ensuring a smooth and continuous solution across subdomain boundaries. Increasing δ encourages enhanced interaction among subdomains, thereby improving accuracy. The window functions and constraining operator align with those in [6], and the normalized L^1 test loss is computed approximately on $M = 350 \times 350$ uniformly-spaced test points by $\frac{1}{M} \sum_{i=1}^M \|\hat{u}(x_i) - u(x_i)\| / \tau$, where τ is the standard deviation of the true solutions set $\{u(x_i)\}_{i=1}^M$.

First, we implement a 16×16 Schwarz domain decomposition along with RaNNs featuring one hidden layer and 100 hidden units in each subdomain. Subsequently, we adjust the number of collocation points N and increase the overlap ratio δ to foster increased communication within overlapping regions. The results in Table 1 indicate that more data and communication lead to better results.

Table 1 Normalized L^1 test loss of a 16×16 domain decomposition.

n	N	$\delta = 2$	$\delta = 3$	$\delta = 4$
2	160×160	3.6e-7	2.3e-7	7.7e-10
	320×320	1.4e-11	3.3e-11	7.7e-13
3	160×160	3.4e-6	2.7e-7	6.1e-10
	320×320	2.3e-10	6.5e-11	7.7e-13
4	160×160	3.5e-4	4.2e-7	1.2e-9
	320×320	1.1e-8	6.4e-11	2.3e-12

For a fixed problem complexity of $n = 5$, we explore different models to solve it. By maintaining a consistent network structure and introducing additional subdomains, we enhance the method’s approximation capability. In Table 2, with 320×320 collocation points set and a constant $\delta = 4$, the ratio of the subdomain size and the overlap width remains the same, then we employ 8×8 , 16×16 , and 32×32 subdomains, demonstrating the trend of larger models yielding improved results. Additionally, we compare these outcomes with those obtained using the penalty method for addressing boundary conditions, with a penalty parameter set to 10, selected as the best option from $\{1, 10, 100\}$. We observe that RaNNs with a constraining operator approach achieve better accuracy.

In practice, enhancing model capacity proves to be an efficient approach as problem complexity escalates. In Table 3, a one-level domain decomposition with increased subdomains is used to solve higher complexity problems, resulting in enhanced accuracy consistency. In comparison to the weak scaling test in [6], our

Table 2 Normalized L^1 test loss of one-level domain decomposition for $n = 5$.

n	J	Constraining operator	Penalty method
5	8×8	6.5e-2	7.1e-1
5	16×16	8.3e-8	2.8e-7
5	32×32	1.3e-9	3.5e-8

results are approximately five orders of magnitude more accurate for the same problem complexity.

Table 3 Normalized L^1 test loss of one-level domain decomposition.

n	J	N	$\delta = 2$	$\delta = 3$	$\delta = 4$
1	2×2	20×20	3.6e-9	2.1e-10	1.4e-10
		40×40	3.9e-12	7.1e-13	5.0e-13
2	4×4	40×40	2.8e-6	2.9e-7	7.7e-11
		80×80	1.6e-10	5.6e-11	4.1e-12
3	8×8	80×80	7.1e-5	3.9e-7	8.1e-10
		160×160	1.2e-8	5.4e-11	7.8e-12
4	16×16	160×160	3.5e-4	4.2e-7	1.2e-9
		320×320	1.1e-8	6.4e-11	2.3e-12
5	32×32	320×320	1.2e-4	6.3e-7	1.3e-9

Example 2 Consider a problem featuring multiscale coefficients:

$$-\operatorname{div}(A^\varepsilon(x)\nabla u^\varepsilon(x)) = f \quad \text{in } \Omega, \quad (11)$$

$$u^\varepsilon(x) = g_D \quad \text{on } \partial\Omega, \quad (12)$$

where $A^\varepsilon(x_1, x_2)$ is a function characterized by two scales:

$$A^\varepsilon(x_1, x_2) = \frac{1}{8\pi^2} \begin{pmatrix} 2(2 + \cos(2\pi\frac{x_1}{\varepsilon}))^{-1} & 0 \\ 0 & 1 + \frac{1}{2}\cos(2\pi\frac{x_1}{\varepsilon}) \end{pmatrix}$$

with $\varepsilon = 0.05$, and f, g_D are given accordingly.

The solution to this problem can be expressed through the asymptotic expansion:

$$u^\varepsilon(x_1, x_2) = \sin(2\pi x_1) \sin(2\pi x_2) + \frac{\varepsilon}{2} \cos(2\pi x_1) \sin(2\pi x_2) \sin(2\pi\frac{x_1}{\varepsilon}).$$

First, we implement an 8×8 overlapping domain decomposition along with RaNNs featuring one hidden layer with 125 neurons in each subdomain. The activation

function is set as *sine*. Then we apply a two-level domain decomposition architecture, where the coarse level consists of a 4×4 domain decomposition, and the fine level comprises an 8×8 decomposition. Each subdomain is associated with a RaNN containing one hidden layer with 100 hidden units. The total number of neurons remains the same as in the one-level approach.

To enhance the performance in capturing different scale information, inspired by the construction of several subnetworks in [16] that act as different Fourier feature mappings to address multi-scale problems, we simplify it by implementing distinct weight parameter initialization in each network. Specifically, the hidden neurons in each network are organized into ten groups. The weight parameters A_j in each group are generated from a uniform distribution $\mathcal{U}(-r, r)$, with different values of r assigned to each group to create variability among them.

On the fine level, the values of r are chosen to be $(1, 2, \dots, 10)$, consistent with the one-level method with 8×8 domain decomposition. For the coarse level, we opt for $(1, 5, \dots, 45)$ to generate large-scale functions and effectively capture low-frequency components. A comparison of these two methods is shown in Table 4, illustrating the accuracy enhancement achieved by incorporating an additional level. The numerical solution, exact solution, and point-wise absolute error are displayed in Figure 1.

Table 4 Relative L^2 error of one-level and two-level domain decomposition.

m	Levels	N	$\delta = 2$	$\delta = 3$
125	[8]	160×160	1.72e-3	1.24e-4
		320×320	6.71e-4	4.08e-5
100	[4,8]	160×160	1.21e-4	2.33e-5
		320×320	1.05e-5	3.07e-6

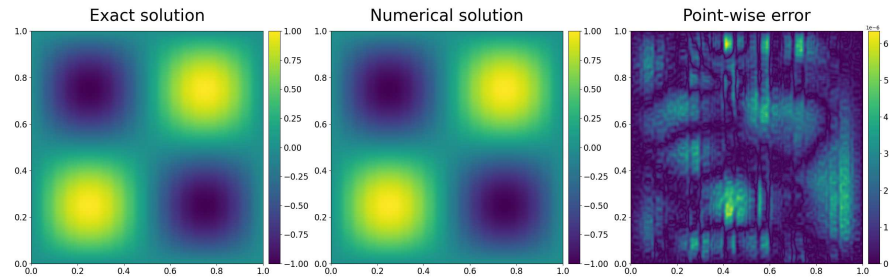


Fig. 1 Results of two-level ([4,8]) domain decomposition with RaNNs.

Remark 1 The results may strongly depend on the random initialization and choice of activation function. As shown, e.g., in [20], the resulting L^2 and H^1 errors may vary by orders of magnitude. While, for sake of brevity, we refrain from a detailed

investigation here, [1, 4] investigate some strategies for appropriately setting the weights and biases to construct an effective underlying approximating subspace, which can effectively mitigate the impact of randomization.

6 Conclusion

In this study, we present an approach to solving PDEs using randomized neural networks on overlapping subdomains connected by local window functions. By formulating the problem as a least-squares optimization, we successfully obtained accurate solutions to the PDEs. In addition, a two-level domain decomposition architecture is introduced to capture diverse scaling information, further enhancing the method's performance.

The combination of RaNNs with DDMs shows great potential for solving complex PDEs, providing a robust solution. However, the resulting linear system may have a large condition number, necessitating the design of suitable preconditioners and the use of iterative methods to effectively solve the least-squares problem. RaNNs with multi-level DDMs need further investigation, and numerical analysis of this approach needs exploration in the future.

References

1. Calabrò, F., Fabiani, G., Siettos, C.: Extreme learning machine collocation for the numerical solution of elliptic PDEs with sharp gradients. *Computer Methods in Applied Mechanics and Engineering* **387**, 114188 (2021)
2. Chen, J., Chi, X., E, W., Yang, Z.: Bridging traditional and machine learning-based algorithms for solving PDEs: the random feature method. *J Mach Learn* **1**(3), 268–298 (2022)
3. Dang, H., Wang, F.: Local randomized neural networks with hybridized discontinuous Petrov–Galerkin methods for Stokes–Darcy flows. *Physics of Fluids* **36**(8) (2024)
4. Dang, H., Wang, F., Jiang, S.: Adaptive growing randomized neural networks for solving partial differential equations. *arXiv preprint arXiv:2408.17225* (2024)
5. Dolean, V., Heinlein, A., Mishra, S., Moseley, B.: Finite basis physics-informed neural networks as a schwarz domain decomposition method. In: *International Conference on Domain Decomposition Methods*, pp. 165–172. Springer (2022)
6. Dolean, V., Heinlein, A., Mishra, S., Moseley, B.: Multilevel domain decomposition-based architectures for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering* **429**, 117116 (2024)
7. Dong, S., Li, Z.: Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations. *Computer Methods in Applied Mechanics and Engineering* **387**, 114129 (2021)
8. Heinlein, A., Howard, A.A., Beecroft, D., Stinis, P.: Multifidelity domain decomposition-based physics-informed neural networks for time-dependent problems. *arXiv preprint arXiv:2401.07888* (2024)
9. Heinlein, A., Klawonn, A., Lanser, M., Weber, J.: Combining machine learning and domain decomposition methods for the solution of partial differential equations—a review. *GAMM-Mitteilungen* **44**(1), e202100001 (2021)

10. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. *Neurocomputing* **70**(1-3), 489–501 (2006)
11. Jagtap, A.D., Kharazmi, E., Karniadakis, G.E.: Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering* **365**, 113028 (2020)
12. Kharazmi, E., Zhang, Z., Karniadakis, G.E.: Variational physics-informed neural networks for solving partial differential equations. *arXiv preprint arXiv:1912.00873* (2019)
13. Kharazmi, E., Zhang, Z., Karniadakis, G.E.: hp-vpinns: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering* **374**, 113547 (2021)
14. Leake, C., Mortari, D.: Deep theory of functional connections: A new method for estimating the solutions of partial differential equations. *Machine learning and knowledge extraction* **2**(1), 37–55 (2020)
15. Li, W., Xiang, X., Xu, Y.: Deep domain decomposition method: Elliptic problems. In: *Mathematical and Scientific Machine Learning*, pp. 269–286. PMLR (2020)
16. Li, X., Wu, J., Tai, X., Xu, J., Wang, Y.G.: Solving a class of multi-scale elliptic PDEs by Fourier-based mixed physics informed neural networks. *Journal of Computational Physics* **508**, 113012 (2024)
17. Liu, X., Lin, S., Fang, J., Xu, Z.: Is extreme learning machine feasible? a theoretical assessment (part i). *IEEE Transactions on Neural Networks and Learning Systems* **26**(1), 7–20 (2014)
18. Moseley, B., Markham, A., Nissen-Meyer, T.: Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics* **49**(4), 62 (2023)
19. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* **378**, 686–707 (2019)
20. Shang, Y., Wang, F.: Randomized neural networks with Petrov–Galerkin methods for solving linear elasticity and Navier–Stokes equations. *Journal of Engineering Mechanics* **150**(4), 04024010 (2024)
21. Sun, J., Dong, S., Wang, F.: Local randomized neural networks with discontinuous Galerkin methods for partial differential equations. *Journal of Computational and Applied Mathematics* **445**, 115830 (2024)
22. Sun, J., Wang, F.: Local randomized neural networks with discontinuous Galerkin methods for diffusive-viscous wave equation. *Computers & Mathematics with Applications* **154**, 128–137 (2024)
23. Yu, B., E, W.: The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics* **6**(1), 1–12 (2018)