





Neural Quantum Spectral Operator Learning for Solving Partial Differential Equations

Chanyoung Kim[1], Myeonghwan Seong[2], Yujin Kim[2],

Daniel K. Park[2†], Youngjoon Hong[1†]

1) Seoul National University, 2) Yonsei University, †Corresponding authors

Motivation

Unsupervised Operator Learning

Operator learning aims to learn a PDE solution operator by neural networks:

- Supervised (FNO, DeepONet): needs large labeled datasets from costly numeric solvers
- Unsupervised (SCLON, PI-DON): label-free, but cost scales as $O\left(N^{2d}\right)$

Partial Differential Equation (PDE)

$$-\epsilon \operatorname{div}(a\nabla u) + \mathcal{F}(u) = f, \quad \text{in } \Omega$$
$$\mathcal{B}(u) = 0, \quad \text{on } \partial\Omega$$

$\mathcal{F}$: differential operator
$\mathcal{B}$: boundary operator

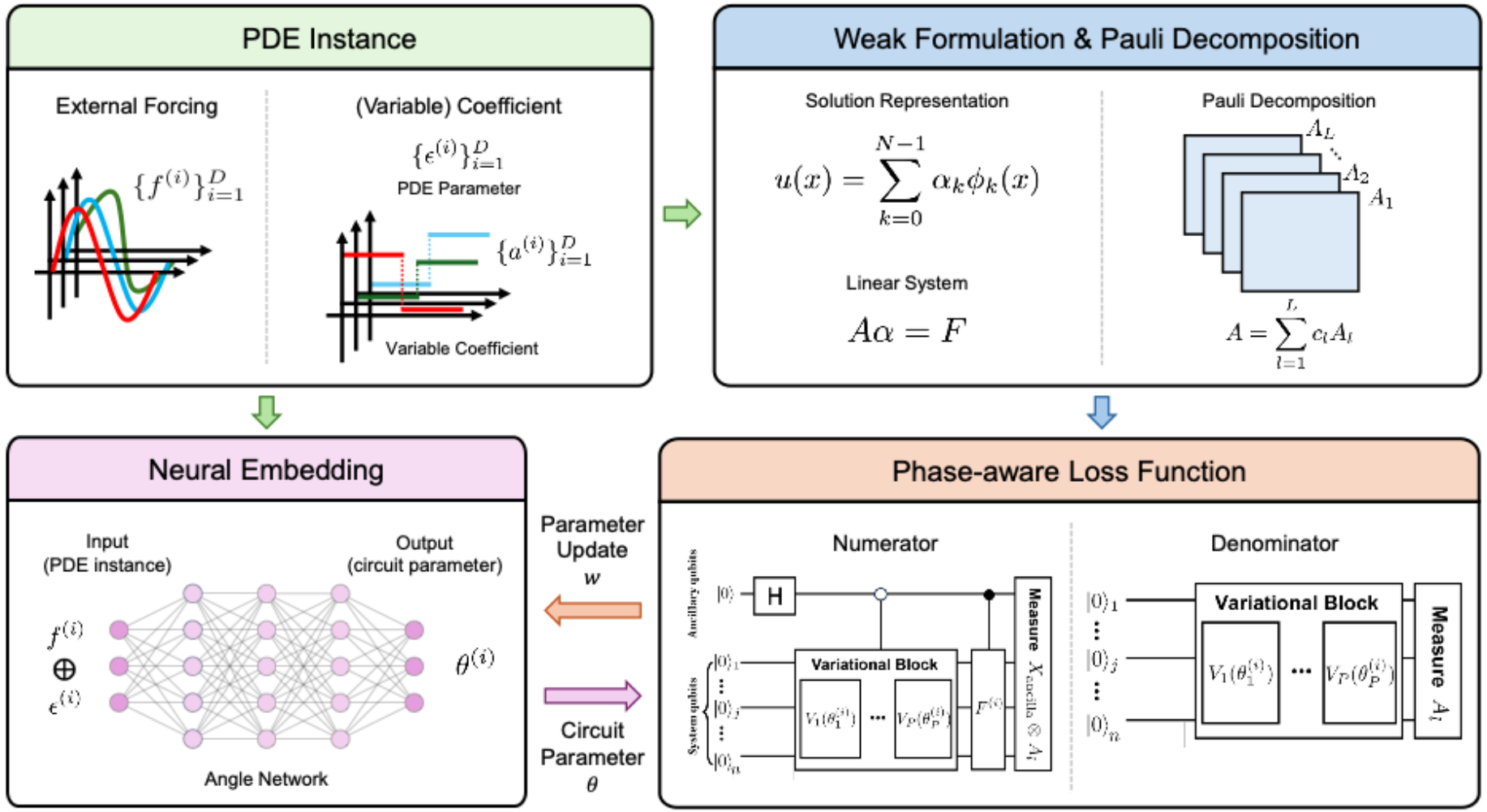


Operator Learning
(learns PDE Solution Operator)

$$\mathcal{G} : f \mapsto u$$

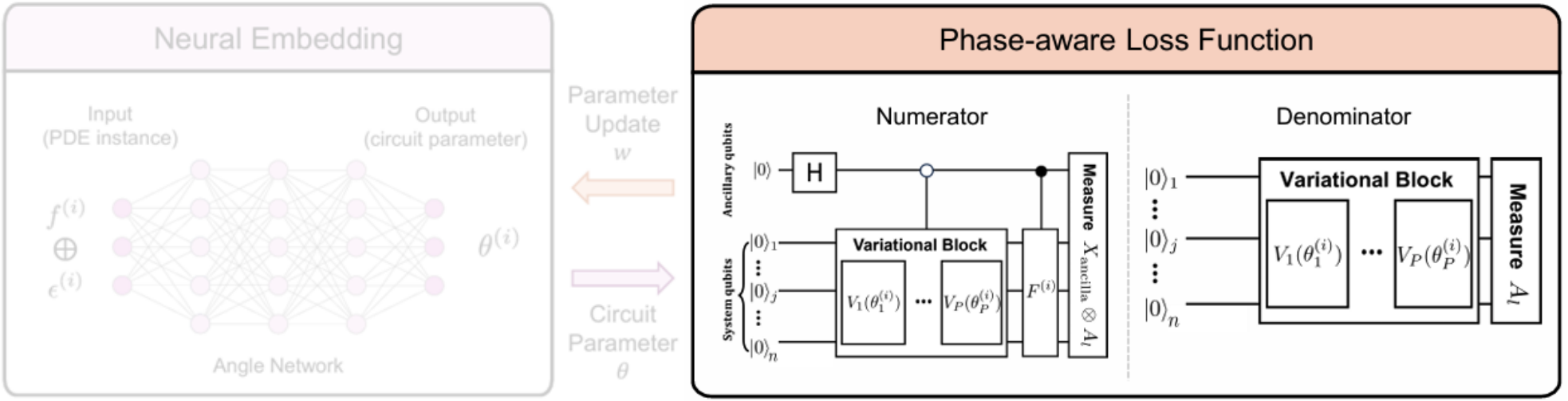
Methodology

Neural Variational Quantum Linear Solver



Methodology

Phase-aware Loss Function

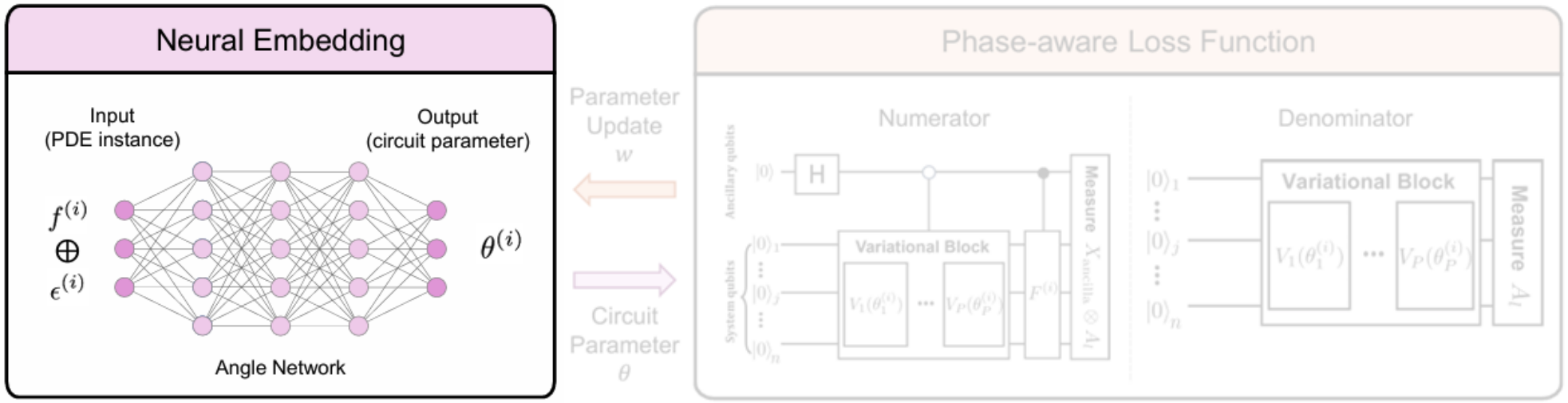


$$\mathcal{L}_{\text{NVQLS}}(w) = \frac{1}{D}\sum_{i=1}^{D}\left(1 - \frac{\text{Re}\left(\sum_{l=1}^{L} c_l \left\langle F^{(i)}\right| A_l \left|\hat{\alpha}(f^{(i)};w)\right\rangle\right)}{\sqrt{\sum_{l=1}^{R} d_l \left\langle \hat{\alpha}(f^{(i)};w)\right| A_l \left|\hat{\alpha}(f^{(i)};w)\right\rangle}}\right)$$

This provides the phase information and enables grouped measurement.

Methodology

Neural Embedding



$$V(\theta) = V(g(f; w)) \quad \text{where} \quad g : \text{neural network}$$

NE enables the multi-instance training with a compact output.

(Conditional) Cost and Memory Estimates

Table 1: Conditional computational cost per step for D PDE instances and peak classical working memory for fixed D.

Model	Per-step computational cost estimate	Classical working memory
NVQLS	$\mathcal{O}\left(DN^d(\log_2 N^d)^4/\epsilon^2 + DN^d(\log_2 N^d)^2\right)$	$\mathcal{O}\left(N^d(\log_2 N^d)^2\right)$
VQLS	$\mathcal{O}\left(DN^{2d}(\log_2 N^d)^5/\epsilon^2\right)$	$\mathcal{O}\left((\log_2 N^d)^2\right)$
SCLON / PI-DON	$\mathcal{O}\left(DN^{2d}\right)$	$\mathcal{O}\left(N^{2d}\right)$

Under assumptions:

- Pauli decomposition scaling (assisted by experiments)
- Ansatz scaling (assisted by experiments)
- Embedding scaling (for specific forcing function spaces)

N: number of basis functions
d: dimension
ε: (root-mean-square) error tolerance

Numerical Experiments

Comparison with Classical Baselines

Table. NVQLS vs. SCLON (1D PDEs, mean over 200 test samples)

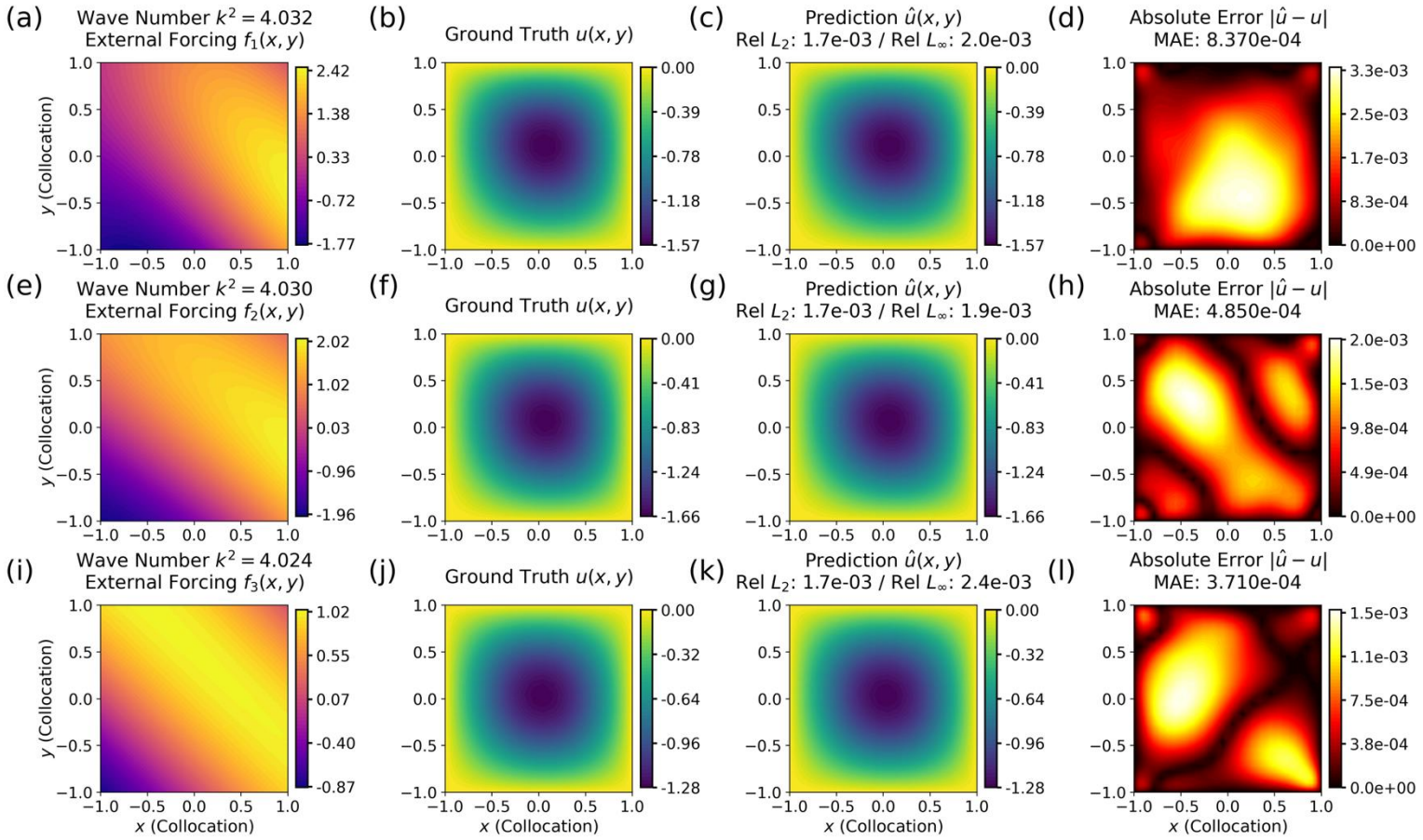
	RD ($\epsilon = 0.1$)		Helm ($k^2 = 4.7$)		CD ($\epsilon = 0.1$)	
Model	Rel. L_2	Rel. L_∞	Rel. L_2	Rel. L_∞	Rel. L_2	Rel. L_∞
NVQLS (Ours)	$\mathbf{1.9 \times 10^{-4}}$	$\mathbf{2.0 \times 10^{-4}}$	$\mathbf{2.5 \times 10^{-3}}$	$\mathbf{2.5 \times 10^{-3}}$	$\mathbf{1.0 \times 10^{-3}}$	$\mathbf{1.1 \times 10^{-3}}$
SCLON	4.8×10^{-3}	5.5×10^{-3}	7.3×10^{-3}	7.6×10^{-3}	1.3×10^{-3}	1.4×10^{-3}

Table. NVQLS vs. FNO (1D wave equation, D = labeled training pairs)

Metric	NVQLS (Ours, $D = 0$)	FNO ($D = 50$)	FNO ($D = 100$)	FNO ($D = 200$)
Rel. L_2 error	2.26×10^{-2}	3.54×10^{-2}	2.14×10^{-2}	$\mathbf{1.30 \times 10^{-2}}$
Rel. L_∞ error	$\mathbf{1.40 \times 10^{-2}}$	3.98×10^{-2}	2.30×10^{-2}	1.42×10^{-2}

Numerical Experiments

Joint Varying Coefficient-Forcing Input



Also: 3D elliptic PDEs (rel. $L_2 \sim 10^{-2}$), discontinuous variable coefficients (Darcy, parabolic)

Takeaways

- Quantum-enhanced unsupervised operator learning:
  - Proposed Phase-aware Overlap Cost / Neural Embedding
  - Improved per-iteration cost and memory scaling (under stated assumptions)
- Validated on diverse PDE settings:
  - Jointly varying coefficient-forcing, 3D PDEs, discontinuous variable coefficients
- Limitations:
  - Hardware-level noise prevents accurate training