# The Deep Ritz Method

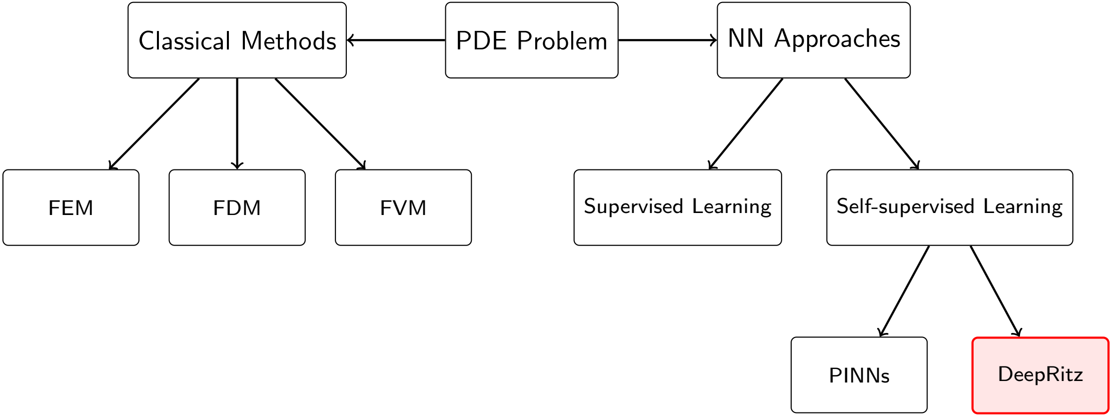
## Deep Learning-Based Numerical Algorithm for Solving Variational Problems

Vincent Scholz & Dr. Yaohua Zang

Professorship for Data-driven Materials Modeling

June 4, 2025

# Neural Network Approaches for Solving PDEs



# Repeat: From Strong to Weak Form: Poisson Equation

**Strong form:**

$$-\Delta u(x) = f(x) \quad \text{in } \Omega, \qquad u(x) = 0 \quad \text{on } \partial\Omega$$

**Step 1: Multiply by test function $v(x) \in H_0^1(\Omega)$**

$$-\int_\Omega \Delta u(x) \cdot v(x)\, dx = \int_\Omega f(x) v(x)\, dx$$

**Step 2: Apply integration by parts (or Green's identity)**

$$\int_\Omega \nabla u(x) \cdot \nabla v(x)\, dx - \int_{\partial\Omega} v(x) \frac{\partial u}{\partial n}(x)\, ds = \int_\Omega f(x) v(x)\, dx$$

**Step 3: Use boundary condition $v(x) = 0$ on $\partial\Omega$**

$$\Rightarrow \int_\Omega \nabla u(x) \cdot \nabla v(x)\, dx = \int_\Omega f(x) v(x)\, dx$$

**Result: Weak form of the Poisson problem**

$$\text{Find } u \in H_0^1(\Omega) \text{ such that } \int_\Omega \nabla u \cdot \nabla v\, dx = \int_\Omega f v\, dx \quad \forall v \in H_0^1(\Omega)$$

# Table of Contents

# Table of Contents

# Solving Equations via Optimization

Many equations can be reformulated as minimization problems.
**Example:** If $F(u) = \nabla J(u)$, then solving $F(u) = 0$ is equivalent to

$$u = \arg\min_{v} J(v)$$

## Key Insight

If we can define a suitable scalar loss $J(u)$, we can solve complex problems using optimization instead of direct equation solving.

# From PDE to Energy Functional (Intuition)

**Poisson equation:** $-\Delta u = f \quad \text{in } \Omega, \text{ with } u = 0 \text{ on } \partial\Omega$

Instead of solving the PDE directly, we ask:

- Can we find an energy functional $\mathcal{E}(u)$ whose minimizer satisfies this PDE?

**Heuristic reasoning:**

- Multiply both sides of the PDE by a test function $v$, integrate, and apply integration by parts
- This leads to a weak form: $\int_\Omega \nabla u \cdot \nabla v = \int_\Omega fv$
- Recognize this as the variation of:

$$\mathcal{E}(u) = \int_\Omega \left( \frac{1}{2} |\nabla u|^2 - fu \right) dx$$

## Key Insight

Minimizing this energy is equivalent to solving the PDE. BCs have to be enforced separately.

# Dirichlet's Principle

The energy functional:

$$\mathcal{E}(u) = \int_\Omega \left( \frac{1}{2} |\nabla u(x)|^2 - f(x)u(x) \right) dx$$

has a unique minimizer $u \in H_0^1(\Omega)$, which solves the Poisson equation.

## Key Insight

We can solve PDEs by minimizing an energy instead of discretizing derivatives — this is the basis of the **DeepRitz method**. BCs have to be enforced separately.

# Summary: Variational Formulation

**Problem:** Find $u$ such that $-\Delta u = f$, $u = 0$ on $\partial\Omega$

**Equivalent task:** Find $u$ minimizing

$$\mathcal{E}(u) = \int_\Omega \left( \frac{1}{2} |\nabla u|^2 - fu \right) dx$$

**Benefits:**

- No need to discretize the differential operator
- Can represent $u(x)$ as a neural network
- Apply standard optimization techniques (e.g., gradient descent)

**Key Insight**

This variational view is what DeepRitz leverages to solve PDEs using neural networks.

# Table of Contents

# The Basic Idea of Deep Ritz

**Key Innovation:**

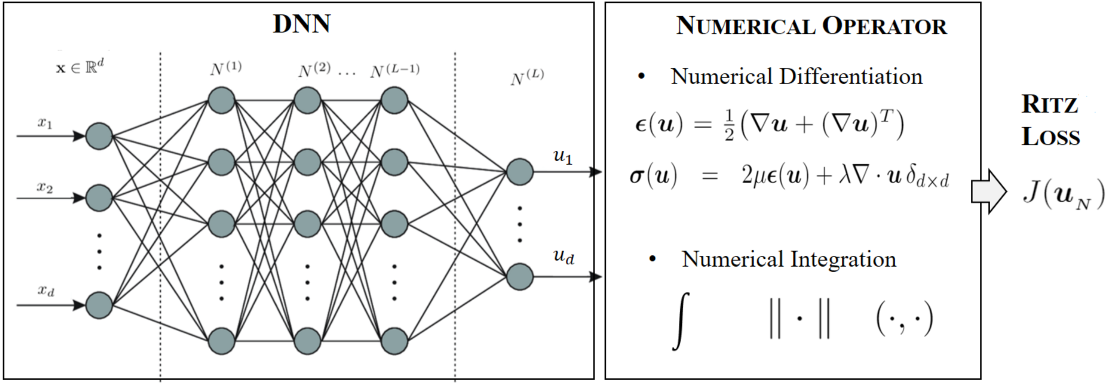
- Use neural networks to approximate solutions
- Leverage variational formulations (not strong form like PINNs)
- Frame PDE solving as energy minimization

**Approach:**

$$\min_{\theta} I(u_\theta)$$

where $u_\theta$ is a neural network

# Overview DeepRitz



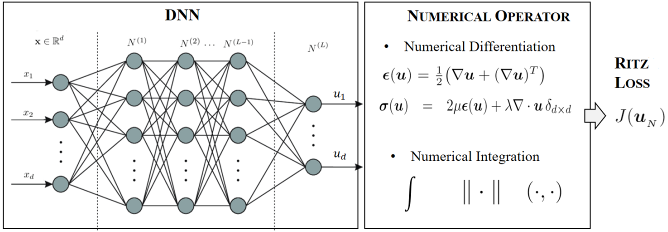
# Step 1: Neural Network Approximation

Approximate the solution $u(x)$ with a feedforward neural network:

$$u_\theta(x) = (\mathcal{N}^L \circ \mathcal{N}^{L-1} \circ \cdots \circ \mathcal{N}^0)(x)$$

where:

- $\mathcal{N}^l$ denotes the $l$-th layer operation
- $\theta$ includes all weights and biases



## Step 2: Loss Function Formulation

The total loss consists of two components:

### 1. Boundary Condition Loss

$$L_{bd}(\theta) = \frac{1}{N_{bd}} \sum_{i=1}^{N_{bd}} |u_\theta(x_i) - g(x_i)|^2$$

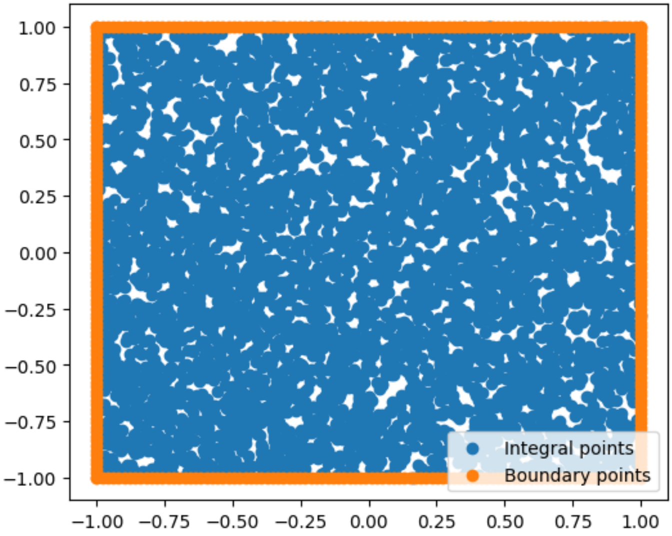
where $\{x_i\}_{i=1}^{N_{bd}}$ are boundary points

### 2. Energy Loss (PDE Term)

$$L_{PDE}(\theta) = \sum_{i=1}^{N_{int}} w_i \left( \frac{1}{2} |\nabla u_\theta(x_i)|^2 - f(x_i) u_\theta(x_i) \right)$$

where $\{x_i, w_i\}_{i=1}^{N_{int}}$ are integration points and weights

# Step 2: Loss Function Formulation



# Total Loss Function and Training

## Combined Loss

$$L(\theta) = \omega_{PDE} \cdot L_{PDE}(\theta) + \omega_{bd} \cdot L_{bd}(\theta)$$

**Training Process:**

- Use standard optimizers (SGD, Adam)
- Update rule: $\theta_{t+1} = \theta_t - l_r \nabla_\theta L(\theta_t)$
- Train over multiple epochs with batch sampling

**Special Activation:** $\sigma(x) = \max\{x^3, 0\}$ (cubic ReLU for smoothness)

# Table of Contents

# Advantages of Deep Ritz

## 1. Efficient Training & Faster Convergence

- Minimizes scalar energy functional
- Simpler optimization landscape
- More stable than pointwise residual methods
- Global measure of error

## 2. Supports Weak Solutions

- Natural for non-smooth solutions
- Works in Sobolev spaces
- Handles singularities well
- No classical differentiability required

# Limitations of Deep Ritz

## 1. Requires Many Integration Points

- Accurate energy evaluation needs dense sampling
- Computationally expensive in high dimensions
- Monte Carlo variance can slow convergence
- Complex geometries are challenging

## 2. Limited to Variational PDEs

- Only works with energy formulations
- Excludes many time-dependent PDEs
- No hyperbolic equations (wave equation)
- More restrictive than PINNs

# When to Use Deep Ritz?

**Suitable When:**

- PDE has known variational form
- Solution may be non-smooth
- Want faster/stable convergence
- Working with elliptic PDEs
- Need weak solution concepts

**Less Suitable When:**

- PDEs lack energy functional
- High-dimensional problems
- Limited computational resources
- Time-dependent evolution
- Hyperbolic equations

# Table of Contents

# Summary: Comparison of PDE Solvers

| Aspect | Classical Methods | PINNs | DeepRitz |
|---|---|---|---|
| Formulation | Discretize PDE | Loss on PDE residual | Minimize energy functional |
| Boundary Conditions | Strong/weak imposition | Soft/hard constraints | Naturally incorporated |
| Mesh Dependency | Requires mesh | No mesh | No mesh |
| High-Dim Performance | Poor (curse of dim.) | Moderate | Often superior |
| Requirements | Any PDE type | PDE residual available | Variational form available |

# References


E, Weinan and Yu, Bing.
The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems.
*arXiv preprint arXiv:1710.00211*, 2017.

Liao, Qinyi and Lin, Tianyi.
Deep Ritz Revisited.
*OpenReview*, 2019.

Partial Differential Equations 2: Variational Methods.
Technical University of Munich, 2019.