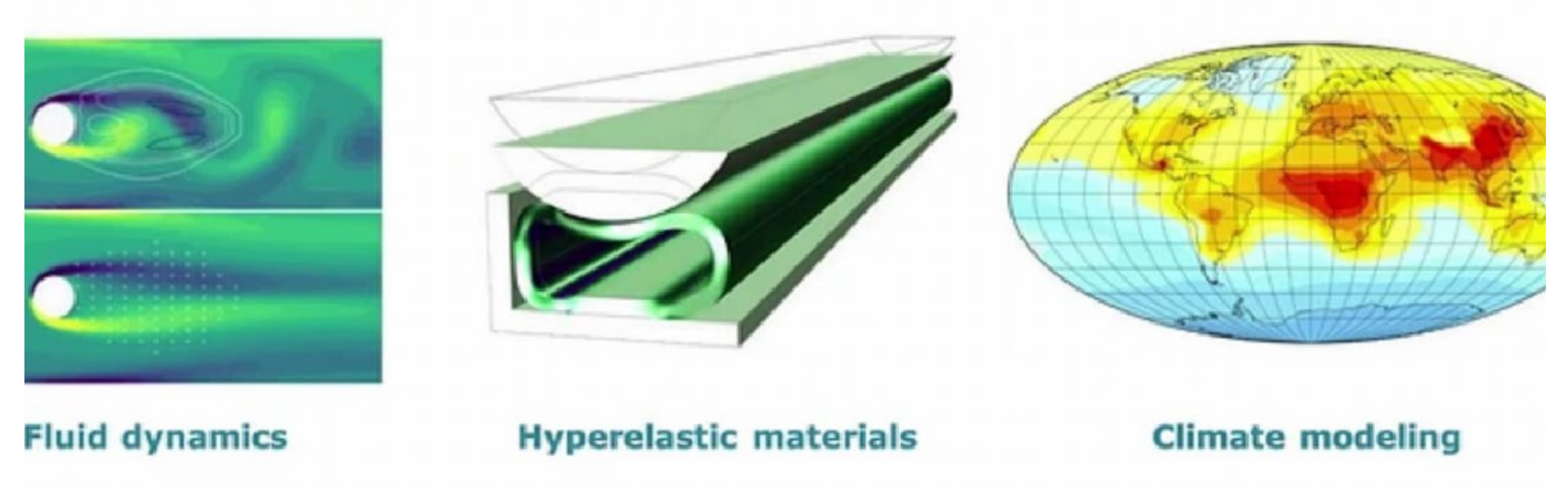
Uncertainty Quantification for Forward and Inverse Problems of PDEs via Latent Global Evolution

Tailin Wu[1]*, Willie Neiswanger[1,2]*, Hongtao Zheng[1]*, Stefano Ermon[3], Jure Leskovec[3]

[1]Westlake University, [2]University of Southern California, [3]Stanford University

Motivation

Deep learning is rapidly becoming a go-to method in scientific and industrial applications, offering a faster alternative to traditional partial differential equation (PDE) solvers with speedups ranging from 10 to 1000 times. However, one key challenge that prevents their wide application in science and industry is that practitioners do not know the uncertainty of the predictions, especially in places with critical decision making.



The goal is to design of an uncertainty quantification tool for time-series fluid prediction models.

Prior works

Most works only deal with stationary PDEs that does not involve time:

- ConvPDE-UQ [1] learns a mapping from the elliptic PDE's parameter f to solution u, with UQ, where the PDE is given by:

$$\begin{cases} \Delta u = f & \text{in} \quad \Omega \\ u = 0 & \text{on} \quad \partial\Omega \end{cases}$$

- Similarly, [2] learns a mapping from elliptic PDE's parameter f to solution u, with UQ, without labeled data.

The ADCME [1] package provides tools for a number of forward and inverse problems. However, it has serious limitations for UQ:

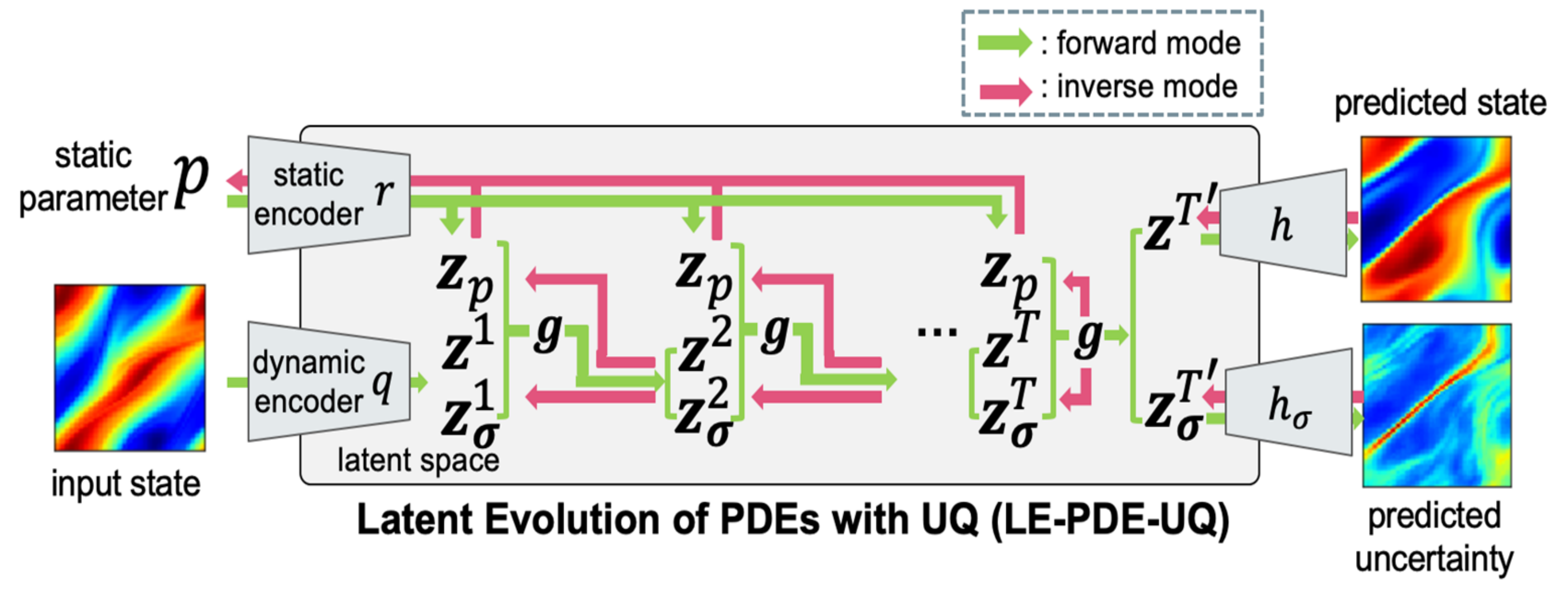
- It provides language for constructing a physics-informed neural network (PINN) as a Bayesian neural network, and then performing MCMC for UQ with these models.
- Limitation: their MCMC procedure is restrictive and not scalable (uses a random-walk Metropolis Hastings algorithm; requires specialized probabilistic programming framework to train). PINN cannot do autoregressive rollout and generalize beyond the training time range.

[1] Winovich, et al. 2019,
[2] Zhu, Yinhao, et al.

Approach

Main idea:

We introduce Latent Evolution of PDEs with Uncertainty Quantification (LE-PDE-UQ), a method for deep learning models with efficient and accurate uncertainty quantification, for forward and inverse problems.



Latent Evolution of PDEs with UQ (LE-PDE-UQ)

- The latent evolution model g deterministically predicts both a global latent vector **z** that encodes the state, and a global latent vector $\mathbf{z}_\sigma$ that encodes the uncertainty. As needed, they decode respectively to the predicted state and predicted uncertainty.
- It allows evolution of state and propagation of uncertainty for long-term rollout. And it can also combine with Deep Ensembles [3] to capture epistemic uncertainty.

Learning objective

$$L = \frac{1}{K}\sum_{k=1}^{K}(L^k_{\text{multi-step}} + L^k_{\text{recons}} + L^k_{\text{consistency}}).$$

$$\text{where} \begin{cases} L^k_{\text{multi-step}} = \sum_{m=1}^{M} \alpha_m \left[\frac{(\hat{U}^{k+m} - U^{k+m})^2}{2\hat{U}_\sigma^{k+m}} + \log \hat{U}_\sigma^{k+m}\right] \\ L^k_{\text{recons}} = \ell(h(q(U^k)[z]), U^k) \\ L^k_{\text{consistency}} = \sum_{m=1}^{M} \frac{\|(g(\cdot, r(p))^{(m)} \circ q(U^k))[z] - q(U^{k+m})[z]\|_2^2}{\|q(U^{k+m})[z]\|_2^2} \end{cases}$$

Inverse mode

Given a specified objective $L_d[p, U^0] = \sum_{k=k_s}^{k_c} \ell(U^t)$ which is a discretized version of $L_d[a, \partial X]$, we define the objective:

$$L_d[p, U^0] = \sum_{m=k_s}^{k_c} \ell_d(\hat{U}^m(p, U^0))$$

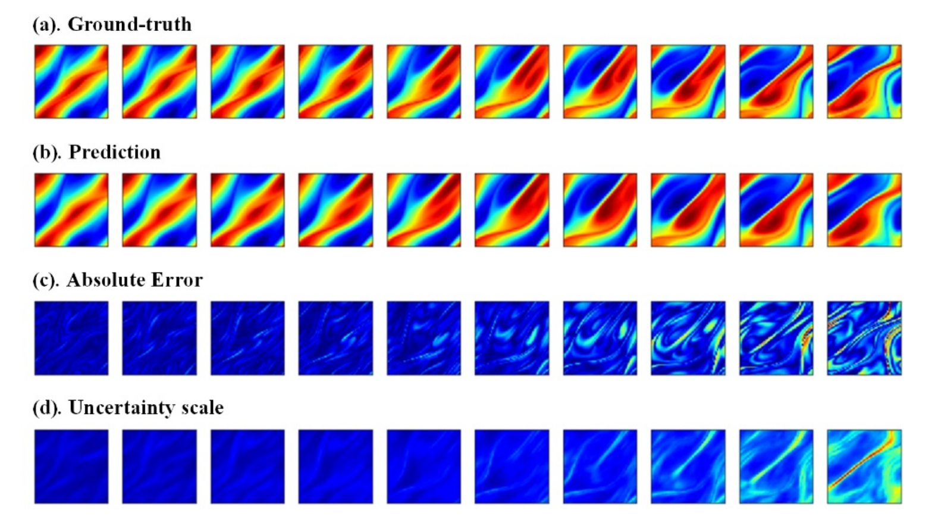
[3] Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. *NeurIPS* (2017).

Result

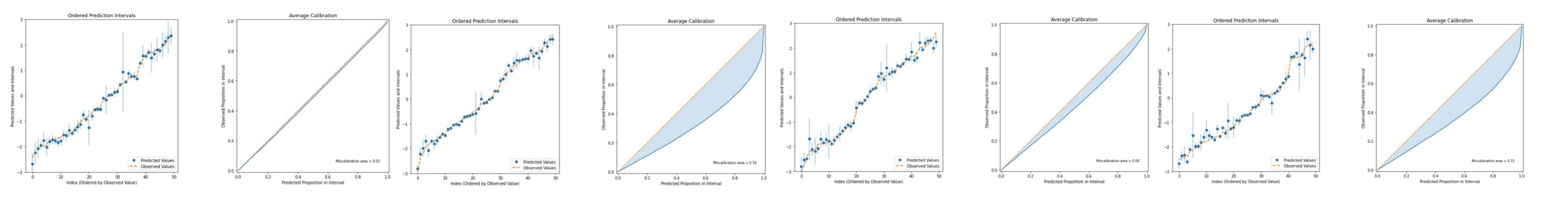
2D forward problems

Comparison of LE-PDE-UQ with multiple baselines

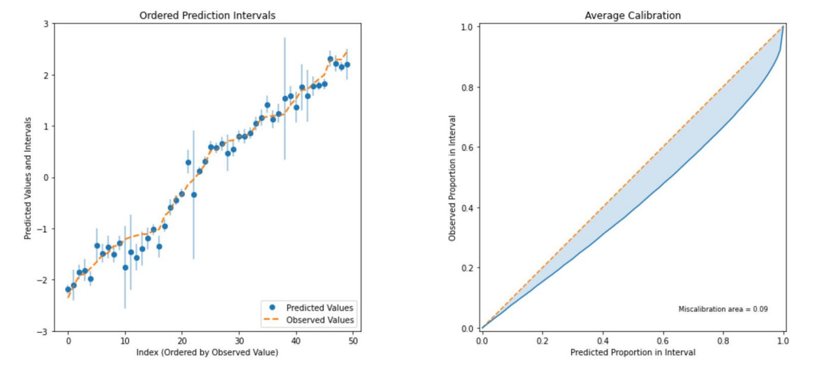
	MA	MACE	RMSCE	L2	MAE
Bayes layer with latent	0.0445	0.0440	0.0500	0.2345	0.2051
Bayes layer without latent	0.2381	0.2357	0.2665	0.2105	0.1830
Dropout, L2=0	0.1778	0.1760	0.1979	0.2079	0.1938
Dropout, L2=1e-5	0.1924	0.1905	0.2143	0.2092	0.1958
Dropout, L2=1e-4	0.2317	0.2294	0.2588	0.2458	0.2320
Dropout, L2=1e-3	0.3281	0.3248	0.3704	0.3534	0.3428
NoLatent (single, with σ)	0.1045	0.1035	0.1175	0.2053	0.1817
NoLatent (ensemble, without σ)	0.2118	0.2096	0.2355	0.1939	0.1657
NoLatent (ensemble, with σ)	0.0602	0.0596	0.0662	0.1939	0.1657
Latent (single, without σ)	-	-	-	**0.1890**	0.1613
Latent (single, with σ)	0.0576	0.0570	0.0649	0.2108	0.1811
Latent (ensemble, without σ)	0.1823	0.1805	0.2024	**0.1895**	**0.1608**
Latent (**ours**, ensemble, with σ)	**0.0142**	**0.0141**	**0.0160**	**0.1895**	**0.1608**



Ablations


LE-PDE-UQ Without deterministic Latent full + L1 Latent full + no z_σ^t

Latent vs. no Latent



Autoregressive rollout:

	MA	MACE	RMSCE	L2	MAE
NoLatent (ensemble, with σ)	0.0602	0.0596	0.0662	0.1939	0.1657
Latent (**ours**, ensemble, with σ)	0.0142	0.0141	0.0160	0.1895	0.1608

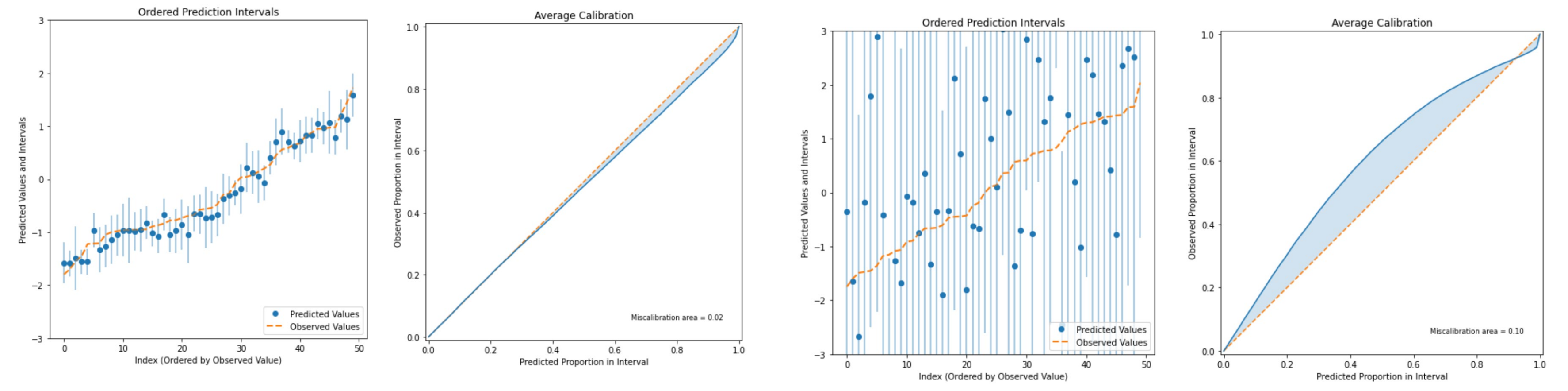
We perform teacher-forcing (provide ground-truth as input) at each rollout step:

	MA	MACE	RMSCE	L2	MAE
NoLatent (ensemble, with σ)	0.0260	0.0258	0.0289	0.1670	0.1420
Latent (**ours**, ensemble, with σ)	0.0101	0.0100	0.0124	0.1562	0.1296

2D inverse problems

Predicting initial states (t=1-9), given observation of future states (t=10-20):

	MA	MACE	RMSCE	L2	MAE
NoLatent (ensemble, with σ)	0.0929	0.0920	0.1055	1.5255	1.2580
Latent (**ours**, ensemble, with σ)	0.0224	0.0222	0.0264	0.1863	0.1505


Latent (ours, ensemble, with σ) NoLatent (ensemble, with σ)

Discussion

In this work, we introduced LE-PDE-UQ to quantify uncertainty in time-dependent PDEs with deep learning surrogates. It leverages latent vectors for precise predictions and robust uncertainty estimates in both forward and inverse problems. Excelling in long-term auto-regressive rollouts, LE-PDE-UQ outperforms major baselines, ensuring precision and stability in uncertainty quantification.