ETH zürich Caltech

Github: Paper:

Solving Poisson Equations using Neural Walk-on-Spheres

Hong Chul Nam[1*], Julius Berner[2*], Anima Anandkumar[2]
[1]ETH Zurich; [2]California Institute of Technology; [*]Equal Contribution

Summary

Goal: Solve Poisson equations of the form

$$\begin{cases} \Delta u = f, & \text{on } \Omega \\ u = g, & \text{on } \partial\Omega \end{cases}$$

For simplicity, we assume $f = 0$, and define $\xi \sim \mathcal{U}(\Omega)$

Standard Approach:

$$\mathcal{L}_{\text{PINN}}[\theta] \coloneqq \mathbb{E}\left[\left(\Delta u_\theta(\xi) - f(\xi)\right)^2\right] + \beta \mathcal{L}_{\text{boundary}}[u_\theta]$$

☹ **Requires high-order derivatives and tuning β**

SDE Representation:

$$\mathcal{L}_{\text{SDE}}[\theta] \coloneqq \mathbb{E}\left[\left(u_\theta(\xi) - g\left(X_\tau^\xi\right)\right)^2\right]$$

where we assume

$$\tau \coloneqq \{t: X_t^\xi \notin \Omega\} \text{ (exit time)}$$

$$X_{t+\Delta t}^\xi \approx X_t^\xi + \sqrt{2}\Delta W_t \text{ (Brownian motion)}$$

$$X_0^\xi = \xi \text{ (random initial point)}$$

$$\Delta W_t \sim \mathcal{N}(0, \Delta t \boldsymbol{I}) \text{ (Gaussian increments)}$$

☹ **Slow convergence until reaching boundary**

NWoS:

Leveraging the **SDE representation** and **Walk-on-Spheres** method, we **recursively solve** Poisson equations on spheres inside the domain

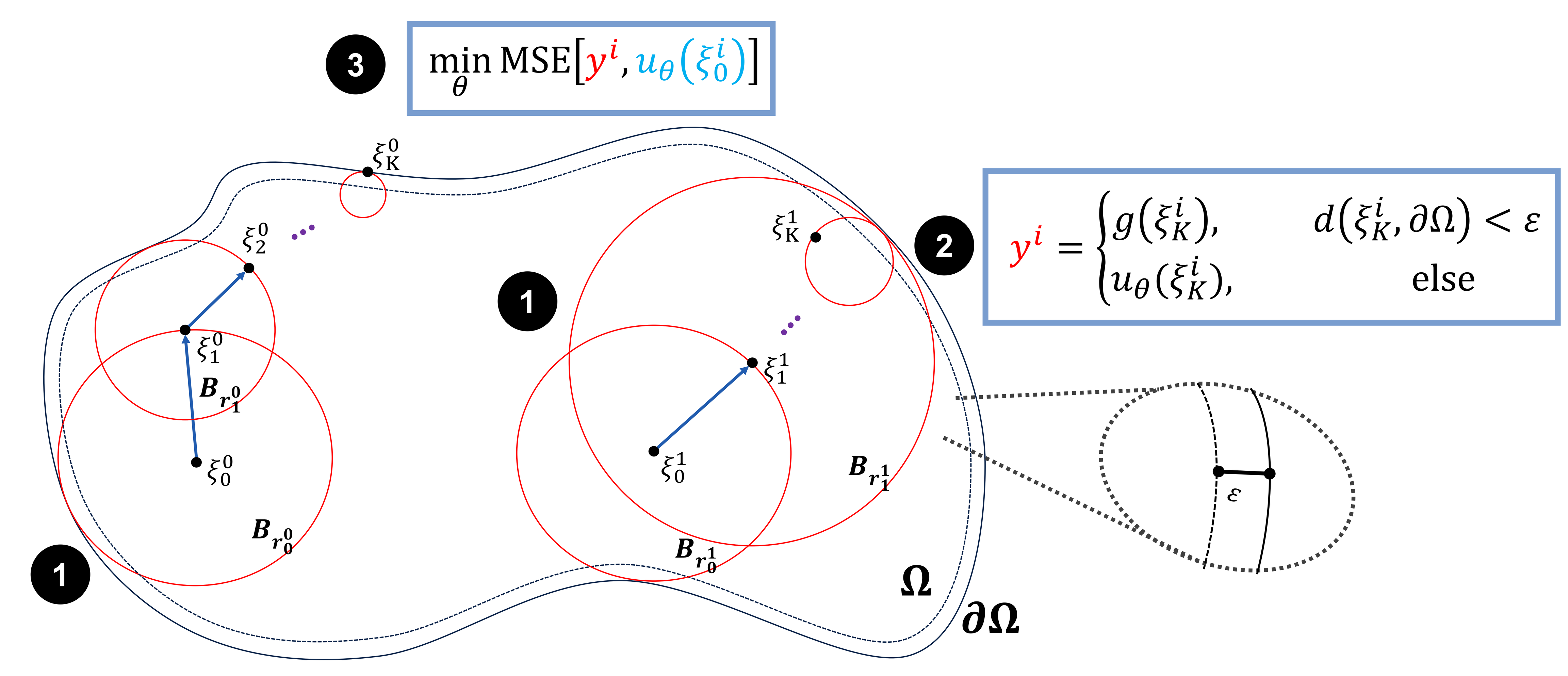
$$\xi_{k+1} \sim X_{\tau_k}^{\xi_k} \sim \partial\mathcal{U}(B_{r_k}(\xi_k)),\ r_k = \min_{x \in \partial\Omega} \|\xi_k - x\|$$

$$\tau_k \coloneqq \{t: X_t^{\xi_k} \notin B_{r_k}(\xi_k)\} \text{ (local exit time)}$$

☺ **Does not require any derivative**

☺ **Fast propagation of boundary information**

Neural Walk-on-Spheres

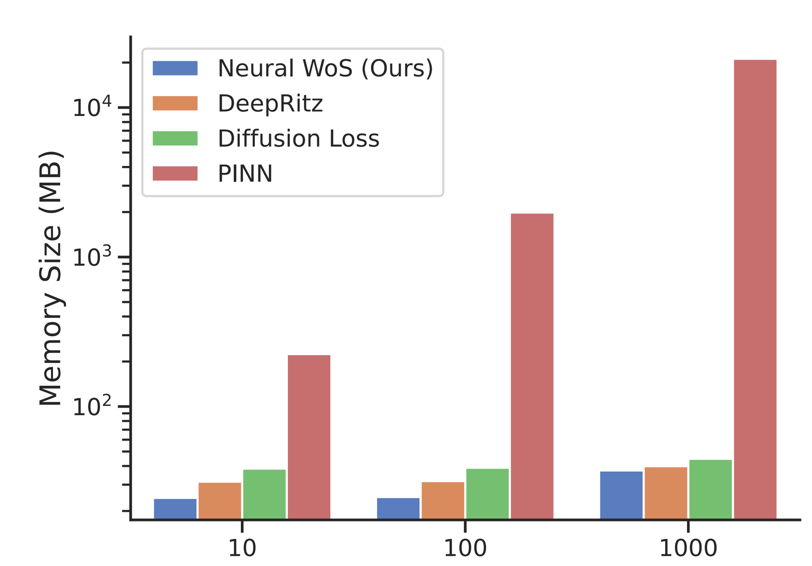


1. Move initial points $\xi_0^i \sim \mathcal{U}(\Omega)$ to boundary via random sampling over spheres $B_r(\xi_k^i)$ at each step k
2. Estimate the solution y^i depending on the convergence of the walk at maximum step K
3. Optimize the model u_θ using y^i as the label

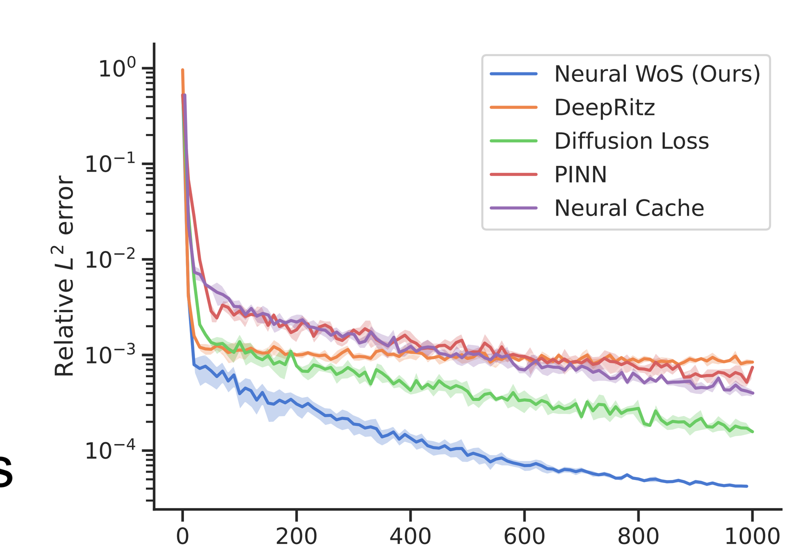
Results

Best performance in comparison to our baselines

Method	Problem			
	Laplace (10d)	Committor (10d)	Poisson Rect. (10d)	Poisson (50d)
PINN	$7.42e^{-4} \pm 1.84e^{-4}$	$4.10^{-3} \pm 1.11e^{-3}$	$1.35e^{-2} \pm 1.57e^{-3}$	$7.70e^{-3} \pm 2.25e^{-3}$
Deep Ritz	$8.43e^{-4} \pm 6.29e^{-5}$	$6.15e^{-3} \pm 5.30e^{-4}$	$1.06e^{-2} \pm 6.20e^{-4}$	$1.05e^{-3} \pm 1.70e^{-4}$
Diffusion loss	$1.57e^{-4} \pm 7.74e^{-6}$	$4.48e^{-2} \pm 6.93e^{-3}$	$9.69e^{-2} \pm 1.03e^{-2}$	$5.96e^{-4} \pm 1.06e^{-5}$
Neural Cache	$3.99^{-4} \pm 4.08e^{-5}$	$1.26e^{-3} \pm 5.82e^{-5}$	$4.98e^{-2} \pm 1.80e^{-2}$	$1.63e^{-2} \pm 1.42e^{-2}$
WoS	$1.08e^{-3} \pm 1.34e^{-6}$	$1.99e^{-3} \pm 9.79e^{-6}$	$2.32e^{-1} \pm 2.09e^{-1}$	$4.50e^{-3} \pm 7.38e^{-4}$
NWoS (ours)	$\mathbf{4.29e^{-5} \pm 2.02e^{-6}}$	$\mathbf{6.56e^{-4} \pm 2.42e^{-5}}$	$\mathbf{2.60e^{-3} \pm 9.99e^{-5}}$	$\mathbf{4.82e^{-4} \pm 1.32e^{-5}}$



Scalable to higher dimensions with minimal memory requirements



Faster convergence with higher accuracy

Key Takeaways

Neural Walk-on-Spheres is

- Applicable to parametric Poisson-type equations on **general domains**
- Supported by **theoretical guarantees** and **free** from the **curse of dimensionality**
- Exploiting a supervised loss with **noisy but cheap and unbiased** estimates for **higher efficiency** and **accuracy**