From Finite Elements to Deep Learning: A Comparative Study of the Deep Ritz Method for Solving High-Dimensional PDEs

Yi He
Cuiying Honors College
Lanzhou University
320230929311

Abstract

Both the Finite Element Method (FEM) and the Deep Ritz Method (DRM) are rooted in the Ritz variational principle. However, while FEM is hindered by the "curse of dimensionality" due to its reliance on meshes, DRM represents a significant advancement by utilizing deep neural networks for mesh-free approximations. Our comparative study verifies this evolution: DRM matches FEM's accuracy in low-dimensional tasks but demonstrates superior scalability in high dimensions. Specifically, in a 10-dimensional benchmark where FEM failed due to memory exhaustion, DRM successfully achieved a relative error of 0.65%. This project is held on `https://github.com/hy-0003/DeepRitzMethod`

1 Introduction

Partial differential equations (PDEs) [2] serve as fundamental mathematical tools for characterizing physical phenomena and engineering problems. Over the past few decades, traditional numerical methods, most notably the Finite Element Method (FEM) [3], have established themselves as the gold standard for solving PDEs. FEM boasts a rigorous theoretical foundation, with well-established properties regarding consistency, stability, and convergence. However, despite its immense success, FEM encounters significant bottlenecks when dealing with high-dimensional problems or complex geometric domains. The cost of mesh generation increases drastically, and the method often suffers from the "curse of dimensionality," making computational costs prohibitive.

Recently, deep learning [4] has emerged as a powerful paradigm for scientific computing, benefited from its universal approximation capability. A prominent innovation in this field is the Deep Ritz Method (DRM), proposed by E and Yu [5]. This method employs deep neural networks to approximate the solution of PDEs and utilizes stochastic gradient descent for optimization, offering a mesh-free alternative to traditional solvers.

It is crucial to recognize that the Deep Ritz Method is not an isolated invention; rather, it shares the exact same mathematical root as the Finite Element Method. Both methods are founded on the classical Ritz method and the variational principle, which transform the problem of solving a PDE into a minimization problem of an energy functional. The fundamental difference lies in the choice of the trial function space: while FEM relies on mesh-based piecewise polynomials, DRM utilizes mesh-free deep neural networks. This shared lineage allows for a fascinating comparative perspective.

In this report, we focus on the Deep Ritz Method and its intrinsic connection to the Finite Element Method. Our goal is threefold: (1) to elucidate the theoretical framework of DRM; (2) to highlight its

This report is based on the open-source project [1] by Zeyu Jia, Dinghuai Zhang, and Zhengming Zou (Peking University). This document adopts their LaTeX template and follows a similar theoretical framework. All remaining content, including numerical experiments and analysis, is the independent work of the author.

"common origin" with FEM via the variational principle; and (3) to present a comparative analysis. We implement both methods to solve the classic Poisson equation:

$$\begin{cases} -\Delta u = f, & \forall x \in \Omega, \\ u = g, & \forall x \in \partial\Omega. \end{cases} \tag{1}$$

Through numerical experiments, we compare their performance in terms of accuracy, convergence behavior, and implementation complexity, demonstrating how these two methods, stemming from the same mathematical idea, diverge in application and effectiveness.

The structure of this report is organized as follows: Section 2 introduces the theoretical basis of the Ritz method and the variational principle, establishing the common ground for both algorithms. Section 3 details the architecture of the Deep Ritz Method and contrasts its implementation logic with FEM. Section 4 presents our numerical experiments and the comparative results. Finally, Section 5 concludes the report with a summary of our findings.

2 The Ritz Method

The Ritz method [2] is a classic numerical technique used to find approximate solutions to boundary value problems based on the variational principle. The core idea is to transform the problem of solving a partial differential equation (PDE) into an equivalent optimization problem: **finding a function $u(x)$ that minimizes an energy functional $I[u]$.**

Since the admissible function space is typically infinite-dimensional (e.g., a Hilbert space), which is computationally intractable, the Ritz method approximates the solution by projecting it onto a finite-dimensional subspace. Specifically, we assume the approximate solution $u_N(x)$ takes the form of a linear combination of pre-selected basis functions:

$$u_N(x) = \phi_0(x) + \sum_{i=1}^{N} c_i \phi_i(x), \tag{2}$$

where $\{\phi_i\}_{i=1}^N$ are linearly independent basis functions that vanish on the boundary (satisfying homogeneous boundary conditions), and ϕ_0 is a function chosen to satisfy the non-homogeneous boundary conditions.

By substituting this ansatz into the energy functional $I[u]$, the original functional is reduced to a multivariate function of the coefficients $\mathbf{c} = (c_1, c_2, \ldots, c_N)$:

$$J(c_1, \ldots, c_N) = I\left[\phi_0 + \sum_{i=1}^{N} c_i \phi_i\right]. \tag{3}$$

Consequently, the problem of finding the optimal function converts to finding the optimal constants c_i that minimize J. This is typically achieved by solving the system of algebraic equations derived from $\frac{\partial J}{\partial c_i} = 0$ for $i = 1, \ldots, N$. In classical spectral methods, the basis functions $\{\phi_i\}$ are often chosen as global smooth functions, such as trigonometric functions (Fourier series) or orthogonal polynomials:

$$\{\sin(kx), \cos(kx)\} \quad \text{or} \quad \{1, x, x^2, \ldots\}. \tag{4}$$

Although the theoretical basis set is of infinite dimension, in practice, we must truncate the series to a finite N to make the computation feasible. This process of seeking solutions within a finite-dimensional trial space is the fundamental strategy shared by the Ritz method, the Finite Element Method (FEM), and the Deep Ritz Method (DRM).

An Example Now we take the homogeneous Dirichlet boundary value problem of the Poisson equation (5) as an example.

$$\begin{cases} \Delta u = -f(x), & x \in \Omega, \\ u = 0, & x \in \partial\Omega. \end{cases} \tag{5}$$

The weak solution of (5) corresponding to the principle of least action is

$$\text{Find } u \in H_0(\Omega), \quad s.t. \quad I(u) = \min_{v \in H_0(\Omega)} I(v) \tag{6}$$

where $H_0(\Omega)$ is the set of admissible functions. Using variational method, we can prove that

$$I(v) = \int_\Omega \left(\frac{1}{2}|\nabla v(x)|^2 - f(x)v(x) \right) dx. \tag{7}$$

As a matter of fact, if we assume that v is an n-dimensional function, then

$$\delta I(v) = \int_\Omega \left(\frac{\partial v}{\partial x_1} \cdot \delta \frac{\partial v}{\partial x_1} + ... + \frac{\partial v}{\partial x_n} \cdot \delta \frac{\partial v}{\partial x_n} - f(x)\delta v(x) \right) dx. \tag{8}$$

Using the integration by parts method for several integrals,

$$\int_\Omega \frac{\partial v}{\partial x_i} \cdot \delta \frac{\partial v}{\partial x_i} dx = \frac{\partial v}{\partial x_i} \cdot \delta v \Big|_{\partial\Omega} - \int_\Omega \delta v \cdot \frac{\partial^2 v}{\partial x_i^2} dx = - \int_\Omega \delta v \cdot \frac{\partial^2 v}{\partial x_i^2}. \qquad 1 \le i \le n \tag{9}$$

Substituting (9) into (8) we can obtain the following equation.

$$\delta I(v) = - \int_\Omega \delta v (\Delta v + f(x)) dx = 0 \tag{10}$$

Because of the strongly convexity of the functional I in (7), the solution of (6) is unique. Hence the solution of (6) is identical to (5).

3 The Finite Element Method

While the general Ritz method provides the theoretical framework for minimizing the energy functional, its practical success relies heavily on the choice of the trial function space. The Finite Element Method [3] can be viewed as a specific realization of the Ritz method, where the infinite-dimensional solution space is approximated by a finite-dimensional subspace of piecewise polynomials. **The optimization process in FEM generally consists of three key step.**

1. Domain Discretization (Mesh Generation): First, the domain Ω is partitioned into a finite number of non-overlapping sub-domains T_k, such as triangles or tetrahedrons, forming a mesh $\mathcal{T}_h$:

$$\bar{\Omega} \approx \bigcup_{T_k \in \mathcal{T}_h} T_k. \tag{11}$$

2. Construction of Basis Functions: Instead of using global basis functions (like trigonometric functions), FEM constructs a finite-dimensional subspace V_h using local basis functions $\{\phi_i(x)\}_{i=1}^{N_h}$. These functions are typically defined as piecewise polynomials (e.g., Lagrange basis) with the property of compact support:

$$\phi_i(x_j) = \delta_{ij}, \tag{12}$$

where x_j are the nodes of the mesh and δ_{ij} is the Kronecker delta. Thus, any function $u_h \in V_h$ can be uniquely represented as:

$$u_h(x) = \sum_{j=1}^{N_h} c_j \phi_j(x), \tag{13}$$

where the coefficients c_j correspond directly to the nodal values of the solution.

3. Solving the Discrete Optimization Problem: Substituting the approximation u_h into the energy functional $I(u)$ defined in (7), the functional minimization problem is transformed into a quadratic programming problem with respect to the coefficients $\mathbf{c} = (c_1, \dots, c_{N_h})^T$:

$$\min_{u_h \in V_h} I(u_h) \iff \min_{\mathbf{c} \in \mathbb{R}^{N_h}} \left(\frac{1}{2} \mathbf{c}^T \mathbf{A} \mathbf{c} - \mathbf{b}^T \mathbf{c} \right). \tag{14}$$

By imposing the first-order optimality condition $\nabla_{\mathbf{c}} I = 0$, this minimization is equivalent to solving a sparse linear system of algebraic equations:

$$\mathbf{A}\mathbf{c} = \mathbf{b}, \tag{15}$$

where $\mathbf{A}$ is the stiffness matrix with entries $A_{ij} = \int_\Omega \nabla\phi_i \cdot \nabla\phi_j dx$, and $\mathbf{b}$ is the load vector with entries $b_i = \int_\Omega f\phi_i dx$.

4 Deep Ritz Method

Using the Ritz Method mentioned above, it is natural to think of solving partial differential equations with deep neural networks. Considering the partial differential equation

$$\Delta u(x) + f(x) = 0, \qquad x \in \Omega. \tag{16}$$

According to Ritz Method, what we need to do is to find the minimum of I, which is

$$\min_{u \in H} I(u), \tag{17}$$

where

$$I(u) = \int_{\Omega} \left(\frac{1}{2} |\nabla u(x)|^2 - f(x)u(x) \right) dx, \tag{18}$$

and H is the set of admissible function. Our main idea is to find a solution approximate to the real solution in favor of multi-layer neural networks' approximation property. And we will use the stochastic gradient descent algorithm to solve the optimization problem (17).

4.1 Building Trail Functions

We use a nonlinear transformation $x \to u_\theta(x)$ defined by deep neural networks to approximate function $u(x)$. Here θ denotes network parameters in our model. Similar to ResNet structure, we use several blocks to construct our networks, where each block consists of two linear transformations, two activation functions and one residual connection. The i-th block can be expressed as

$$s_i = \phi(W_{i2} \cdot \phi(W_{i1} \cdot s_{i_1} + b_{i1}) + b_{i2}) + s_{i-1}, \tag{19}$$

where s_i is the output of the i-th layer, $W_{i1}, W_{i2} \in R^{m \times m}, b_{i1}, b_{i2} \in R^m$ and ϕ is the activation function.

Because our partial differential equation involves the Laplacian transform, we hope that the second derivative of the function $u(x)$ is not a constant. To ensure the smoothness of function u_θ, we apply the non-linear function (20) as the activation function, instead of ReLU (rectified linear unit) function.

$$\phi(x) = \max\{x^3, 0\}. \tag{20}$$

The residual connection in [6] helps to avoid gradient vanishing problems. After several blocks, we adopt a linear transform to the final result. The architecture of residual connection can be viewed in 1. And the whole network can be expressed as

$$u_\theta(x) = a \cdot f_n(x) \circ \ldots \circ f_1(x) + b, \tag{21}$$

where $f_i(x)$ is the i-th block and $a \in R^m, b \in R$. Note that the input vector x is not necessarily m-dimensional. In order to handle this mismatch, we can pad x with a vector of zeros. In our model, we always assume $d < m$.

After building our trail functions, the rest of our work is to minimize the $I(u)$ in (18)

4.2 Boundary Conditions via Penalty Method

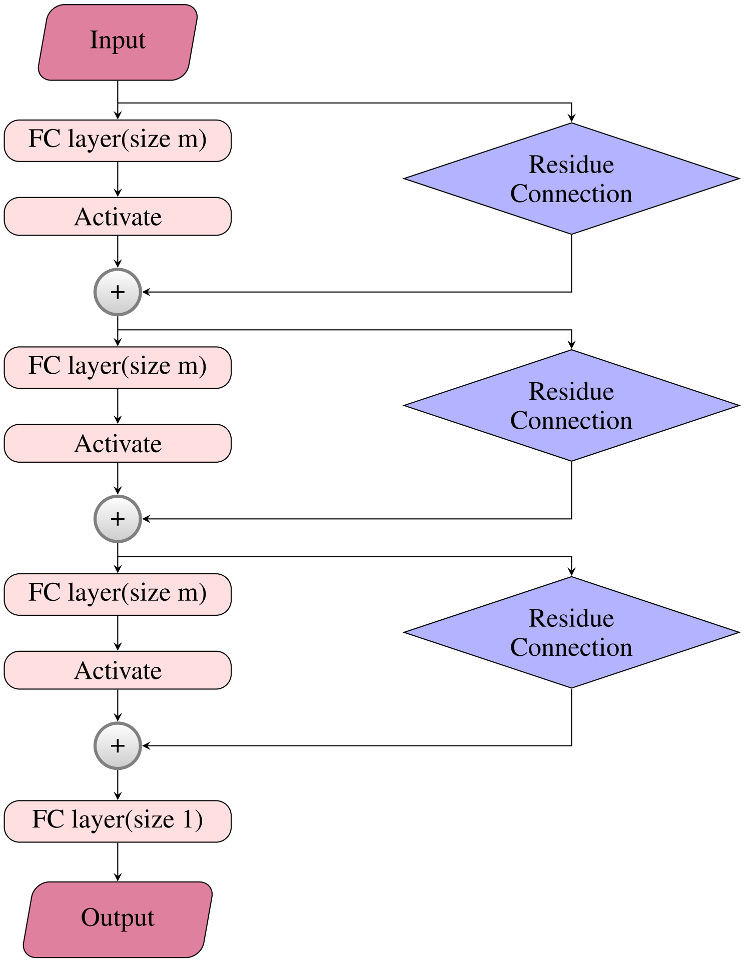
The variational problem (17) naturally corresponds to the Neumann boundary condition. However, for the Dirichlet boundary condition $u(x) = g(x)$ on $\partial\Omega$, the trial functions defined by neural networks do not automatically satisfy this constraint. To address this, the Deep Ritz Method employs the penalty method.

We modify the original energy functional by adding a boundary penalty term. The total loss function $\mathcal{L}(u)$ is defined as:

$$\mathcal{L}(u) = I(u) + \beta \int_{\partial\Omega} (u(x) - g(x))^2 ds, \tag{22}$$

where $\beta > 0$ is a hyperparameter representing the penalty weight. As β increases, the constraint $u|_{\partial\Omega} \approx g$ is enforced more strictly. The integral is taken over the surface of the boundary $\partial\Omega$. Consequently, our optimization goal shifts from minimizing $I(u)$ to minimizing the total loss $\mathcal{L}(u)$.

Figure 1: Network Structure of Deep Ritz Method



4.3 Monte Carlo Integration

The calculation of the total loss $\mathcal{L}(u)$ involves integrals over both the domain Ω and the boundary $\partial\Omega$. We define the boundary integrand as $h(x, \theta) = (u(x; \theta) - g(x))^2$. Similar to the domain integral, we adopt Monte Carlo integration for the boundary term.

By sampling N_{in} random points $\{x_i\}_{i=1}^{N_{in}}$ inside Ω and N_{bd} random points $\{y_j\}_{j=1}^{N_{bd}}$ on $\partial\Omega$, the total loss is approximated as:

$$\mathcal{L}(u) \approx \frac{|\Omega|}{N_{in}} \sum_{i=1}^{N_{in}} g(x_i, \theta) + \beta \frac{|\partial\Omega|}{N_{bd}} \sum_{j=1}^{N_{bd}} h(y_j, \theta), \tag{23}$$

where $|\Omega|$ is the volume of the domain and $|\partial\Omega|$ is the surface area (or measure) of the boundary. This unified Monte Carlo approach allows us to train the network using data sampled from the entire geometry.

4.4 Stochastic Gradient Descent and Infinite Data

To minimize the energy functional $I(u)$, we employ the Stochastic Gradient Descent (SGD) algorithm. A key insight of the Deep Ritz Method is the natural synergy between Monte Carlo integration and SGD.

In standard deep learning, SGD operates on a fixed dataset. However, in solving PDEs, we do not have a fixed dataset; instead, we have a continuous domain. At each training step k, we generate a

fresh batch of random points $\{x_{i,k}\}_{i=1}^{N_{batch}}$ inside Ω (and similarly on $\partial\Omega$ for boundary conditions). The parameter update rule is given by:

$$\theta^{k+1} = \theta^k - \eta \nabla_\theta \hat{\mathcal{L}}(\theta^k), \tag{24}$$

where $\hat{\mathcal{L}}$ denotes the empirical loss computed from the current batch of interior and boundary points.

This formulation reveals a profound advantage: since the training data $\{x_{i,k}\}$ is re-sampled at every iteration, the neural network never sees the same data twice. Consequently, the optimization process directly minimizes the population risk (the true integral) rather than the empirical risk on a fixed grid. This effectively prevents overfitting, as the network is forced to learn the solution over the entire continuous domain rather than memorizing specific grid points.

Remark about reference report [1]

We find it necessary to clarify certain descriptions regarding the numerical integration and sampling strategy presented in the reference report [1], which we believe are imprecise or potentially misleading in the context of Deep Ritz Method.

Terminology Correction: The reference report refers to the spatial discretization as "Euler Numerical Integration." In standard numerical analysis, Euler's method typically refers to an iterative method for solving ordinary differential equations (ODEs), rather than calculating spatial integrals. The method used here is theoretically closer to *Monte Carlo Integration* (when using random points) or *Riemann Sum* (when using a grid), rather than Euler integration.

The Pitfall of Fixed Grid: The reference report mentions using a fixed uniform grid with a step size of 0.001. We argue that this strategy contradicts the core advantage of the Deep Ritz Method. A fixed grid suffers from the *curse of dimensionality* (e.g., a 10-dimensional unit hypercube with step 0.001 would require 10^{30} points), making it infeasible for high-dimensional problems.

Risk of Overfitting: Furthermore, training a neural network on a fixed set of grid points poses a severe risk of **overfitting**. The network might essentially memorize the solution at these specific coordinates while failing to satisfy the equation in the gaps between points. As discussed in Section 4.2 and 4.3, the power of DRM lies in the stochastic nature of SGD with *freshly sampled* points at each iteration, which effectively simulates an infinite dataset and minimizes the population risk.

5 A Unified Perspective: From Linear to Non-linear Ritz Method

Although the Finite Element Method and the Deep Ritz Method appear distinct in implementation—one solving linear systems and the other training neural networks—they share the same mathematical foundation. Both can be viewed as specific numerical realizations of the classical Ritz method [7], differing primarily in their choice of the approximation ansatz.

The general Ritz method seeks an approximate solution u_h in a finite-dimensional parameter space $\mathcal{S}_\Lambda$, where Λ denotes the set of parameters:

$$u_h = \underset{v \in \mathcal{S}_\Lambda}{\operatorname{argmin}}\, I(v). \tag{25}$$

5.1 Linear Parametrization (FEM)

In FEM [8], the approximation space is constructed as a linear span of fixed basis functions $\{\phi_i\}_{i=1}^N$. The trial function is expressed as a linear combination:

$$u_h(x; \mathbf{c}) = \sum_{i=1}^{N} c_i \phi_i(x). \tag{26}$$

Here, the parameters Λ are the coefficients $\mathbf{c} = \{c_i\}$. Since the functional $I(u)$ is quadratic with respect to u, and u is linear with respect to $\mathbf{c}$, the resulting optimization problem is convex and quadratic, leading to a linear system $\mathbf{Ac} = \mathbf{b}$ [3]. While mathematically elegant, this approach is constrained by the grid quality and suffers from the curse of dimensionality.

5.2 Non-linear Parametrization (DRM)

In contrast, the Deep Ritz Method [5] constructs the approximation space using deep neural networks. Supported by the Universal Approximation Theorem [9, 10], a neural network can approximate any continuous function given sufficient width and depth. The trial function is parametrized as:

$$u_h(x;\theta) = \mathcal{N}(x;\theta), \tag{27}$$

where Λ corresponds to the network weights and biases θ. Unlike FEM, the dependence of u_h on θ is highly **non-linear**. Consequently, the minimization of $I(u_h)$ becomes a non-convex optimization problem.

5.3 Summary

The transition from FEM to DRM represents a shift from a linear, mesh-dependent ansatz to a non-linear, mesh-free ansatz. While FEM leverages the geometric orthogonality of basis functions for fast linear solving, DRM exploits the high-dimensional expressivity of neural networks. Thus, the Deep Ritz Method can be essentially interpreted as a **Non-linear Ritz Method**.

6 Numerical Results

6.1 Convergence Analysis of High-Frequency Solutions

To further evaluate the capability of the Deep Ritz Method in capturing high-frequency spatial oscillations, we consider the following Poisson equation:

$$\begin{cases} -\Delta u = 18\pi^2 \sin(3\pi x)\sin(3\pi y), & (x,y) \in \Omega = (0,1)^2, \\ u = 0, & (x,y) \in \partial\Omega. \end{cases} \tag{28}$$

The exact solution is given by $u(x,y) = \sin(3\pi x)\sin(3\pi y)$, which presents a multi-modal "egg carton" shape with higher spatial frequency compared to the previous examples.

Experiment 1 we set the boundary penalty parameter $\beta = 500$ to ensure strict Dirichlet boundary conditions. We compare the training snapshots of DRM against the standard Finite Element Method (FEM) and the exact solution at different iterations ($N = 1000$, $N = 3000$, and $N = 10000$). The results are visualized in Figure 2. The convergence behavior aligns with the Frequency Principle of neural networks:

Iteration 1000 (Top Row): The network learns the low-frequency mean profile first. As seen in the 3D plot (plasma colormap) and the 1D slice (red dotted line), the solution is overly smooth and fails to capture the peaks and valleys of the high-frequency sine waves.

Iteration 3000 (Middle Row): The network begins to capture the correct mode (shape) of the solution. However, there is a visible amplitude error; the DRM solution (red dotted line) underestimates the extrema compared to the exact solution (black line), indicating that the energy functional minimization is incomplete.

Iteration 10000 (Bottom Row): The method achieves convergence. The DRM solution visually matches both the Exact and FEM solutions. The 1D cross-section shows a precise overlap between the DRM prediction and the exact solution, confirming that the Deep Ritz Method can effectively resolve high-frequency oscillations with sufficient training.

Experiment 2 In the Deep Ritz Method, the boundary condition is imposed via a soft constraint penalty term $\beta \int_{\partial\Omega} u^2 ds$. The choice of β is critical for balancing the minimization of potential energy inside the domain against the satisfaction of Dirichlet boundary conditions. To investigate this, we conducted experiments with $\beta \in \{100, 500, 5000\}$ while fixing the number of iterations at $N = 10,000$. The results are presented in Figure 3.

Case $\beta = 100$ (Under-penalized): As shown in the left panel of Figure 3, a small penalty parameter fails to strictly enforce the boundary condition. Although the model converges to the general sine-wave shape, the boundary values are visibly curved and "floating" (non-zero) rather than flat. This indicates that the penalty weight is insufficient to force the solution to zero on $\partial\Omega$.

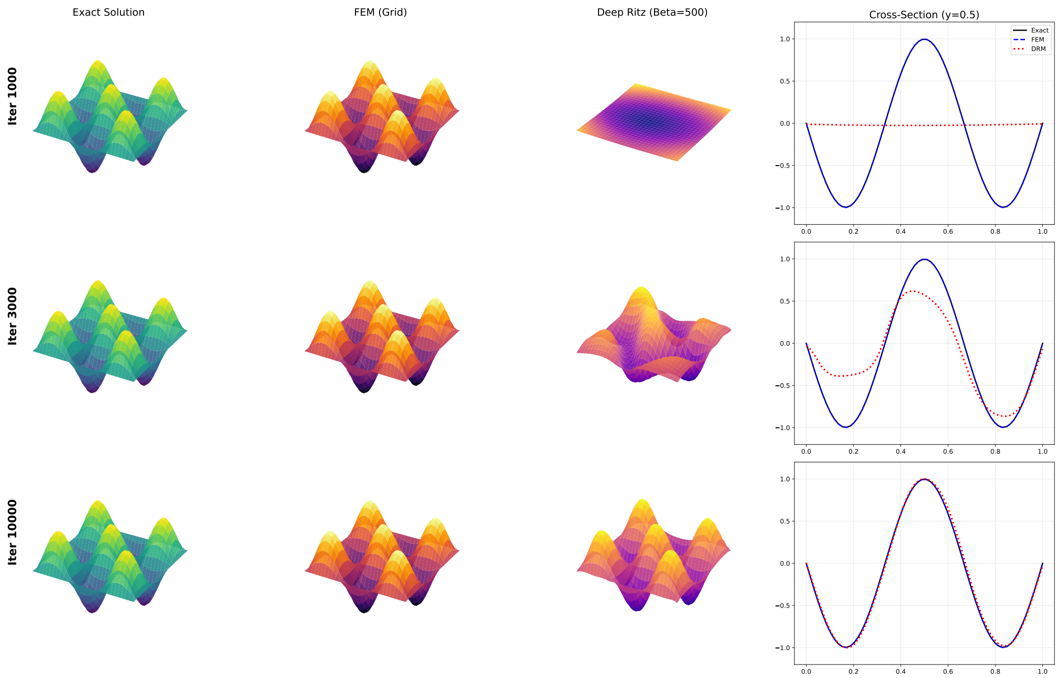


Figure 2: Evolution of the Deep Ritz Method solution during training. Left to Right: Exact Solution, FEM Solution, DRM Solution, and 1D Cross-section comparison. Top to Bottom: Snapshots at iteration 1000, 3000, and 10000.

Case $\beta = 500$ **(Balanced):** The middle panel demonstrates that $\beta = 500$ provides an optimal balance. The boundary conditions are strictly satisfied ($u|_{\partial\Omega} \approx 0$), and the interior solution correctly captures the high-frequency oscillations. This configuration yields the most accurate approximation of the exact solution.

Case $\beta = 5000$ **(Over-penalized):** Surprisingly, as shown in the right panel, an excessively large penalty parameter leads to a failure in convergence. The solution collapses to a nearly flat plane ($u \approx 0$ everywhere). This phenomenon occurs because the optimization landscape is dominated by the boundary penalty term. The network prioritizes minimizing the boundary error to zero immediately, getting stuck in a trivial local minimum where $u \equiv 0$, ignoring the interior source term f required to drive the solution away from zero.

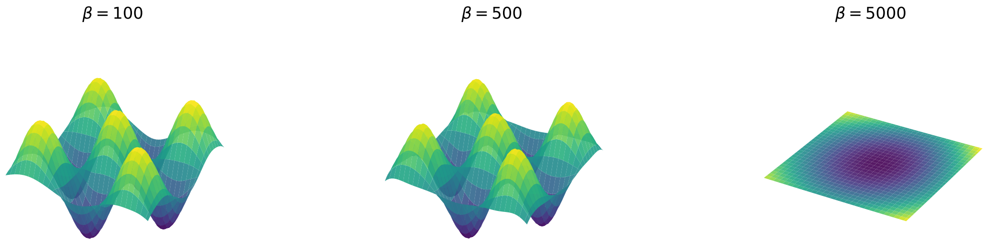


Figure 3: The impact of different penalty parameters β on the solution after 10,000 iterations. From left to right: $\beta = 100$ (Under-penalized), $\beta = 500$ (Balanced), and $\beta = 5000$ (Over-penalized).

6.2 Breaking the Curse of Dimensionality

A fundamental limitation of traditional grid-based methods is the "Curse of Dimensionality," where the computational cost and memory requirements grow exponentially with the dimension d. To demonstrate the mesh-free advantage of the Deep Ritz Method, we consider a high-dimensional Poisson equation on the hypercube $\Omega = (0, 1)^{10}$.

Problem Setup We consider the following 10-dimensional Poisson equation:

$$\begin{cases} -\Delta u(x) = \pi^2 \sum_{i=1}^{10} \sin(\pi x_i), & x \in \Omega \subset \mathbb{R}^{10}, \\ u(x) = \sum_{i=1}^{10} \sin(\pi x_i), & x \in \partial\Omega. \end{cases} \tag{29}$$

The exact solution is constructed as the superposition of sine waves in each dimension: $u(x) = \sum_{i=1}^{10} \sin(\pi x_i)$.

Limitations of Grid-Based Methods To empirically verify the computational limits, we attempted to solve this problem using a standard FEM solver with a uniform grid. Even with a very coarse grid resolution of $N = 10$ points per dimension, the total number of degrees of freedom (DOF) grows according to N^d.

Table 1 presents a comparison of the computational resources required for 4-dimensional and 10-dimensional problems under the same grid settings ($N = 10$).

Table 1: Comparison of Computational Requirements for FEM at Different Dimensions (Grid size $N = 10$ per dim)

Dim (d)	**Grid Points** (N)	**Total DOF** (N^d)	**Vector Memory** (Approx.)	**Matrix Memory** (Approx.)	**Status**
4	10	1.0×10^4	$\approx$ 0.00 GB	$\approx$ 0.00 GB	**Success**
10	10	1.0×10^{10}	74.51 GB	2346.93 GB	**Failed (OOM)**

As shown in Table 1, for the 4-dimensional case, the problem size (10^4 DOF) is trivial and can be solved instantly. However, for the 10-dimensional case, the number of unknowns explodes to 10^{10} (10 billion). Storing the solution vector alone requires approximately **74.51 GB** of RAM, and storing the sparse stiffness matrix requires an estimated **2.4 TB (2346.93 GB)**, which far exceeds the memory capacity of standard workstations (limit set to 32 GB). Consequently, the grid-based approach fails to initialize due to memory exhaustion (Out of Memory), illustrating the intractability of high-dimensional PDEs using traditional methods.

Deep Ritz Method: A Mesh-Free Solution In stark contrast to the memory exhaustion encountered by grid-based methods, the DRM successfully solves the 10-dimensional Poisson equation. By sampling points randomly from the high-dimensional domain Ω, DRM maintains a linear computational complexity with respect to the dimension d, effectively breaking the "Curse of Dimensionality." Since directly visualizing a 10-dimensional hypersurface is impossible, we evaluate the quality of the solution by examining a 2D cross-section. We vary the first two dimensions $x_1, x_2 \in [0, 1]$ while fixing the remaining eight dimensions at the center of the domain ($x_3 = \cdots = x_{10} = 0.5$).

Figure 4 presents the comparison between the exact solution and the DRM prediction on this slice, as well as the training convergence history.

As shown in the right panel of Figure 4, the relative L_2 error decreases rapidly during the initial training phase. Quantitative monitoring of the training process reveals that the model achieves high accuracy with stable convergence. Specifically, after 9,000 iterations, the relative error consistently fluctuates around 1%. The model achieves a minimum error of **0.65%** at iteration 10,000 and maintains this high precision in subsequent checks (e.g., 0.65% at iteration 11,500 and 12,500). Despite the high dimensionality, DRM requires only negligible memory (a few MBs for network weights) compared to the terabytes required by FEM, demonstrating its capability to solve problems that are computationally intractable for traditional methods.

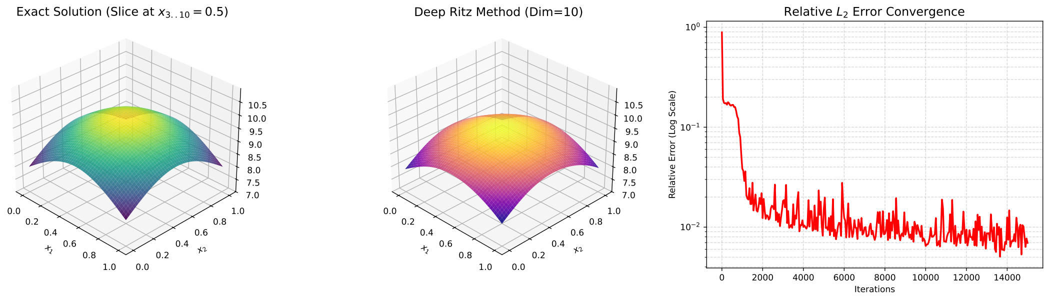


Figure 4: Results of the Deep Ritz Method for the 10-dimensional Poisson equation. Left: The exact solution on a 2D slice (varying x_1, x_2, fixed $x_{3...10} = 0.5$). Middle: The DRM predicted solution on the same slice, showing accurate capture of the high-dimensional structure. Right: The convergence of relative L_2 error, which stabilizes around $0.7\% \sim 1.0\%$.

7 Discussion: Recent Advancements and Applications

While this report focuses on the foundational capabilities of the Deep Ritz Method (DRM) in solving high-dimensional Poisson equations, the method has evolved significantly since its inception. Recent research has addressed key algorithmic limitations and expanded its application to complex physical systems.

7.1 Algorithmic Innovations

Deep Nitsche Method A known challenge in the original DRM formulation is the imposition of Dirichlet boundary conditions via penalty terms [11]. The convergence is often sensitive to the choice of the penalty parameter λ. To address this, Liao and Ming proposed the *Deep Nitsche Method* [12], which incorporates Nitsche's variational method into the learning framework. This approach provides a consistent and robust mechanism for enforcing boundary conditions without the instability associated with large penalty factors.

Overcoming Spectral Bias Neural networks generally exhibit "spectral bias," meaning they tend to learn low-frequency functions quickly but struggle to capture high-frequency details. This limits the accuracy of DRM for problems with multi-scale features. Recent works have integrated *Fourier Feature Mapping* [13] into the DRM architecture. By mapping input coordinates to a higher-dimensional feature space using sinusoidal functions, the network's ability to resolve high-frequency oscillations is significantly enhanced.

7.2 Applications in High-Dimensional Physics

The most profound impact of DRM and its variants has been in fields where the "Curse of Dimensionality" traditionally blocks progress.

Quantum Mechanics: The method has been successfully adapted to solve the high-dimensional stationary Schrödinger equation. Han et al. [14] demonstrated that deep learning-based energy minimization can accurately compute the ground state energies of many-electron systems, a problem that is computationally intractable for standard grid-based methods.

Material Science: DRM has also been applied to non-convex energy minimization problems in material science, such as modeling the microstructure evolution in martensitic phase transformations, where the solution landscape is highly complex and riddled with local minima.

8 Conclusion and Discussion

Summary and Insights Our systematic evaluation reveals that the Deep Ritz Method (DRM) represents a fundamental paradigm shift in solving Partial Differential Equations, particularly for high-dimensional problems. Experimentally, we identified that the boundary penalty parameter β is a critical factor; an under-penalized model ($\beta = 100$) leads to boundary drift, while an over-penalized model ($\beta = 5000$) causes optimization collapse. However, with a balanced configuration ($\beta = 500$), DRM demonstrated remarkable superiority over grid-based methods. In the 10-dimensional Poisson problem, where the standard Finite Element Method failed due to tractable memory requirements (estimated at 2.4 TB), DRM successfully achieved a stable relative error of **0.65%** with negligible computational cost. This confirms that DRM effectively transforms the exponential complexity $O(N^d)$ of traditional methods into a linear complexity $O(d)$, making it an indispensable tool for high-dimensional physics.

Limitations and Future Works Despite its success, the standard DRM relies on "soft" constraints, which introduces optimization challenges in complex energy landscapes. The sensitivity of the penalty term suggests that the method requires careful hyperparameter tuning to avoid local minima. To address this, future research should integrate the *Deep Nitsche Method* [12] to enforce boundary conditions through a variationally consistent framework, and employ *Fourier Feature Mappings* [13] to mitigate spectral bias in high-frequency problems. Furthermore, recognizing the fragmentation of resources in this rapidly evolving field, we plan to initiate an open-source project, **Awesome-DeepRitzMethod**, to curate state-of-the-art papers, codebases, and benchmarks, thereby facilitating future reproducibility and development in the scientific machine learning community.

Code Availability

The source code and datasets generated during the current study are available in the GitHub repository: `https://github.com/hy-0003/DeepRitzMethod`. The repository includes all scripts required to reproduce the numerical experiments and the high-dimensional results presented in this report.

Acknowledgements

First and foremost, I extend my sincere gratitude to Zeyu Jia, Dinghuai Zhang, and Zhengming Zou from the School of Mathematical Sciences at Peking University. Their open-source LaTeX template, along with their insightful report and codebase, provided a robust foundation and valuable reference for the implementation of the Deep Ritz Method presented in this work.

I am deeply indebted to Professor Li, my instructor for the *Finite Element Method* course. It was his guidance that introduced me to the rigorous world of FEM and provided the initial impetus for this project, encouraging me to bridge the gap between classical numerical methods and modern deep learning.

Finally, on a personal note, I feel incredibly honored to engage with such cutting-edge mathematics at the age of twenty. This journey extends beyond this report; I am currently conducting research on the **Deep BSDE Method** in a related domain and plan to release a preprint in the coming semester. Through these efforts, I believe I am offering a small, yet significant, response to the scientific curiosity and dreams I have cherished since childhood.

References

[1] Zeyu Jia, Dinghuai Zhang, and Zhengming Zou. Deepritzmethod: Source code and report. `https://github.com/ZeyuJia/DeepRitzMethod`, 2018. GitHub repository.

[2] L.C. Evans and American Mathematical Society. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 1998.

[3] Susanne C Brenner and L Ridgway Scott. *The mathematical theory of finite element methods*, volume 15. Springer Science & Business Media, 2008.

[4] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[5] Bing Yu et al. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *arXiv preprint arXiv:1710.00211*, 2017.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Walther Ritz. Über eine neue methode zur lösung gewisser variationsprobleme der mathematischen physik. *Journal für die reine und angewandte Mathematik*, 135:1–61, 1909.

[8] Philippe G Ciarlet. *The finite element method for elliptic problems*. SIAM, 2002.

[9] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[10] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[11] Weinan E and Bing Yu. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[12] Yulei Liao and Pingbing Ming. Deep nitsche method: Deep ritz method with essential boundary conditions. *Communications in Computational Physics*, 29(5):1365–1384, 2021. Preprint available as arXiv:1912.01309 (2019).

[13] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547, 2020.

[14] Jiequn Han, Linfeng Zhang, and Weinan E. Solving many-electron schrödinger equation using deep neural networks. *Journal of Computational Physics*, 399:108929, 2019.