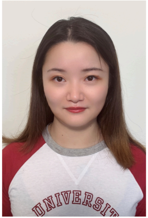
Embed & Emulate: Learning to estimate parameters of dynamical systems with uncertainty quantification

Ruoxi Jiang and Rebecca Willett

University of Chicago



Background: Parameter Estimation

Goal: estimate parameters ϕ of a physics simulation that best fit real data:

$$\boxed{\mathbf{Z}} = \boxed{H}(\phi) + \boxed{\boldsymbol{\eta}}$$

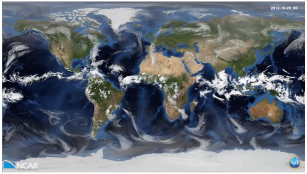
Observation

Noise

Physics based *computationally complex* simulator

In climate forecasting, H corresponds to the climate models.

Uncertainty quantification of $\hat{\phi}$ is vital.



Overview of Existing Work

Classical method:[1] For some predefined moment function m

$$\hat{\phi} = \arg\min_{\phi} \|m(\mathbf{Z}) - m(H(\phi))\|^2_{\boldsymbol{\Sigma}[m(\mathbf{Z})]}.$$

- need expert knowledge to choose m, requires repeated (slow) runs of H for each new observation $\mathbf{Z}$, gradient-free optimization methods like ensemble Kalman inversion highly dependent on prior p_{ϕ}.

Supervised regression: Learn a neural Network (f_θ) so that

$$\hat{\phi} = f_\theta(\mathbf{Z}).$$

- difficult to get accurate uncertainty estimates.

[1] E.g.: Schneider, Tapio, Lan, Shiwei, Stuart, Andrew, et al. "Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations". *Geophysical Research Letters*.

Overview of Existing Work

Vanilla Emulator:[2] Learn a neural network $\hat{H}_\theta$ so that $\mathbf{Z} \approx \hat{H}_\theta(\phi)$ and solve

$$\hat{\phi} = \arg\min_{\phi} \|m(\mathbf{Z}) - m(\hat{H}_\theta(\phi))\|^2_{\mathbf{\Sigma}[m(\mathbf{Z})]}.$$

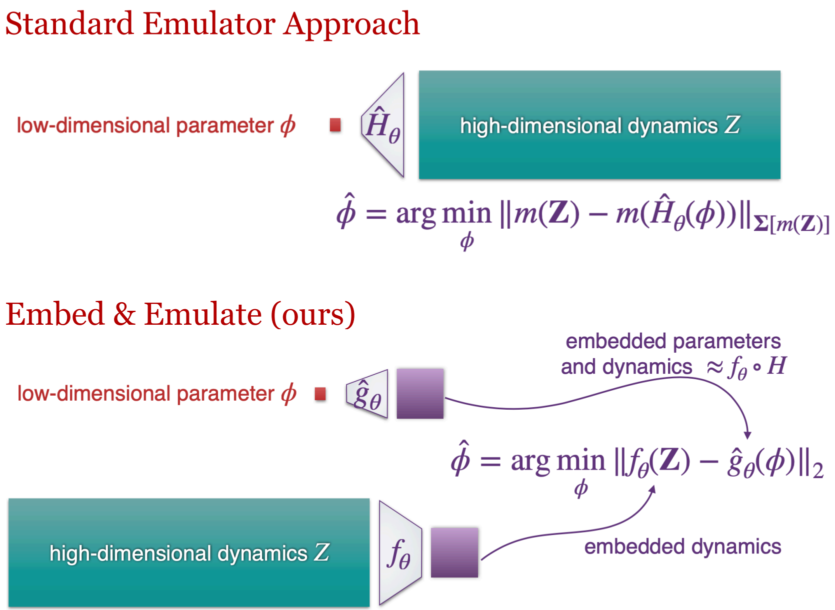
- require known m, must emulate high-dimensional dynamics, and very sensitive to initial conditions.

Simulation-based inference:[3] Directly estimate posterior $p(\phi|\mathbf{Z})$ using neural conditional density estimation

- can be unstable in high demensions.

[2] E.g.: Raissi, Maziar, Perdikaris, Paris, and Karniadakis, George Em. "Physics informed deep learning (part I): Data-driven solutions of nonlinear partial differential equations". *arXiv preprint arXiv:1711.10561*.

[3] E.g.: Lueckmann, Jan-Matthis, Boelts, Jan, Greenberg, David, et al. "Benchmarking simulation-based inference".


Standard Emulator Approach
low-dimensional parameter ϕ
$\hat{H}_\theta$
high-dimensional dynamics Z
$\hat{\phi} = \arg\min_{\phi} \|m(\mathbf{Z}) - m(\hat{H}_\theta(\phi))\|_{\boldsymbol{\Sigma}[m(\mathbf{Z})]}$
Embed & Emulate (ours)
embedded parameters
and dynamics $\approx f_\theta \circ H$
low-dimensional parameter ϕ
$\hat{g}_\theta$
$\hat{\phi} = \arg\min_{\phi} \|f_\theta(\mathbf{Z}) - \hat{g}_\theta(\phi)\|_2$
high-dimensional dynamics Z
f_θ
embedded dynamics

Embed & Emulate (ours)

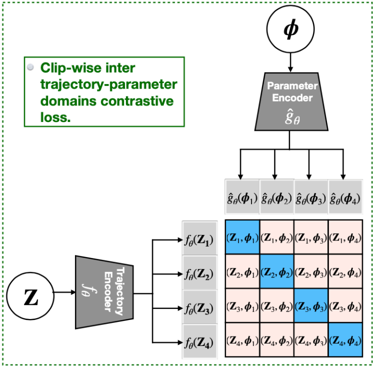


Figure 1: Inter-domain contrastive learning scheme: Diagonals are dot products between representations of "positive" pairs $(\mathbf{Z}_i, \phi_i)$.

- We design an "emulator" that fits well in the context of parameter estimation problem.
- We use CLIP-wise loss to align the metric space of the "emulator" and the embedding network.
- We use contrastive loss to capture intra-domain structural information to learn meaningful embeddings.

Example estimating Lorenz-96 parameters

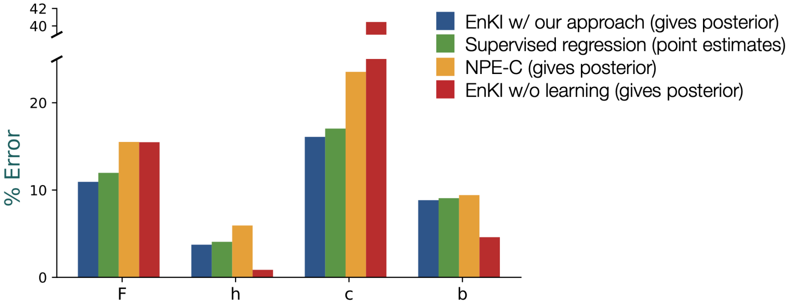


Table 1: Computation time for 500 training samples + 200 testing samples (including time to generate training data, reported in minutes).

	EnKI *w/* our approach	Supervised Regression	NPE-C	EnKI *w/o* learning
Total	52.0 (0.87 h)	43.0 (0.72 h)	52.0 (0.87 h)	8,000 (5.5 d)

Summary

Thank you! Code is available at:

https://github.com/roxie62/Embed-and-Emulate