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

THE UNIVERSITY OF CHICAGO

Ruoxi Jiang Rebecca Willett

Background: Parameter estimation

Goal: We aim to learn a mapping from observations of a multichannel dynamical system, $\mathbf{Z} \in \mathbb{R}^{T \times d}$, to the underlying parameters $\boldsymbol{\phi} \in \mathbb{R}^k$:

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

- H is a physics-based computationally complex simulator with no analytical expression;
- We seek quantifiable uncertainty measures instead of point estimates to cover the full range of possible outcomes.
- η is some noise vector.

Real-world Application: In climate projection, H corresponds to the climate models implemented with computationally demanding software systems, and the goal is to learn parameterization schemes from global observations and high-resolution simulations. Uncertainty quantification is vital, as a small error in $\hat{\boldsymbol{\phi}}$ might lead to a dramatic change in the forecasts of dynamics.

Baseline: Moments with physics-based simulators

Objective function: Schneider et al. [2] tries to find $\hat{\boldsymbol{\phi}}$ by minimizing the Mahalanobis distance:

$$J_{\text{moment}}(\boldsymbol{\phi}; \mathbf{Z}) := \|m(\mathbf{Z}) - m(H(\boldsymbol{\phi}))\|_{\boldsymbol{\Sigma}[m(\mathbf{Z})]}.$$

- $m(\cdot)$ is the time-average of a predefined moments function composed of order statistics of different spatial channels of the dynamics.
- $\boldsymbol{\Sigma}[m(\mathbf{Z})]$ is a diagonal matrix with the diagonals correspond to temporal variance of the moments function.

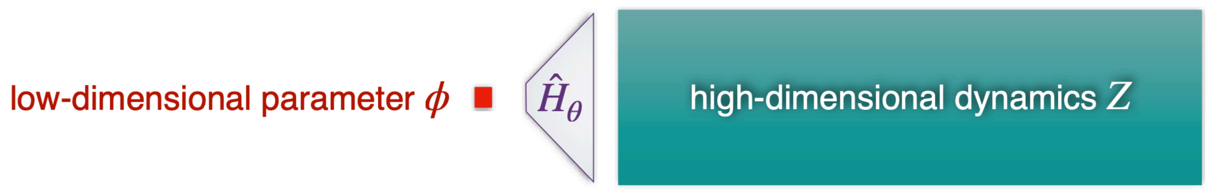
Optimization method: Schneider et al. [2] proposes the use of Ensemble Kalman Inversion (EnKI) [1] to address $J_{\text{moment}}(\boldsymbol{\phi})$. With a specified prior $p(\boldsymbol{\phi})$, the EnKI iteratively updates an ensemble of estimates and results in an estimated posterior distribution.

Challenges:

1. **Computationally expensive:** Running H is required for each particle at each iteration.
2. **Requiring domain expertise:** Defining $m(\cdot)$ requires knowledge of the underlying physical system.
3. **Requiring prior information:** Bad priors might result in bad estimates.

Standard Emulator Approach: Learn $\hat{H}_\theta$

Instead of running costly numerical simulations H, learning a neural network $\hat{H}_\theta$ so that $\mathbf{Z} \approx \hat{H}_\theta(\boldsymbol{\phi})$.

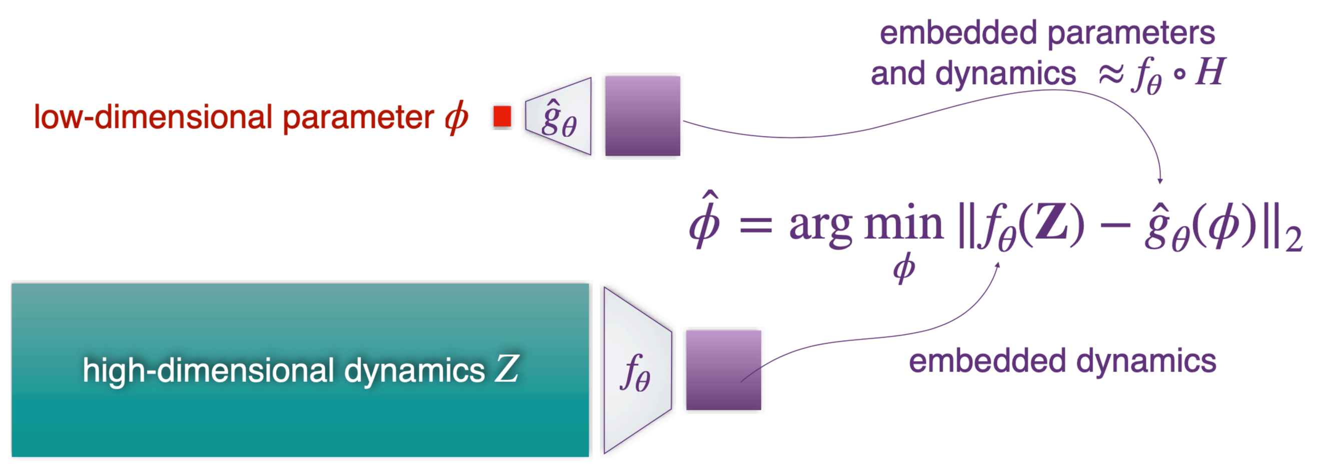


$$\hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\phi}} \|m(\mathbf{Z}) - m(\hat{H}_\theta(\boldsymbol{\phi}))\|_{\boldsymbol{\Sigma}[m(\mathbf{Z})]}$$

- Require known $m(\cdot)$, must emulate high-dimensional statistics, and very sensitive to initial conditions.

Our approach: Embed & Emulate

We seek to learn **feature embeddings** f_θ of the dynamics jointly with **an emulator** $\hat{g}_\theta$ that can replace high-cost simulators. Instead of using $m(\cdot)$, $\hat{g}_\theta$ learns to emulate the map $g := f_\theta \circ H$.



The Embed & Emulate framework

We design an "emulator" that fits well in the context of the parameter estimation problem.

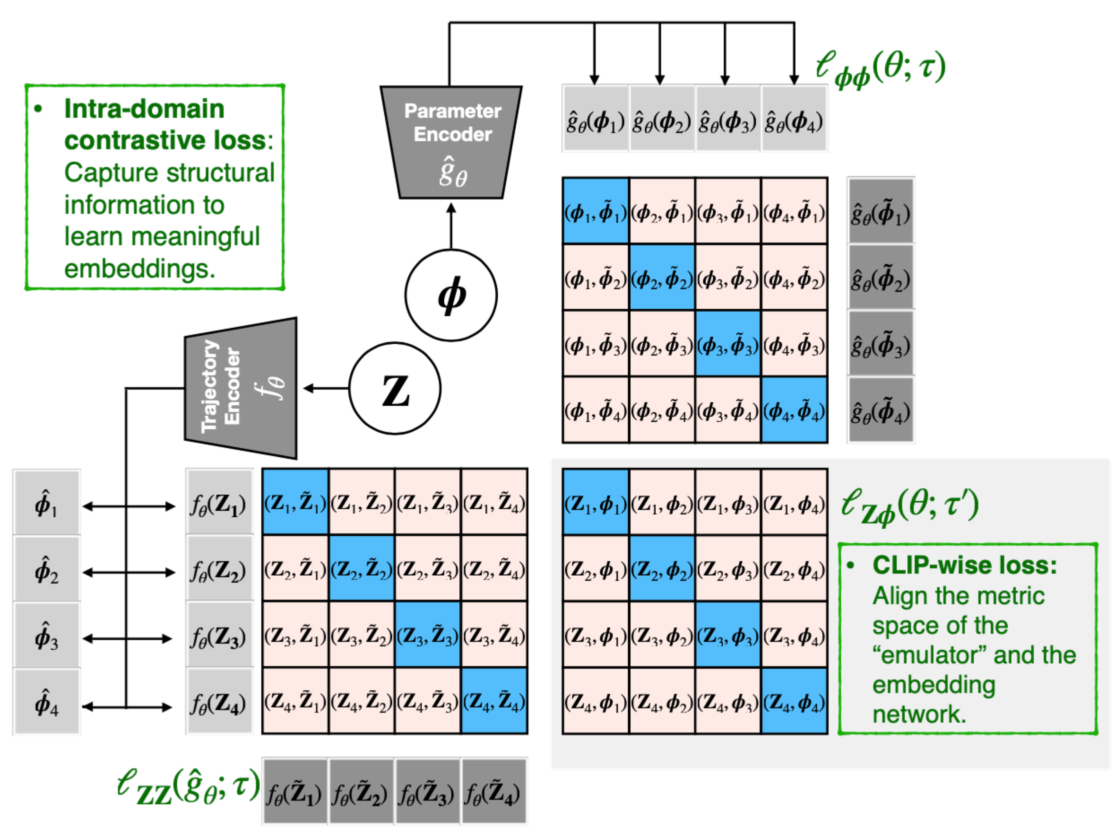


Figure 1. Contrastive learning schemes: Within each block, diagonals are dot products between representations of "positive" pairs $(\mathbf{Z}_i, \tilde{\mathbf{Z}}_i)$, $(\boldsymbol{\phi}_i, \tilde{\boldsymbol{\phi}}_i)$, and matched $(\mathbf{Z}_i, \boldsymbol{\phi}_i)$.

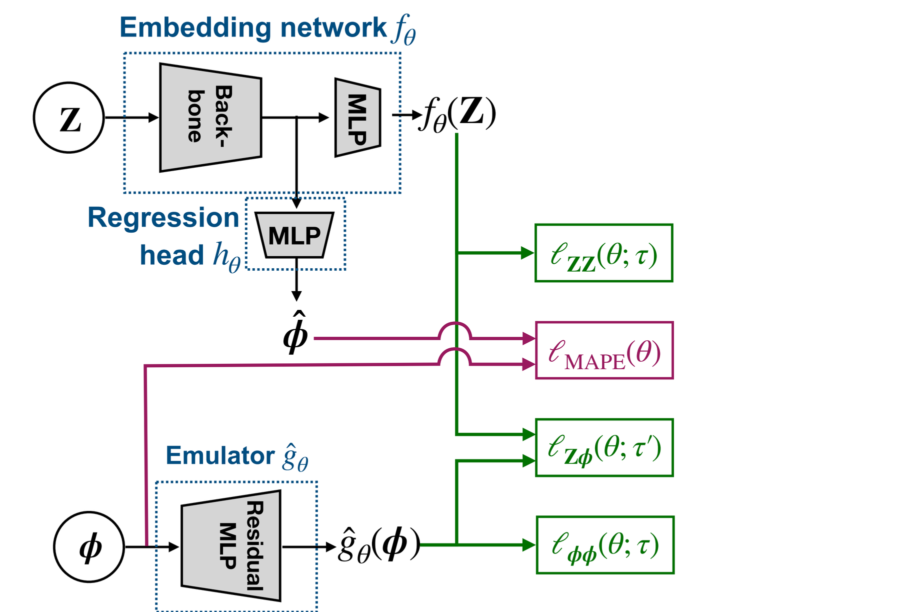


Figure 2. Components of our network and loss functions. Green blocks correspond to contrastive losses, and the purple block corresponds to an added regression head which is shown to be empirically useful.

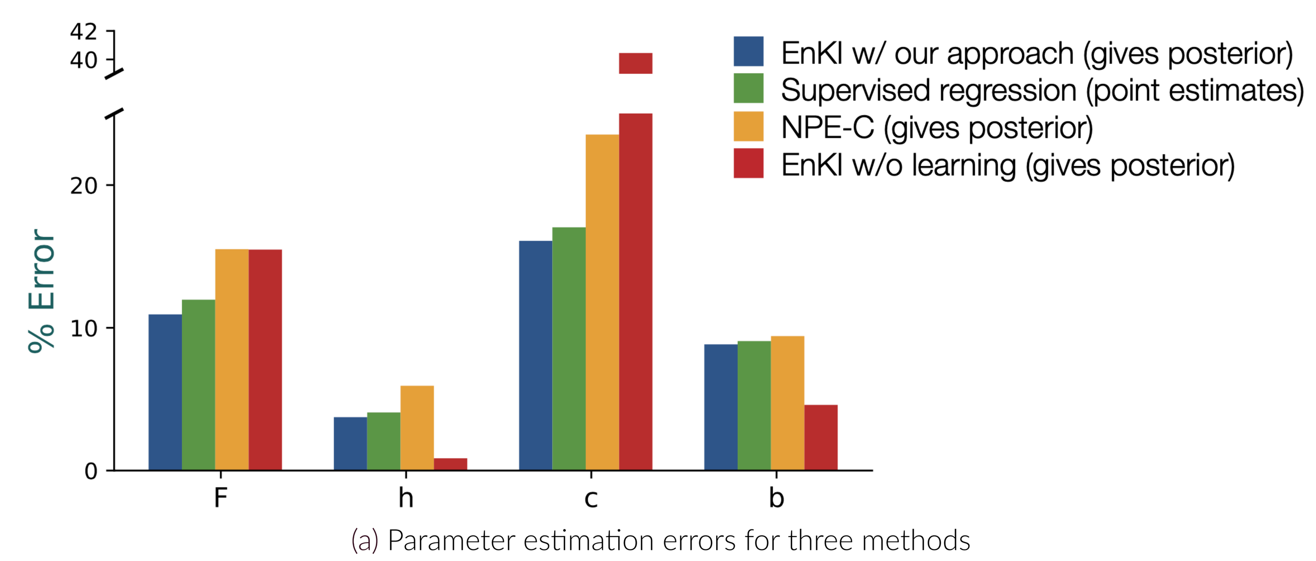
Experiments

We tested on a coupled 396-dimensional Lorenz 96 system. Training data is simulated with length $T = 100$ and $\mathrm{d}t = 0.1$, resulting in $\mathbf{Z}_i \in \mathbb{R}^{396 \times 1{,}000}$. Each tested observation is with length $T = 1{,}000$.

- **Supervised regression baseline:** We propose another baseline by learning the direct mapping from $\mathbf{Z}_i$ to $\hat{\boldsymbol{\phi}}_i$ to provide only point estimates.
- **Neural Posterior Estimation (NPE-C):** We compare against directly estimating posterior $p(\boldsymbol{\phi}|\mathbf{Z})$ using neural conditional density estimation which can be unstable in high dimensions.

Setup: We use a Gaussian fixed prior $p_{\boldsymbol{\phi},\text{fixed}}$ for the baseline EnKI w/o learning. For Embed & Emulate, we adopt an empirical Bayes approach: alter the mean of $p_{\boldsymbol{\phi},\text{empB}}$ with the estimate from the regression head.

1st: Higher quality estimates with lower computation time.



(a) Parameter estimation errors for three methods

Figure 3. **Averaged MAPE for varying training size.** Embed & Emulate is able to achieve a lower error than both the EnKI approach based on a fixed moment vector objective and classical numerical solver [2] and a straightforward supervised regression approach that is unable to produce uncertainty estimates.

	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)

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

	$F\downarrow$	$h\downarrow$	$c\downarrow$	$b\downarrow$
EnKI *w/o* Learning	0.910	0.019	2.443	0.393
EnKI *w/* Embed & Emulate	**0.615**	**0.073**	**1.561**	**0.720**
Supervised Regression	0.707	0.104	1.785	0.917
NPE-C	0.844	0.106	2.117	0.853

Table 2. **Continuous Ranked Probability Score (CRPS) evaluated on 200 test samples.** The errors of the uncertainty estimates are almost always lower for Embed & Emulate than for an EnKI method using a classical numerical solver or for a supervised regression baseline.

2nd: Visualize uncertainty with noisy observations.

The noise vector $\boldsymbol{\eta} \sim \mathcal{N}(0, r\boldsymbol{\Gamma})$, where $\boldsymbol{\Gamma}$ is the temporal covariance of the trajectory $\mathbf{Z}$, and r is a scaling value.

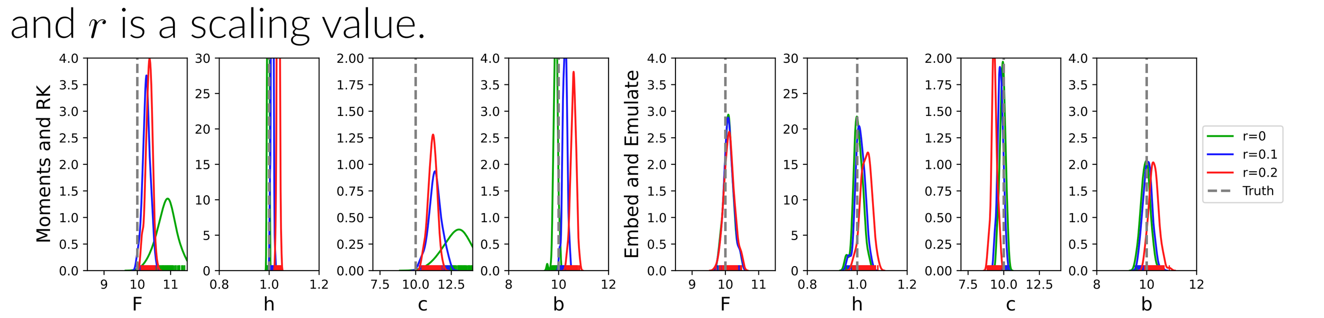


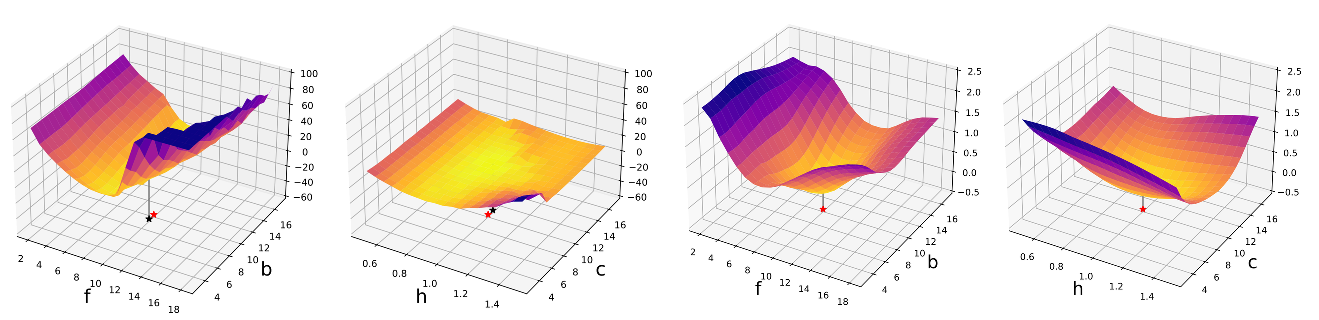
Figure 4. **Impact of observation noise.** Reconstructed posterior distributions, comparing a classical numerical solver (Runge-Kutta) plugged into EnKI to minimize J_{moment} (left) with Embed & Emulate (right). Both variants of EnKI use the fixed prior $p_{\boldsymbol{\phi},\text{fixed}}$ and are run for 70 iterations with 100 particles. **Results.** Embed & Emulate (right) produces posterior estimates that are consistent over a range of noise levels, while the baseline using J_{moment} is much more sensitive to variations in noise levels r.

3rd: Ablation study on the roles of regression head.

	$F\downarrow$	$h\downarrow$	$c\downarrow$	$b\downarrow$
(a) No regression head ($p_{\boldsymbol{\phi},\text{fixed}}$)	16.75 (2.62)	6.02 (1.63)	22.16 (6.35)	6.60 (1.86)
(b) Embed & Emulate ($p_{\boldsymbol{\phi},\text{fixed}}$)	8.39 (1.12)	1.77 (0.82)	15.82 (1.61)	3.85 (0.91)
(c) Embed & Emulate ($p_{\boldsymbol{\phi},\text{empB}}$)	**3.39 (1.03)**	**1.21 (0.76)**	**4.53 (1.52)**	**3.02 (0.90)**

Table 3. **Average MAPE (MdAPE, median absolute percentage error over 200 test instances (n = 4,000)**. (a) f_θ and $\hat{g}_\theta$ correspond to a generic emulator trained without the regression loss ℓ_{MAPE} and the original prior $p_{\boldsymbol{\phi},\text{fixed}}$; (b) f_θ and $\hat{g}_\theta$ correspond to the emulator learned with our Embed & Emulate framework and the original prior $p_{\boldsymbol{\phi},\text{fixed}}$; and (c) Embed & Emulate framework and the empirical Bayes prior $p_{\boldsymbol{\phi},\text{empB}}$. **Results:** Having the regression component of the loss complement the contrastive losses yields a substantial improvement in parameter estimation accuracy.

4th experiment: Visualize the objective function.



(a) Marginal heatmap of $J_{\text{moment}(\boldsymbol{\phi};\mathbf{Z}^*)}$ using Runge-Kutta (b) Marginal heatmap of $\|f_\theta(\mathbf{Z}^*) - \hat{g}_\theta(\boldsymbol{\phi})\|_2$.

Figure 5. **Heatmap visualization showing values of objective functions.** (a) the marginal heatmap of predefined moments objective function $J_{\text{moment}}(\boldsymbol{\phi}; \mathbf{Z}^*)$; and (b) the marginal heatmap of $\|f_\theta(\mathbf{Z}^*) - \hat{g}_\theta(\boldsymbol{\phi})\|_2$ with Embed & Emulate using learned emulator. The red stars in both plots show the locations of the true parameters $\boldsymbol{\phi}^*$, while the black stars show the locations of the points with the minimum function value.

References

[1] Marco A Iglesias, Kody J H Law, and Andrew M Stuart. Ensemble kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, mar 2013. doi: 10.1088/0266-5611/29/4/045001. URL https://doi.org/10.1088%2F0266-5611%2F29%2F4%2F045001.

[2] Tapio Schneider, Shiwei Lan, Andrew Stuart, and Joao Teixeira. Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, 44(24):12–396, 2017.