Parametric reduced order modeling for nonlocal PDEs

Yumeng Wang
Department of Mathematics and Statistics
Missouri University of Science and Technology

Advisor: Prof. Yanzhi Zhang

The 9th Annual Meeting of SIAM Central States Section
University of Missouri-Kansas City
October 5–6, 2024

Outline

1. Problem set-up and motivation
2. Proposed method: Convolutional neural network-based ROM
3. Numerical experiments
4. Summary

Problem and Motivation

Parameterized nonlocal PDEs

General parametric nonlocal PDEs:

$$\begin{aligned} \mathcal{L}_t^{\boldsymbol{\mu}} u(\mathbf{x}, t; \boldsymbol{\mu}) &= \mathcal{L}_{\mathbf{x}}^{\boldsymbol{\mu}} u(\mathbf{x}, t; \boldsymbol{\mu}) + \mathcal{F}(\mathbf{x}, t, u; \boldsymbol{\mu}), && \text{for } \mathbf{x} \in \Omega, && t \in (0, T], \\ \mathcal{B} u(\mathbf{x}, t; \boldsymbol{\mu}) &= g(\mathbf{x}, t; \boldsymbol{\mu}), && \text{for } \mathbf{x} \in \Omega_\Gamma, && t \in (0, T], \end{aligned}$$

where

- $\mathcal{L}_t^{\boldsymbol{\mu}}$ and $\mathcal{L}_{\mathbf{x}}^{\boldsymbol{\mu}}$ are the linear differential operators for t and x
- $\mathcal{F}$ represents the nonlinear terms of u
- $\mathcal{B}$ represents a linear boundary operator

Here, $\boldsymbol{\mu}$ may appear in the differential terms with respect to time or space, nonlinear terms, boundary conditions, or in initial conditions if a time-dependent problem is considered.

Parameterized nonlocal PDEs

Challenges:

- Many application requires to solve the parametric nonlocal PDEs repeatedly for different parameters

  Example: fractional Poisson equation in image inpainting

$$(-\Delta)^{\frac{\alpha}{2}} u(\mathbf{x}) = f(\mathbf{x}), \qquad \text{for } \alpha \in (0, 2)$$

- Numerical methods needs intensive storage and computation cost
  - Solving dense stiffness matrices for spatial nonlocal problem
  - Requiring entire history from $t = 0$ for temporal nonlocal problem
  - Singularity kernels
  - High dimensional PDEs

Strategy: Reduced order modeling

Motivation

Existing ROMs: (Guan et al, 2017, Barrault et al, 2004, Gruber et al, 2022, Burkovska et al, 2020, Bonito et al., 2020, Maday et al, 2012)

They mainly focus on the fractional Poisson equation:

$$(-\Delta)^{\frac{\alpha}{2}} u(\mathbf{x}) = f(\mathbf{x}), \qquad \text{for } \alpha \in (0, 2)$$

Challenges of developing ROMs for nonlocal problems:

- Lack of affine dependence
- Singularity
- Nonlocal boundary conditions

Goal: We develop a neural network based ROM, nonlinear and non-intrusive method, for nonlocal parameterized PDEs.

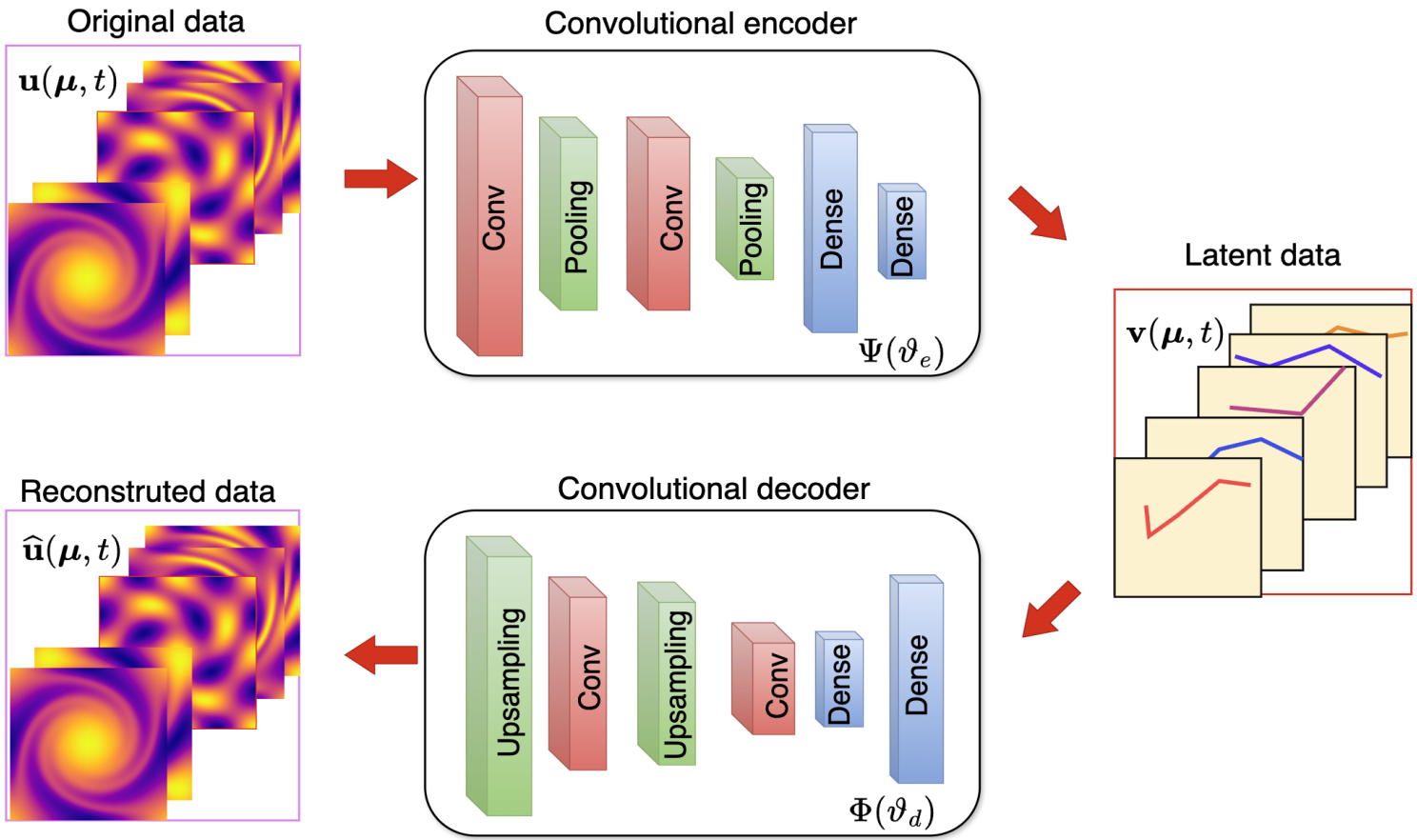
Proposed methods

Two parts:

1. Dimensional reduction with CAE
2. Solution approximation in latent space

Reduced order model: CAE

Convolutional autoencoder (CAE) : With convolutional layer, CAE learn an low-dimensional representation of high-dimensional data, via reconstructing the input data accurately. This proves effectively in addressing complex nonlinear problems.



Reduced order model: CAE

CAE: Denote the encoder as $\Psi : \mathbb{R}^N \to \mathbb{R}^n$, decoder as Φ:$\mathbb{R}^n \to \mathbb{R}^N$

$$\begin{aligned} \mathbf{v} &= \Psi(\mathbf{u}; \boldsymbol{\vartheta}_{\mathrm{e}}), \\ \widehat{\mathbf{u}} &= \Phi(\mathbf{v}; \boldsymbol{\vartheta}_{\mathrm{d}}), \end{aligned}$$

Here, $\mathbf{u} \in \mathbb{R}^N$, $\widehat{\mathbf{u}} \in \mathbb{R}^N$, $\mathbf{v} \in \mathbb{R}^n$, and dimension $n \ll N$.

Loss function:

$$\begin{aligned} \mathcal{L}_{\mathrm{CAE}}(\boldsymbol{\vartheta}_{\mathrm{e}}, \boldsymbol{\vartheta}_{\mathrm{d}}) &= \frac{1}{N_{\mathrm{tr}}^t N_{\mathrm{tr}}^{\boldsymbol{\mu}}} \sum_{i=1}^{N_{\mathrm{tr}}^{\boldsymbol{\mu}}} \sum_{m=1}^{N_{\mathrm{tr}}^t} \|\mathbf{u}(\boldsymbol{\mu}_i, t_m) - \widehat{\mathbf{u}}(\boldsymbol{\mu}_i, t_m)\|_{\mathrm{mse}} \\ &= \frac{1}{N_{\mathrm{tr}}^t N_{\mathrm{tr}}^{\boldsymbol{\mu}}} \sum_{i=1}^{N_{\mathrm{tr}}^{\boldsymbol{\mu}}} \sum_{m=1}^{N_{\mathrm{tr}}^t} \|\mathbf{u}(\boldsymbol{\mu}_i, t_m) - \Phi\big(\Psi(\mathbf{u}(\boldsymbol{\mu}_i, t_m); \boldsymbol{\vartheta}_{\mathrm{e}}); \boldsymbol{\vartheta}_{\mathrm{d}}\big)\|_{\mathrm{mse}} \end{aligned}$$

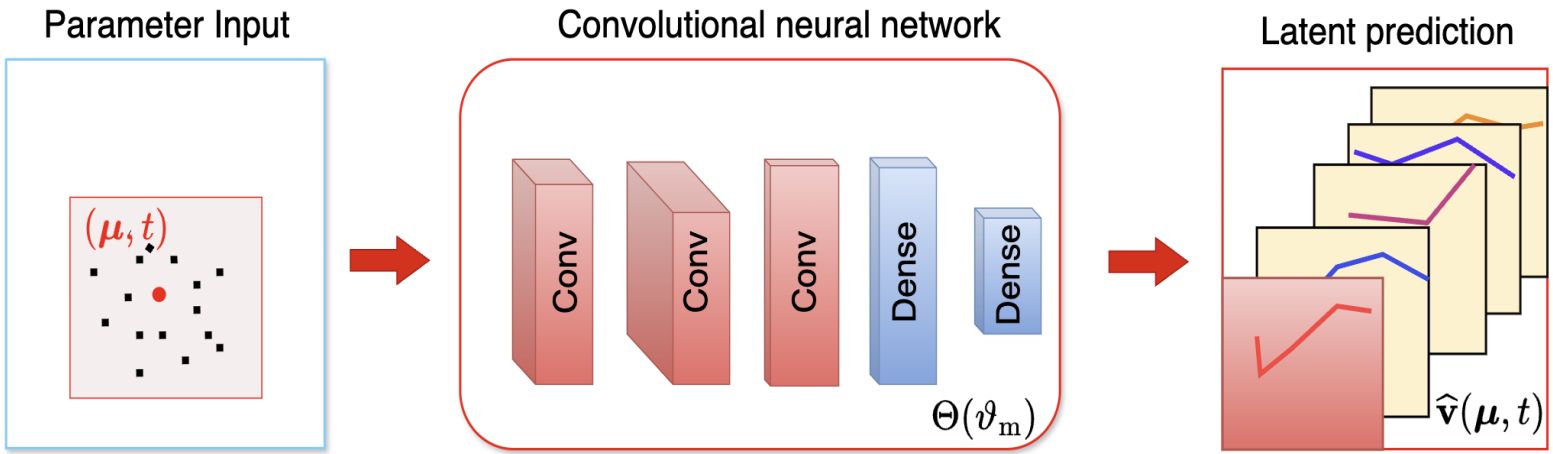
- $\{\mathbf{u}(\boldsymbol{\mu}_i, t_m) \in \mathbb{R}^N \mid 1 \le i \le N_{\mathrm{tr}}^{\boldsymbol{\mu}},\ 1 \le m \le N_{\mathrm{tr}}^t\}$ represents dataset
- $\|\cdot\|_{\mathrm{mse}}$ represents the mean squared error (MSE)

Latent Mapping with CNN

CNN: Denote CNN as $\Theta : \mathbb{R}^p \times \mathbb{R} \to \mathbb{R}^n$,

$$\widehat{\mathbf{v}} = \Theta(\boldsymbol{\mu}, t; \boldsymbol{\vartheta}_{\mathrm{m}})$$

Note: time is processed as an additional paramete



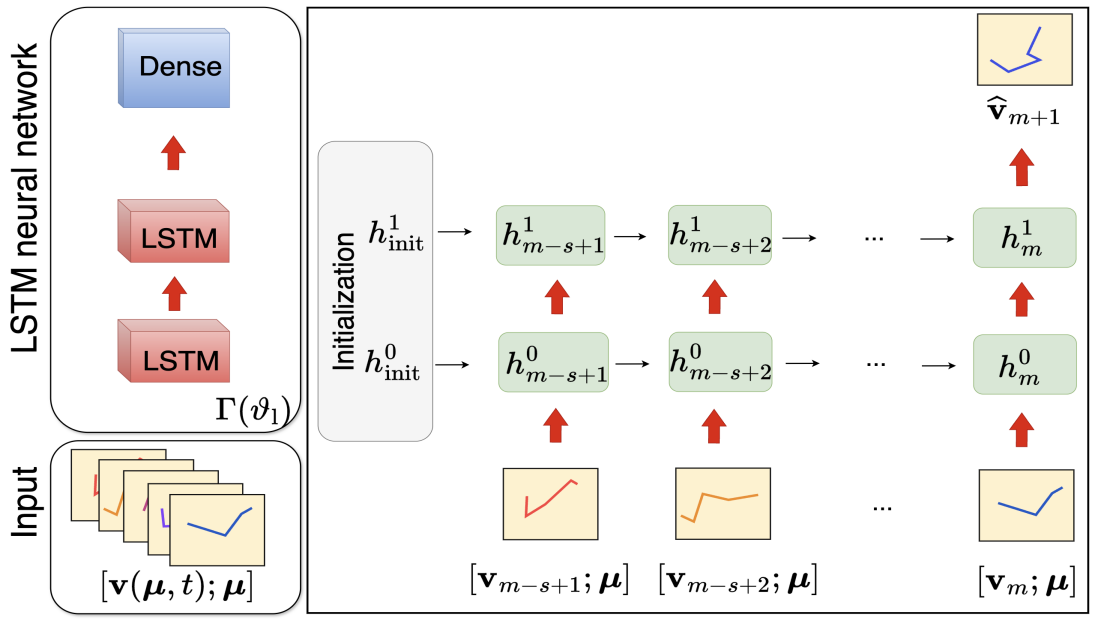
Loss function:

$$\mathcal{L}_{\mathrm{Map}}(\boldsymbol{\vartheta}_{\mathrm{m}}) = \frac{1}{N_{\mathrm{tr}}^{t} N_{\mathrm{tr}}^{\boldsymbol{\mu}}} \sum_{i=1}^{N_{\mathrm{tr}}^{\boldsymbol{\mu}}} \sum_{m=1}^{N_{\mathrm{tr}}^{t}} \left\| \mathbf{v}(\boldsymbol{\mu}_i, t_m) - \Theta(\boldsymbol{\mu}_i, t_m; \boldsymbol{\vartheta}_{\mathrm{m}}) \right\|_{\mathrm{mse}}$$

Latent Time Marching with LSTM

LSTM: Denote the LSTM as $\Gamma\colon \mathbb{R}^{n+p} \to \mathbb{R}^n$, i.e.,

$$\widehat{\mathbf{v}}(\boldsymbol{\mu}, t_{m+1}) = \Gamma\left(\left\{[\mathbf{v}(\boldsymbol{\mu}, t_{m-k+1}); \boldsymbol{\mu}]\right\}_{k=1}^{s}; \boldsymbol{\vartheta}_{\mathrm{l}}\right)$$



Loss function:

$$\mathcal{L}_{\mathrm{LSTM}}(\boldsymbol{\vartheta}_{\mathrm{l}}) = \frac{1}{(N_{\mathrm{tr}}^t - s) N_{\mathrm{tr}}^{\boldsymbol{\mu}}} \sum_{i=1}^{N_{\mathrm{tr}}^{\boldsymbol{\mu}}} \sum_{m=s}^{N_{\mathrm{tr}}^t} \left\| \mathbf{v}(\boldsymbol{\mu}_i, t_{m+1}) - \Gamma\left(\left\{[\mathbf{v}(\boldsymbol{\mu}, t_{m-k+1}); \boldsymbol{\mu}]\right\}_{k=1}^{s}; \boldsymbol{\vartheta}_{\mathrm{l}}\right) \right\|_{\mathrm{ms}}$$

Latent Space Modeling

Latent space modeling:

- **CNN-Mapping:** A convolutional neural network (CNN) learns the mapping from the parameter space to the latent solution space.
- **LSTM Time marching:** A recurrent neural network (RNN), particularly a long short-term memory network (LSTM), models the temporal dynamics in the latent space under specific parameter conditions.

Difference:

CNN	**LSTM**
Mapping	Time marching
Time dependent & independent problem	Time dependent problem
Arbitrary time	Fixed time points
$t_{\text{new}} \rightarrow \widehat{u}(t_{\text{new}})$	$\widehat{u}(t_0) \rightarrow \widehat{u}(t_1) ... \rightarrow \widehat{u}(t_n)$

Training and Prediction

Offline training stage: Our proposed method has a decoupled training strategy where the CAE and latent-space models are trained separately.

- Training CAE for low dimensional latent representation
- Latent space models : Training CNN-Mapping or LSTM Time marching

Online prediction stage: Given the unseen parameters, CNN mapping:

$$(\boldsymbol{\mu}_{\text{new}}, t_{\text{new}}) \xrightarrow{\Theta_m(\boldsymbol{\vartheta}_{\text{m}}^*)} \widehat{\mathbf{v}}(\boldsymbol{\mu}_{\text{new}}, t_{\text{new}}) \xrightarrow{\Phi(\boldsymbol{\vartheta}_{\text{d}}^*)} \widehat{\mathbf{u}}(\boldsymbol{\mu}_{\text{new}}, t_{\text{new}})$$

Properties:

- CAE could effectively reduce high-dimensions to low-dimensions, a nonlinear noninstrusive method
- Latent CNN mapping can predict arbitrary time within time frame
- Bypass the challenge of traditional ROM methods in solving parametrized PDEs

Numerical Experiments

Benchmark Test: Nonlocal Poisson Equation

Nonlocal parameterized 2D Poisson equation on the domain $\Omega = (-1, 1)^2$:

$$\begin{aligned} (-\Delta)^{\frac{\alpha}{2}} u(\mathbf{x}) &= f(\mathbf{x}, \mu_1, \mu_2), \qquad \text{for } \mathbf{x} \in \Omega, \\ u(\mathbf{x}) &= 0, \qquad \text{for } \mathbf{x} \in \mathbb{R}^2 \backslash \Omega, \end{aligned}$$

where

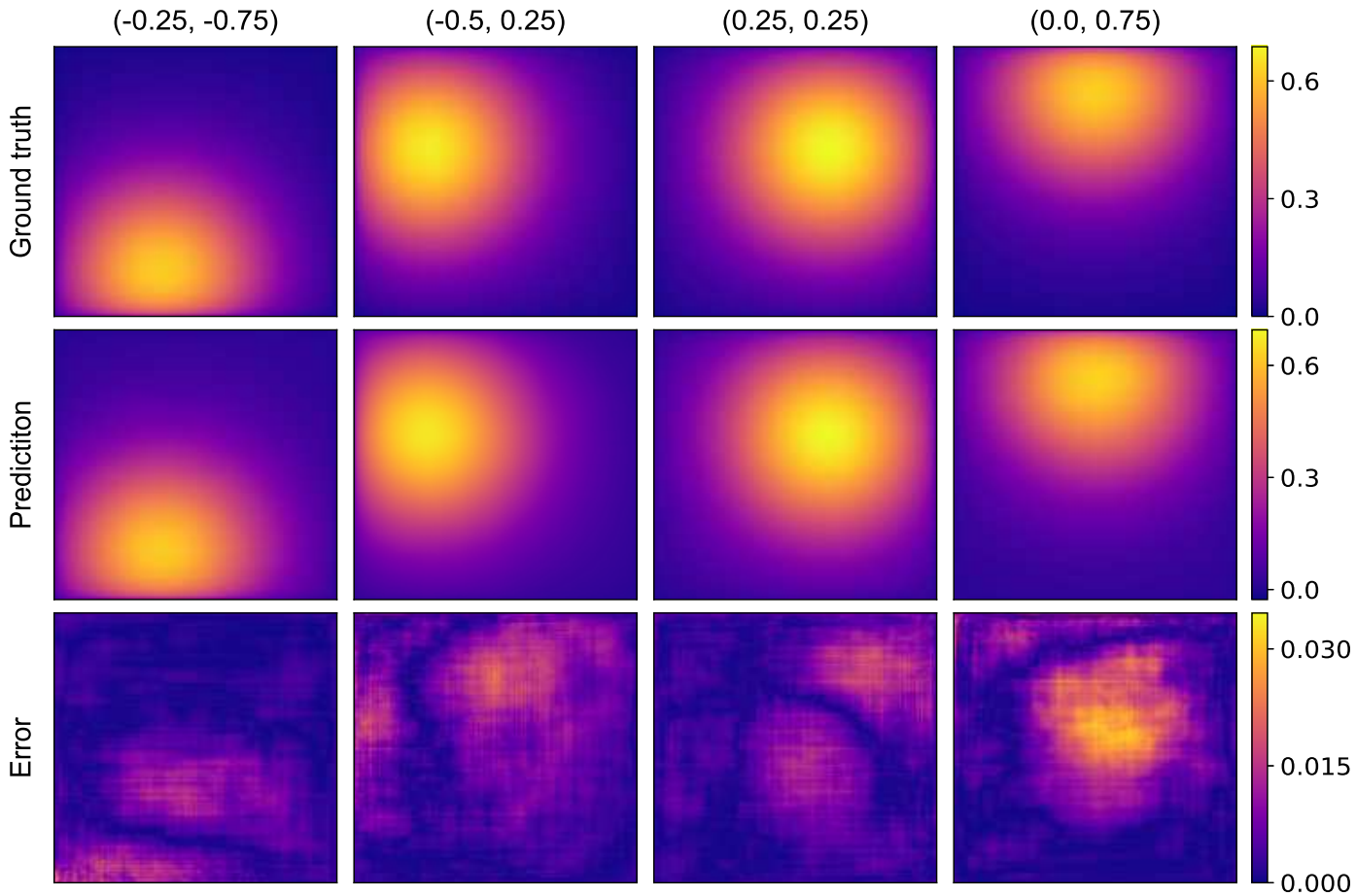
$$f(\mathbf{x}, \mu_1, \mu_2) = \exp[-2((x - \mu_1)^2 + (y - \mu_2)^2)],$$

for $\alpha \in (0, 2]$, and $\mu_1, \mu_2 \in [-1, 1]$.

Data: High-fidelity data are prepared by numerically solving the equation.

- Input grid dimensions: $N_x = N_y = 256$, Input dimension: 65536
- Training data: $N_{\text{tr}}^{\boldsymbol{\mu}} = 396$ with equispaced points of $(11, 6, 6)$ sampled from (α, μ_1, μ_2) parameter space
- Testing data: $N_{\text{test}}^{\boldsymbol{\mu}} = 931$ with non-overlapping equispaced points of $(19, 7, 7)$

Benchmark Test: Nonlocal Poisson Equation

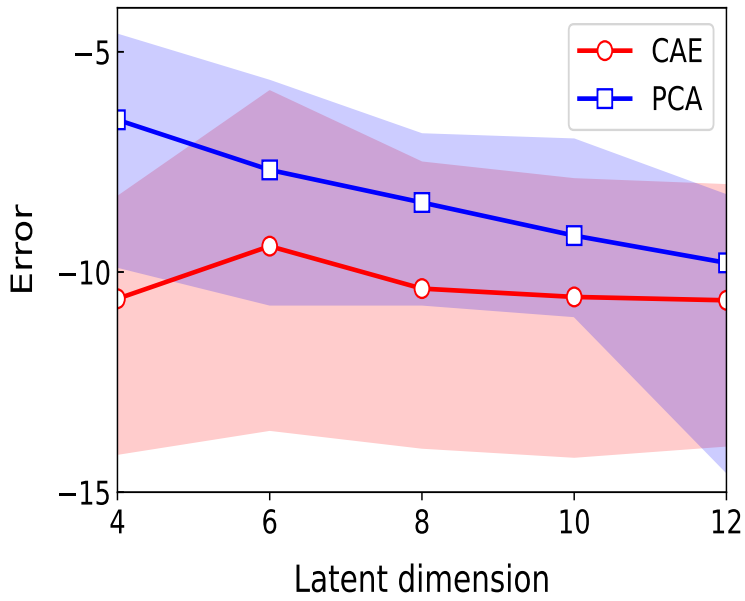


$$\alpha = 0.48$$

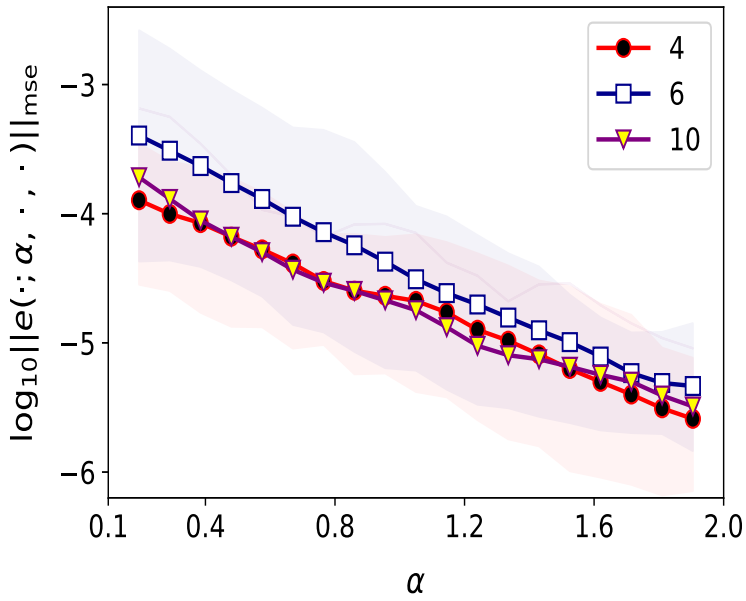
Here the latent dimension $n = 4$

Benchmark Test: Nonlocal Poisson Equation

Comparison



PCA vs CAE



Effect of latent dimension

Nonlocal Parameterized 2D Allen–Cahn Equation

Nonlocal parameterized 2D Allen–Cahn equation defined on $\Omega = (-1, 1)^2$, $t \in (0, 10]$

$$\partial_t u(\mathbf{x}, t) = 10 \int_\Omega J_\varepsilon(\mathbf{x} - \mathbf{x}')\big(u(\mathbf{x}', t) - u(\mathbf{x}, t)\big)\mathrm{d}\mathbf{x}' + \beta(u - u^3),$$

where $J_\varepsilon(\mathbf{x} - \mathbf{x}') = \frac{1}{\varepsilon} e^{-|\mathbf{x}-\mathbf{x}'|^2/\varepsilon^2}$ for $\varepsilon > 0$. and the initial condition is

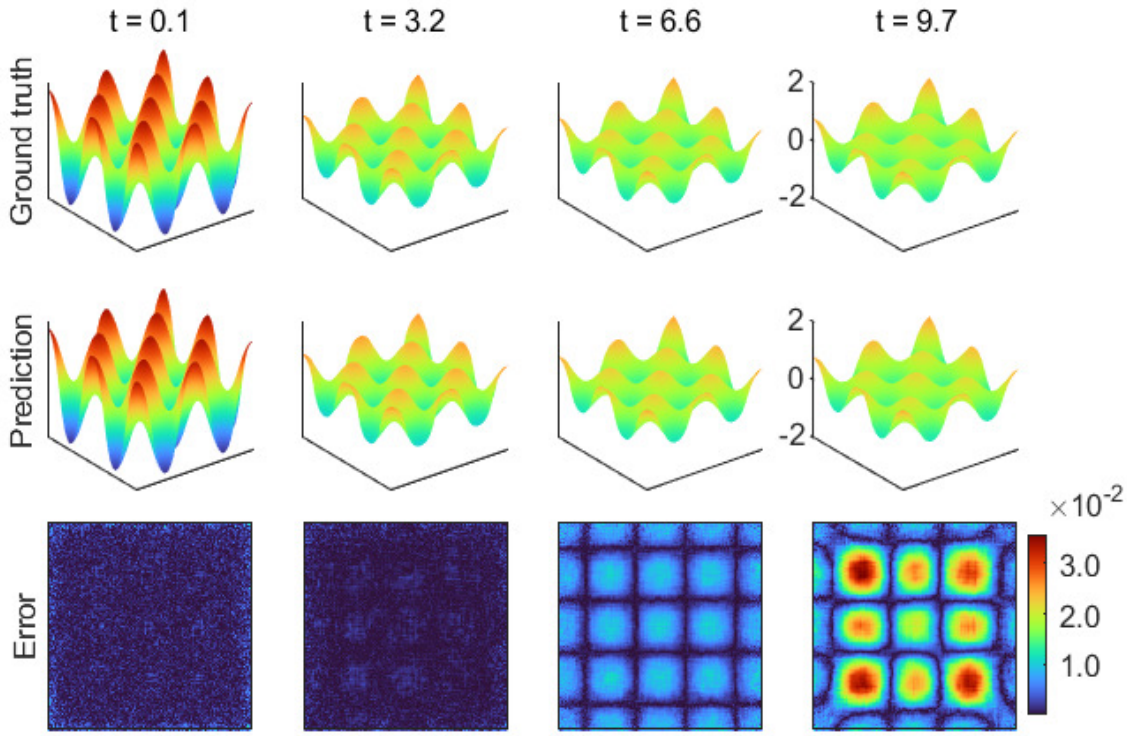
$$u(\mathbf{x}, 0) = 2\cos(2\pi x)\cos(2\pi y), \qquad \text{for } \mathbf{x} \in \bar{\Omega},$$

Here the parameters are $\varepsilon \in [0.05, 0.1]$ and $\beta \in [0.1, 0.5]$.

Data:

- $N_x = N_y = 128$ and $N_t = 10000$; Input dimension = 16384
- Training data: $N_{\text{tr}}^{\boldsymbol{\mu}} = 30$, equispaced points sampled of $(6, 5)$ in the parameters space. Each training data includes 29 (Nonuniformly sampled)
- Testing data: $N_{\text{te}}^{\boldsymbol{\mu}} = 336$, equispaced points sampled of $(14, 24)$

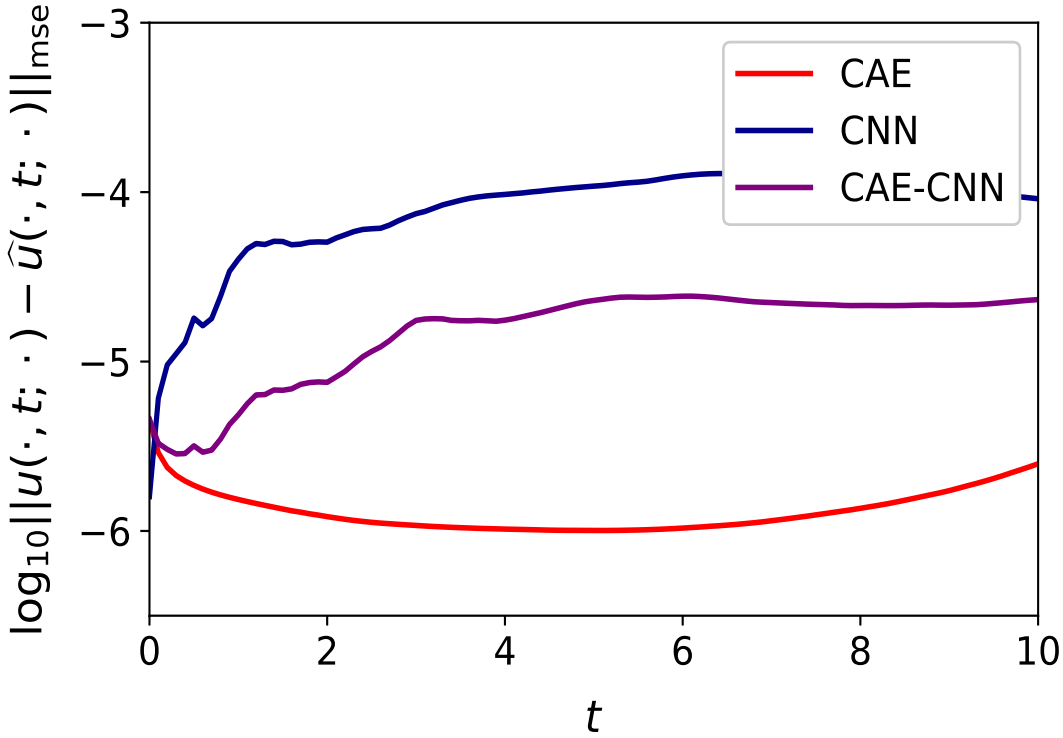
Allen–Cahn Equation



$$\varepsilon = 0.0955, \beta = 0.468$$

- Latent dimension $n = 4$
- LSTM performs similar result as CNN mapping

Allen–Cahn Equation



Model performance

Computational Efficiency

Test 5000 parameters:

	Train/Test	$N = 2^{14}$	$N = 2^{16}$
Offline Training	CAE	3.552e+4	9.107e+4
	CNN	230	443
Online Prediction	$t = 1$	2.999	9.074
	$t = 10$	3.05	9.095
Numerical Methods	$t = 1$	3.264e+4	5.335e+5
	$t = 10$	3.334e+5	5.240e+6

Table: Computational time of our model & traditional method (seconds)

Observation:

- CAE dominates the offline stage time, depending on data dimension and volume; CNN is insignificant
- The online stage predicts solution for varying parameters fast
- Numerical methods take more time with time and dimension increase

Summary

Our proposed method is the first neural network based ROMs to solve parameterized nonlocal PDEs

- The framework separates high-dimensional prediction of nonlocal parameterized PDEs into two sub-tasks: reduce order model and latent space model
- CAE could reduce computational cost significantly for solving nonlocal problem, nonlinear and nonintrusive
- CAE learns is more efficient in nonlinear problem than traditional PCA
- CNN-mapping learns continuous mapping both in time and parameters space
- LSTM-time marching as an alternative method for time dependent PDEs, can learn temporal dynamics efficiently

Thank You!

Yumeng Wang
Department of Mathematics and Statistics
Missouri University of Science and Technology

Email: yw2bc@umsystem.edu

More information: *Convolutional neural network-based reduced-order modeling for parametric nonlocal PDEs*, Y. Wang, S. Zhou, and Y. Zhang, 2024 (submitted)