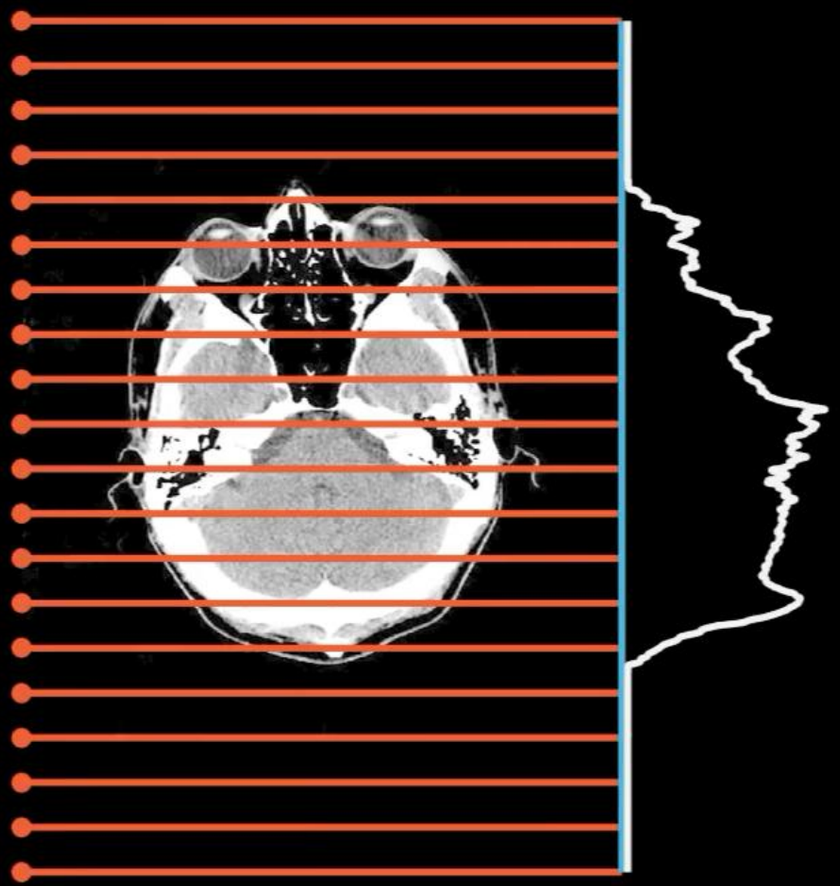
Machine Learning for Inverse Problems in Climate Science

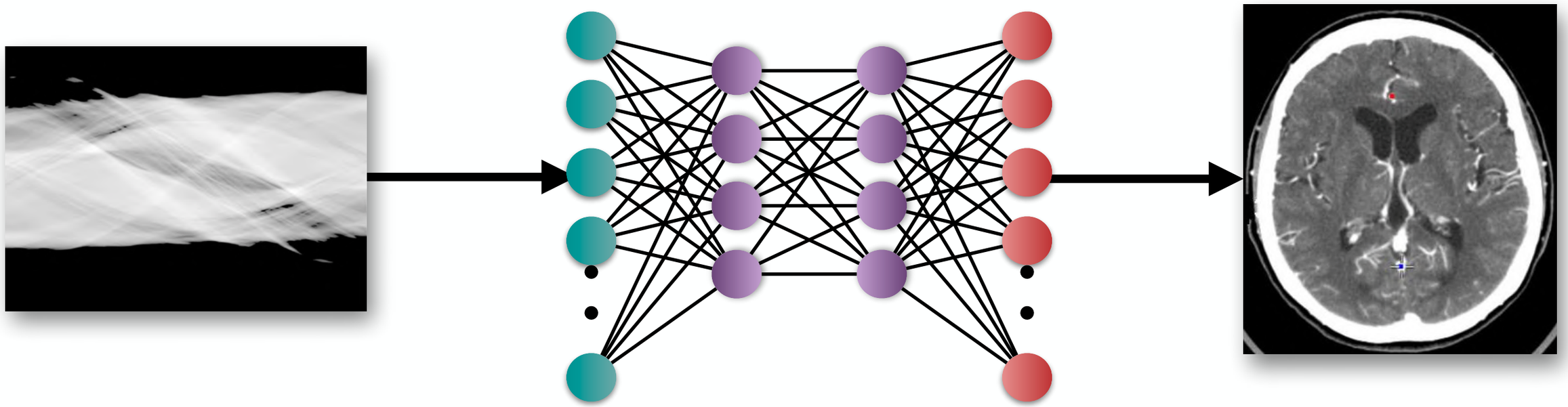
Rebecca Willett, University of Chicago



Video credit: Samuli Siltanen https://www.youtube.com/watch?v=q7Rt_OY_7tU

Can machine learning help reconstruct images?

Train deep neural network to reconstruct CT images from sinogram measurements



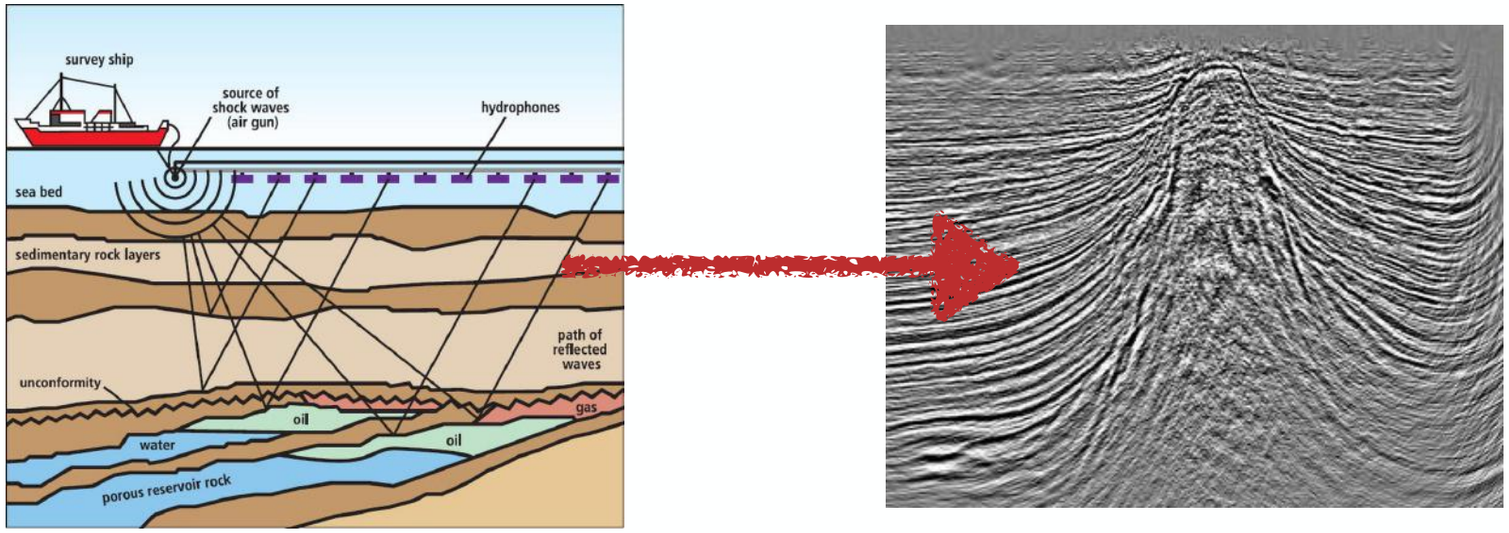
This approach can require ***many*** training samples.

It also ignores everything we know about the data collection process.

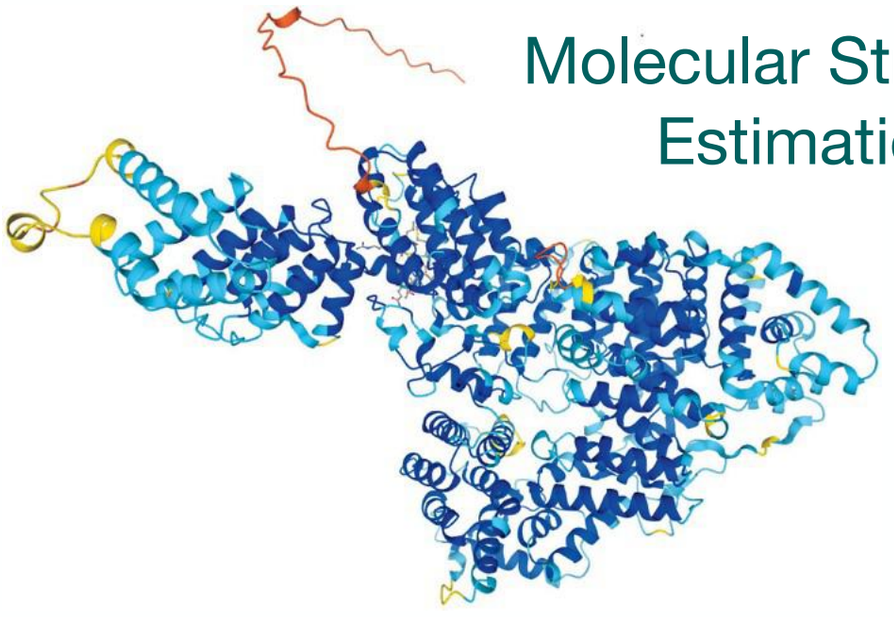
Zhu, Liu, Rosen, Rosen, 2017; Arridge, Maass, Öktem, Schönlieb, 2019; Ongie, Jalal, Metzler, Baraniuk, Dimakis, Willett, 2020; Akçakaya, Yaman, Chung, Ye, 2022; Sahel, Bryan, Cleary, Farhi, Eldar, 2022; Kamilov, Bouman, Buzzard, Wohlberg, 2022

There are *many* settings in which we have both training data and physical models.

Seismic Imaging



Molecular Structure Estimation



Climate Forecasting

Also fluid dynamics, turbulence, particle accelerators, scattering, automatic control…

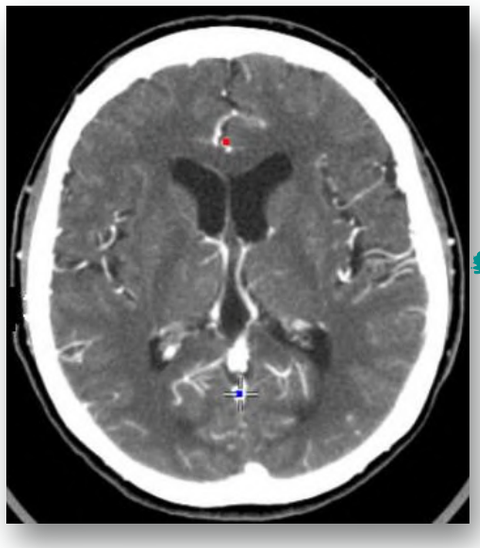
Physics-based models can inform neural network architectures and training, improving the reliability, efficiency and interpretability of ML systems.

Example: linear inverse problems in imaging

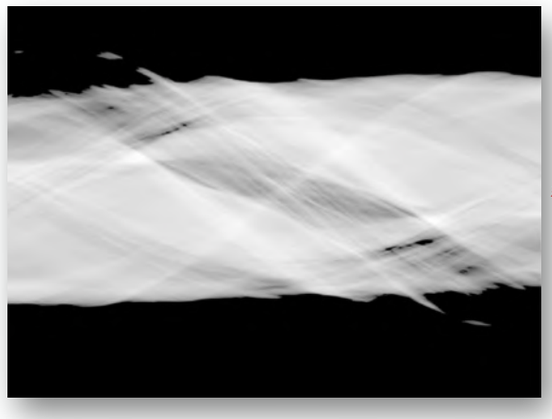
Observe: $y = Hx + \varepsilon$

Goal: Recover x from y

x



y



"forward model" H reflects the physics of the imaging system

Image reconstruction by supervised learning

1. Collect training data pairs (x_i, y_i) using a known forward model:

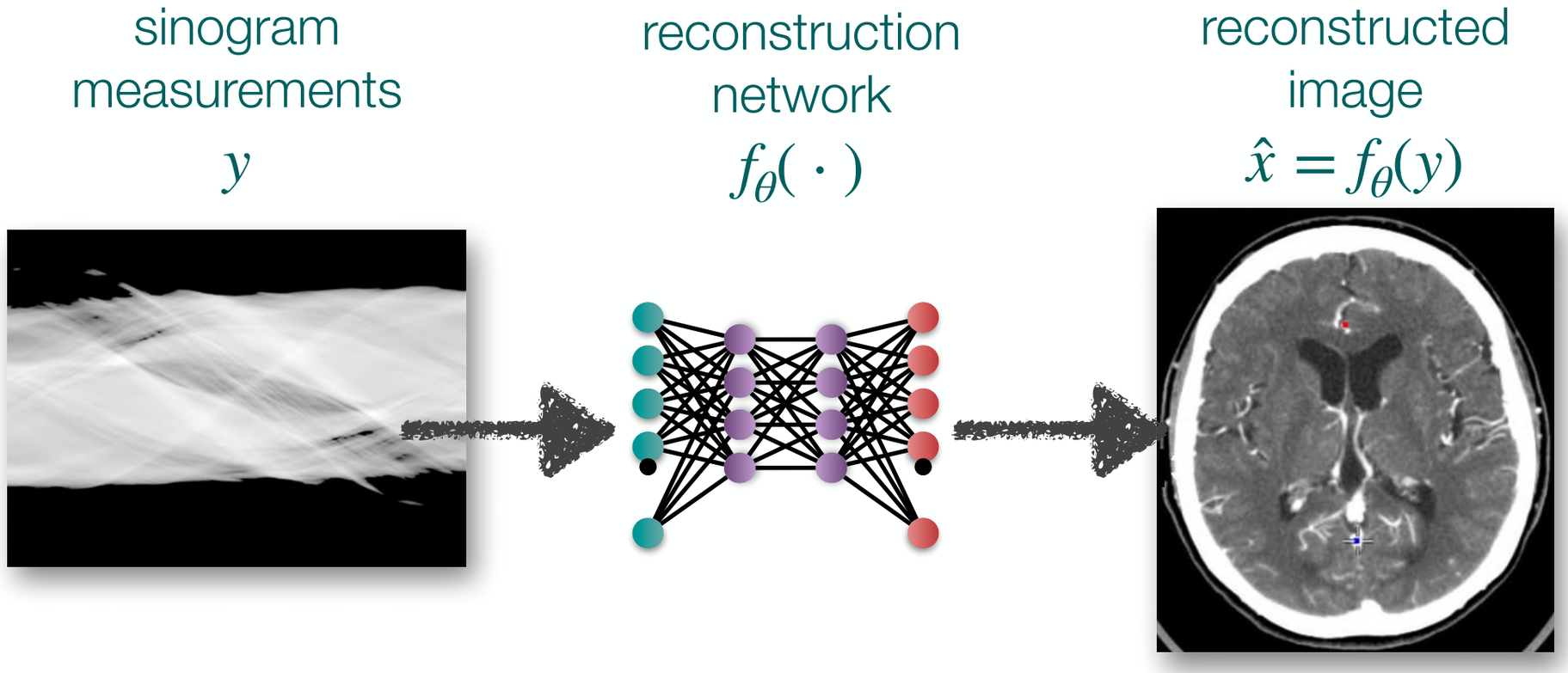
$$y_i = Hx_i + \varepsilon_i$$

2. Train a reconstruction network f_θ by minimizing over a loss; e.g.

$$\min_\theta \sum_i \|x_i - f_\theta(y_i)\|_2^2$$

3. Reconstruct new measurements y by $\hat{x} = f_\theta(y)$

e.g.,
CT Scan



Can we design the neural network f_θ to reflect our knowledge of the underlying physics (i.e. H)?

Yes! To do so, we leverage decades of accumulated knowledge of inverse problems, data assimilation, and optimization

Arridge, Maass, Öktem, Schönlieb, 2019 ; Ongie, Jalal, Metzler, Baraniuk, Dimakis, Willett, 2020; Monga, Li, Eldar, 2021

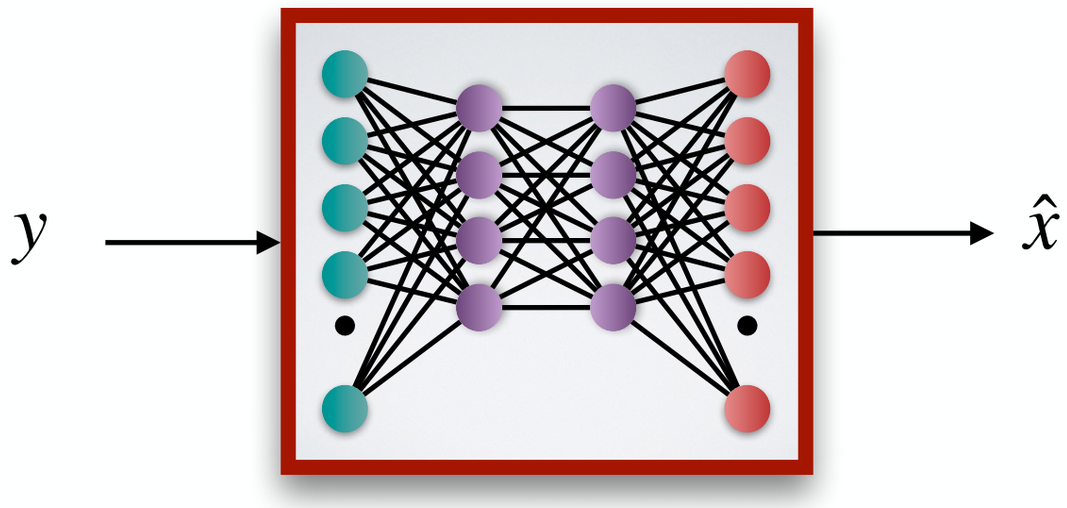
Classical approach to solving inverse problems

$$\underset{x}{\text{minimize}} \; \|Hx - y\|^2 + R(x)$$

Data fit term measures how well image x fits observation y, taking physical model H into account

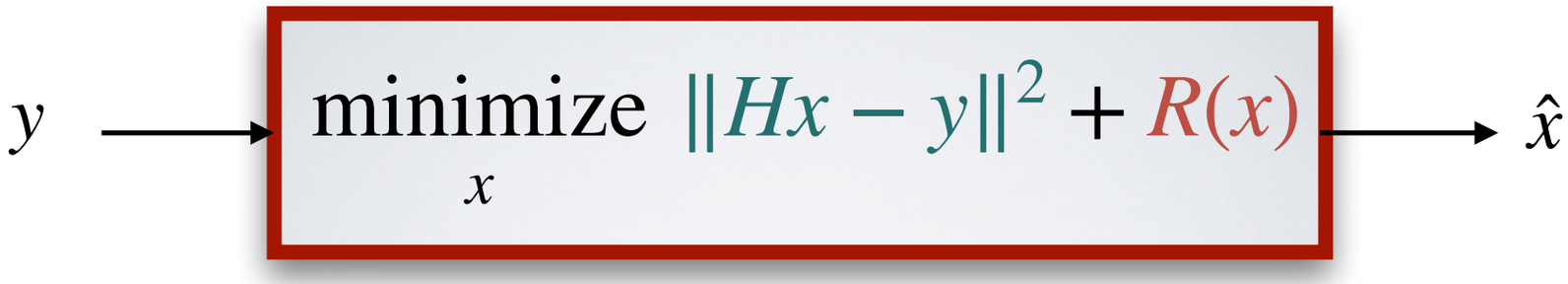
Regularization function measures to what extent an image x has expected geometry (e.g. smoothness or sharp edges)

Learning to reconstruct



Instead of using choosing $R(x)$ a priori based on smoothness or geometric models,
can we learn a regularizer using training data?

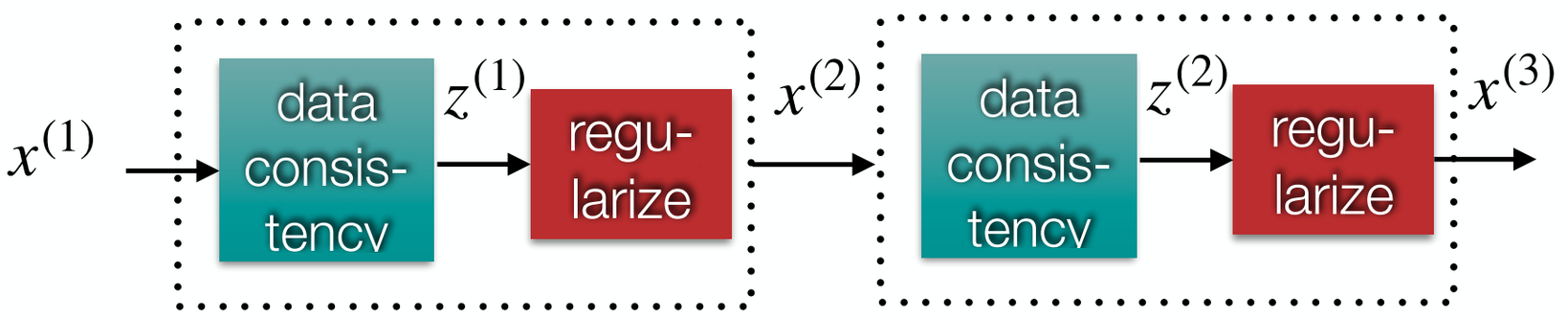
Optimization framework



for $k = 1, 2, \ldots$

$$z^{(k)} = x^{(k)} - \eta H^\top (Hx^{(k)} - y)$$ **data consistency step**

$$x^{(k+1)} = \text{regularize}(z^{(k)}, R)$$ **regularization step (e.g. proximal operator)**



repeat until convergence

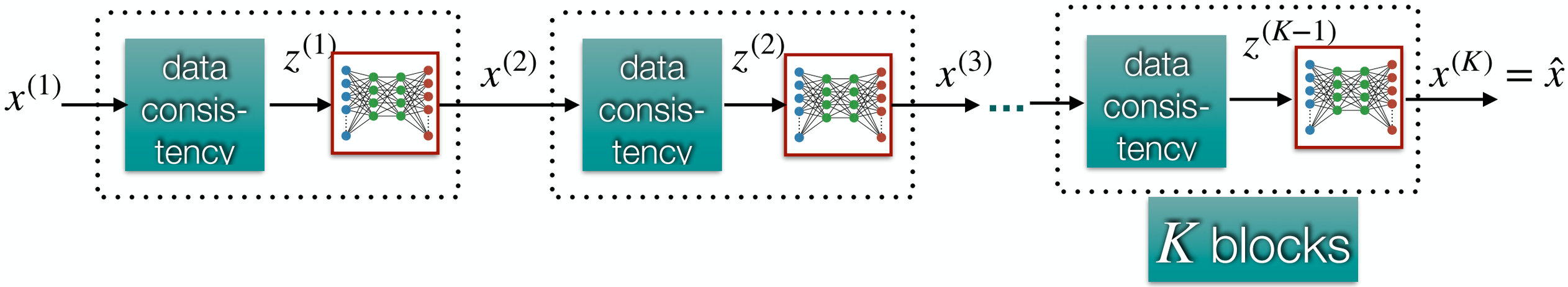
Deep Unrolling

$$y \longrightarrow \boxed{\underset{x}{\text{minimize}} \ \|Hx - y\|^2 + R(x)} \longrightarrow \hat{x}$$

for $k = 1, 2, \ldots$

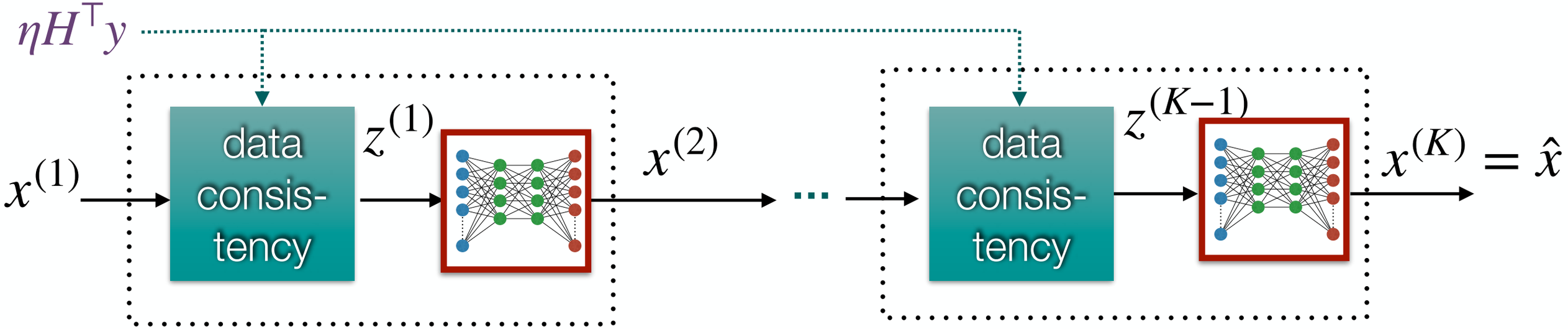
$$z^{(k)} = x^{(k)} - \eta H^\top (Hx^{(k)} - y)$$ **data consistency step**

$$x^{(k+1)} = \textbf{CNN}(z^{(k)})$$ **regularization step**



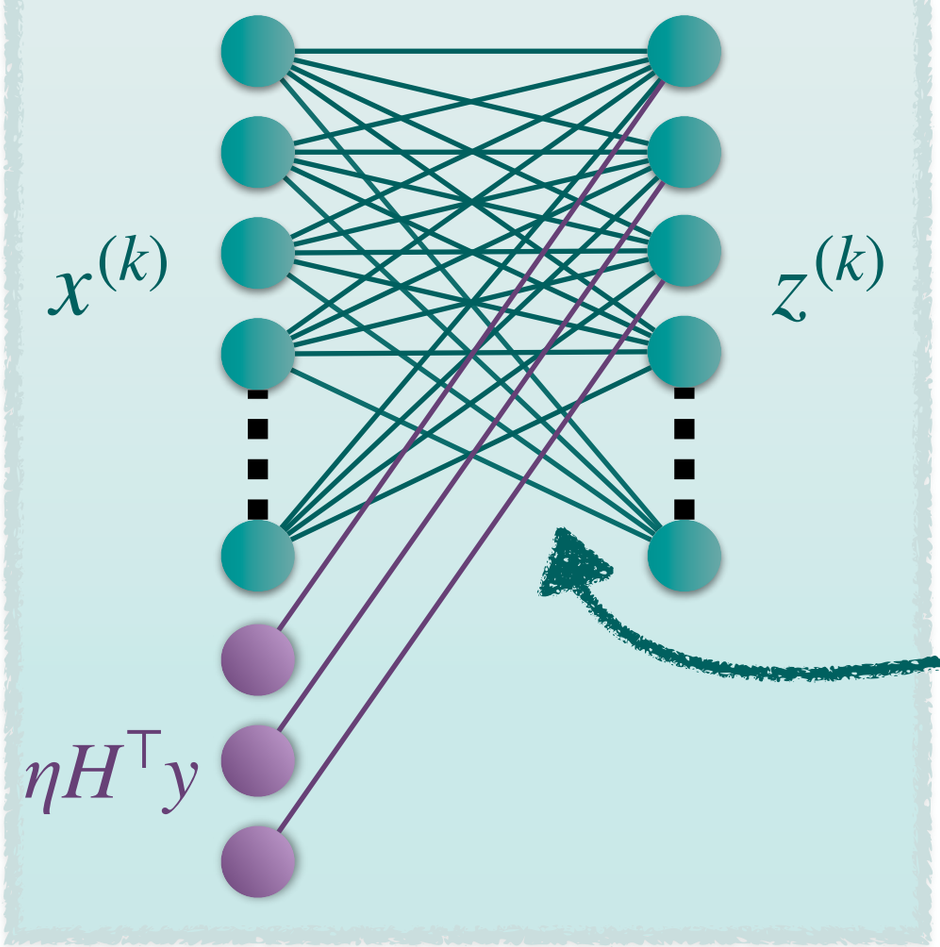
"Unroll" K iterations, train end-to-end in a supervised manner

Physics-guided neural network architecture



data consis-tency

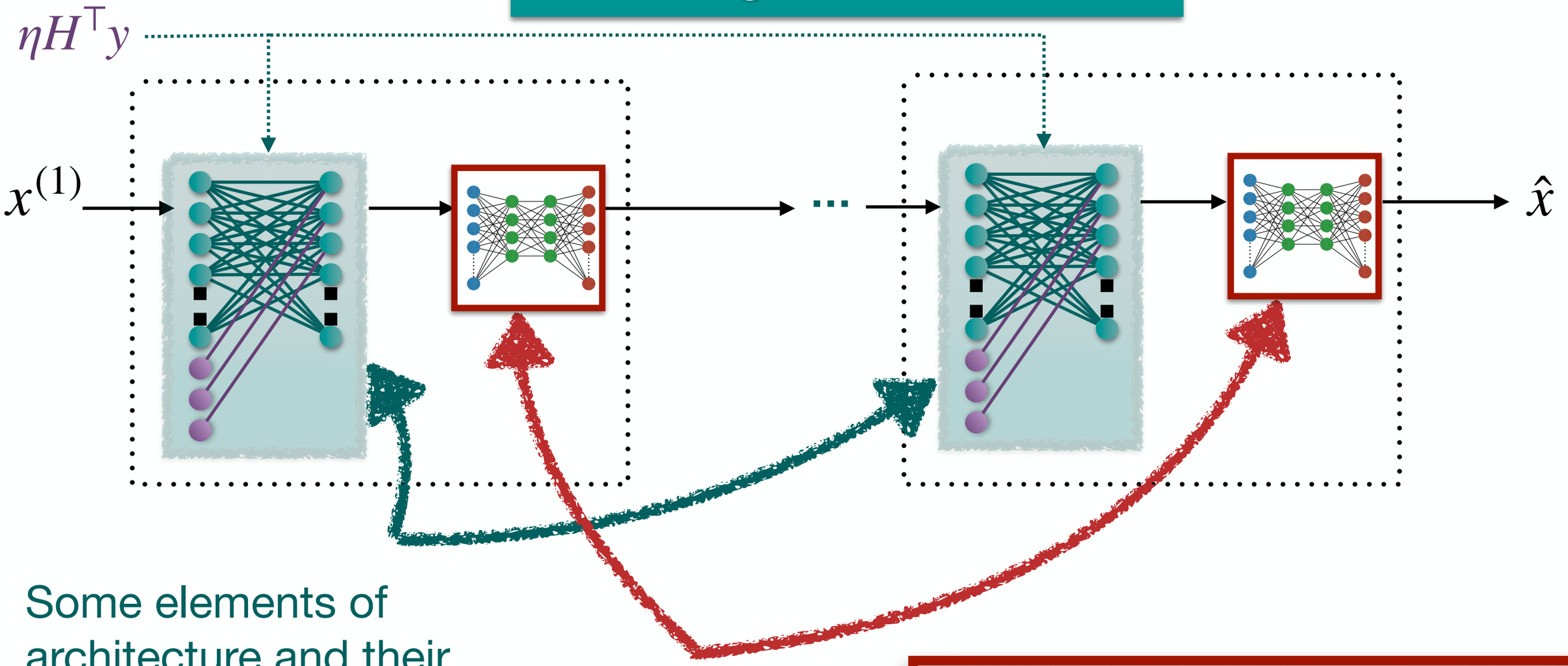
$$(I - \eta H^\top H)x^{(k)} + \eta H^\top y$$



weight matrix = $I - \eta H^\top H$ is determined by physical model instead of learned from data

Physics-guided neural network architecture

One big neural network

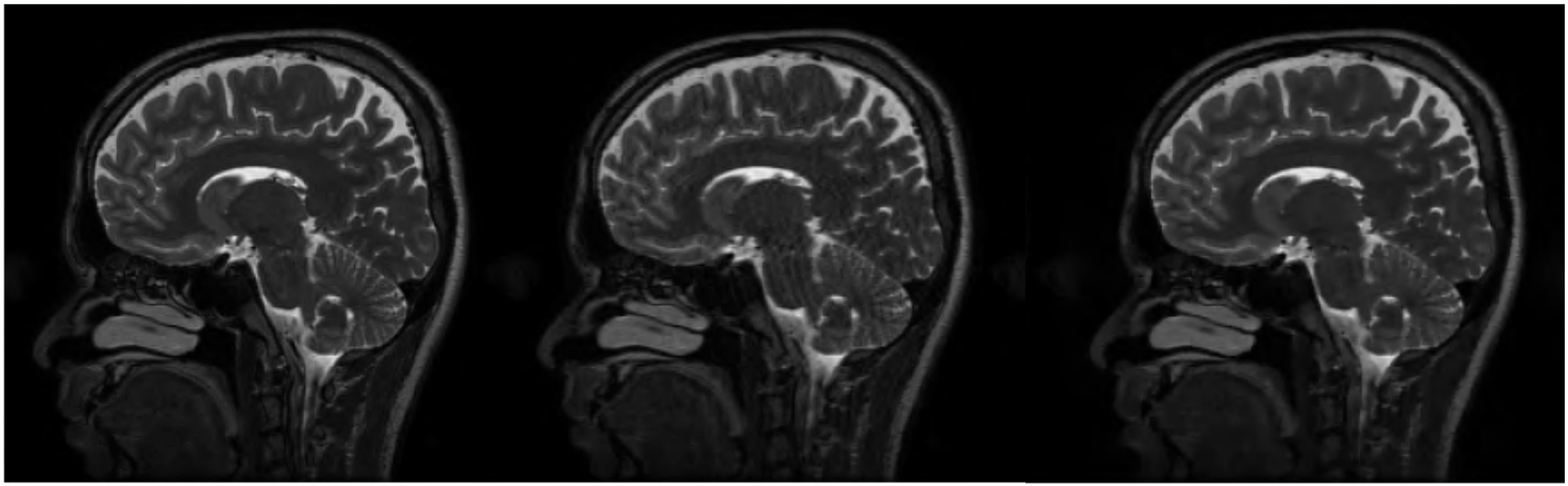


Some elements of architecture and their weights are fixed, determined by choice of optimization method, forward model H, and observed image data y.

Some weights to be learned from training data

Physical models, inverse problem methods and optimization theory lead to novel architectures and highly effective learning methods

Example: MRI reconstruction



Original Image

Machine learning method
higher accuracy
16 s to compute

Classical method
lower accuracy
350 s to compute

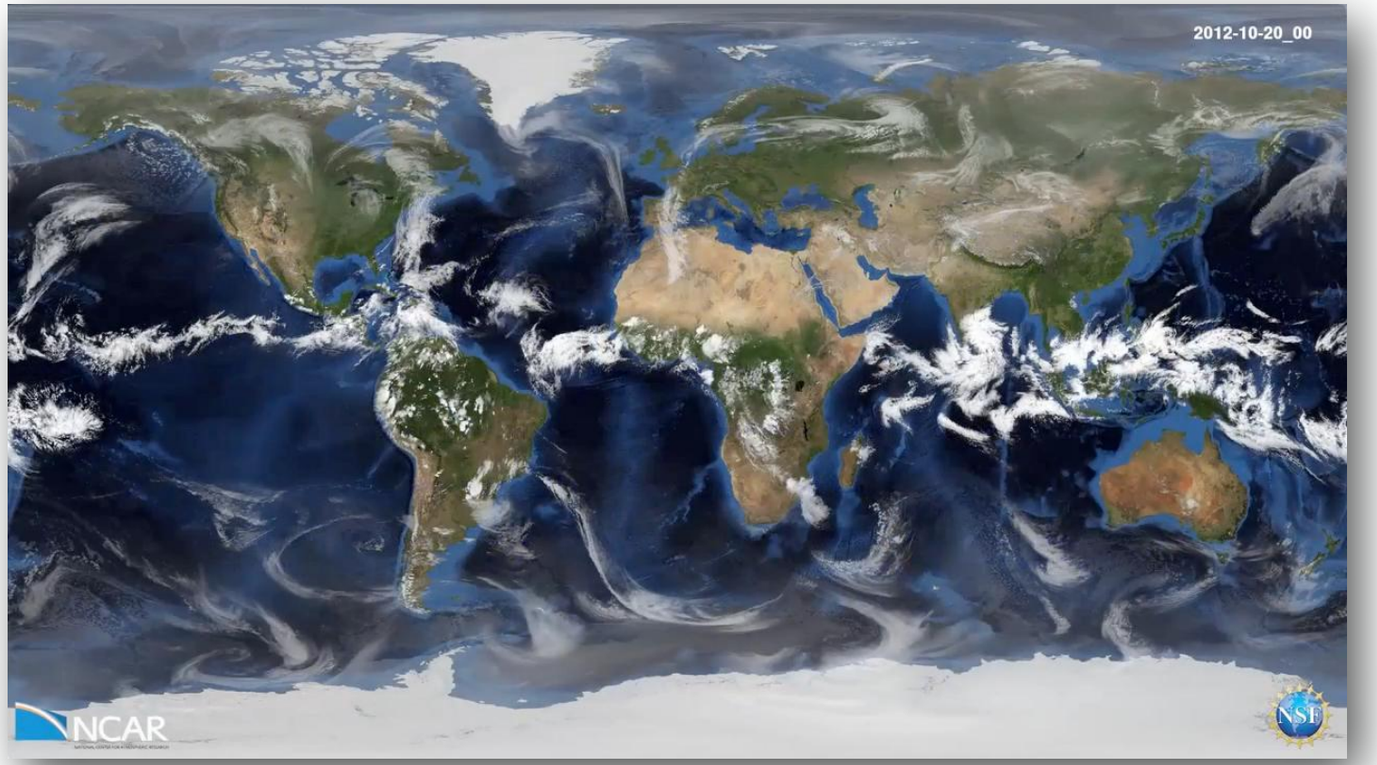
Data assimilation

European Centre for Medium-Range Weather Forecasts:

> "To make a forecast we need to know the current state of the atmosphere and the Earth's surface (land and oceans). The weather forecasts produced at ECMWF use data assimilation to estimate initial conditions for the forecast model from meteorological observations."

Here $\dim(x_t) = 10^8$ and $\dim(y_t) = 10^6$, and data is collected in 6-hour windows.

Additional applications in tracking, molecular chemistry, robotics, phylogenetics, economics, geosciences, and much more.



https://www.ecmwf.int/en/research/data-assimilation

State space modeling

Initialization

$$x_0 \sim p_0(x_0)$$

Transition

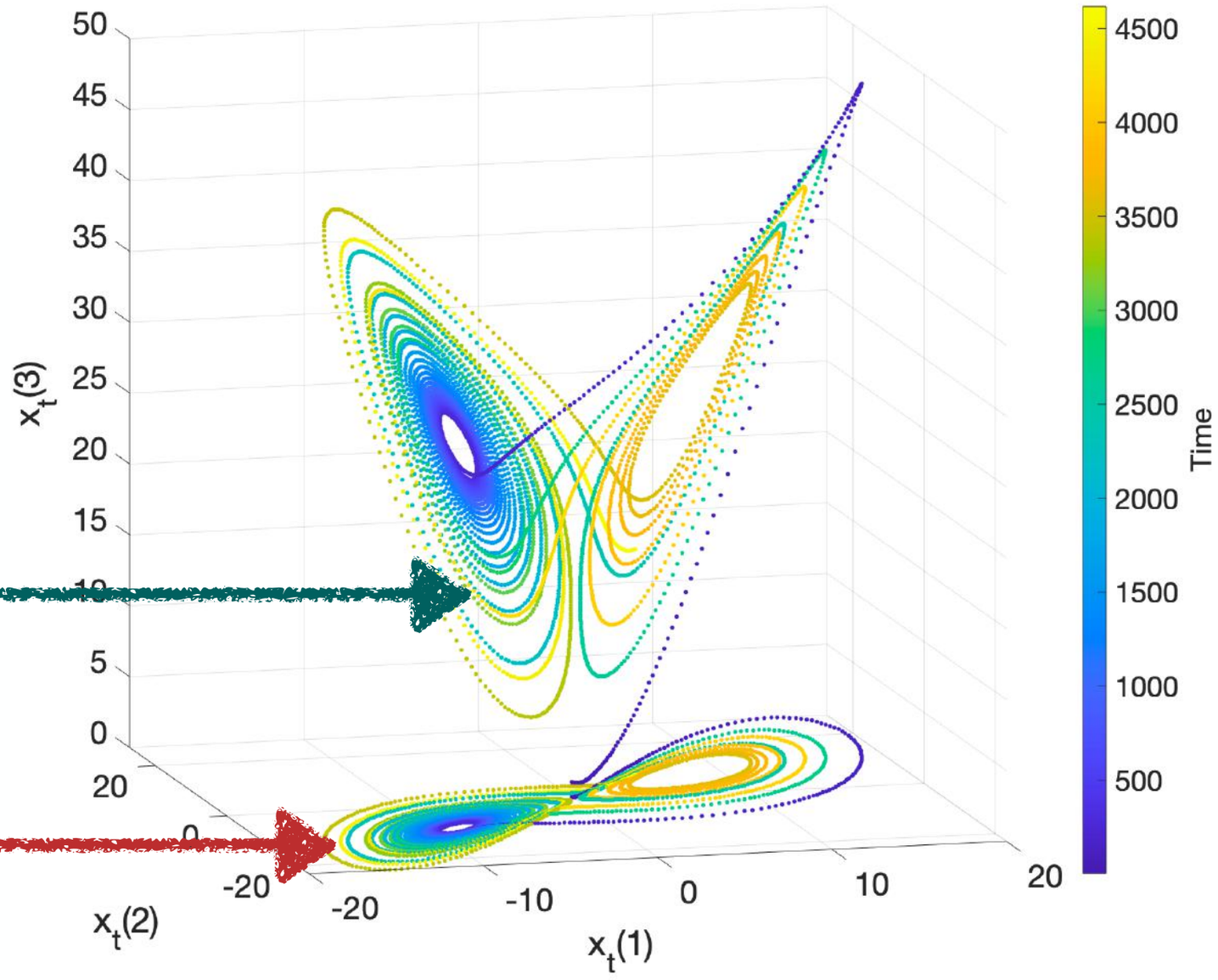
$$x_t = F(x_{t-1}) + Q^{1/2}\xi_t, \qquad \xi_t \sim \mathcal{N}(0,I)$$

Observation

$$y_t = Hx_t + R^{1/2}\eta_t, \qquad \eta_t \sim \mathcal{N}(0,I)$$

time-varying state x_t

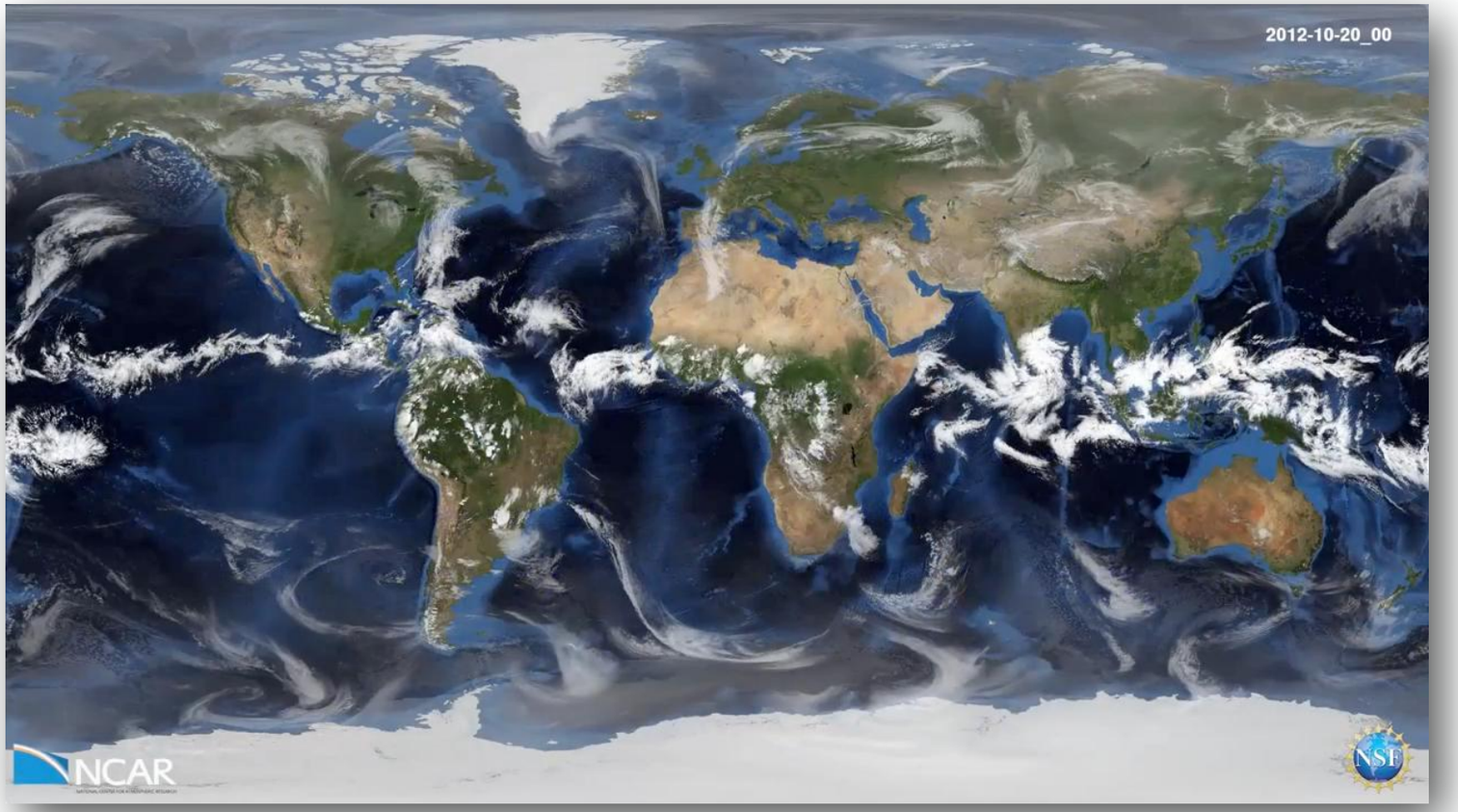
observed projections y_t



Machine learning for data assimilation

- Making accurate forecasts requires having a good model of underlying dynamics

- These models may have unknown parameters or only be approximate — and sometimes we have no model at all!



Data assimilation: estimating dynamics from indirect data

Initialization $x_0 \sim p_0(x_0)$

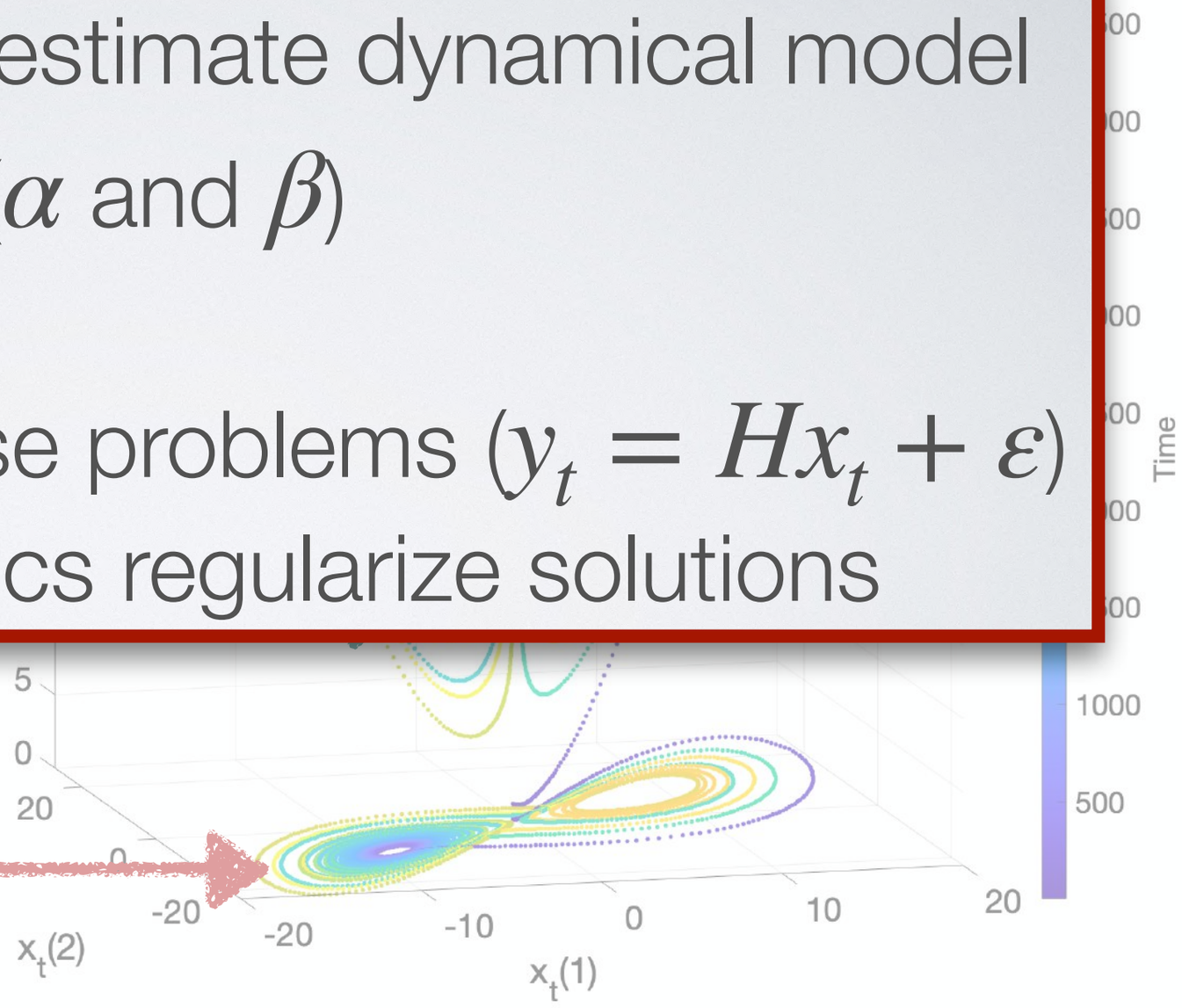
Transition $x_t = F_\alpha(x_{t-1}) + Q_\beta^{1/2}\xi_t, \qquad \xi_t \sim \mathcal{N}(0,I)$

Observation $y_t = Hx_t + R^{1/2}\eta_t, \qquad \eta_t \sim \mathcal{N}(0,I)$

Goal: Given $y_{1:T}$, estimate dynamical model (α and β)

Collection of inverse problems ($y_t = Hx_t + \varepsilon$) where dynamics regularize solutions

observed projections y_t



Our approach

- **General strategy:** choose $\theta := (\alpha, \beta)$ (e.g. neural network weights) that maximizes likelihood of observations

$$\hat{\theta} = \arg\max_{\theta} \mathscr{L}(\theta) \quad \text{where} \quad \mathscr{L}(\theta) := \log p(y_{1:T} | \theta)$$

- **Challenge:** log-likelihood generally does not have closed-form expression and must be numerically approximated

- **Insight 1:** Stochastic filtering tools like the Ensemble Kalman Filter (EnKF) yield approximate likelihood

- **Insight 2:** Using auto-differentiation to calculate likelihood gradients improves accuracy of learning optimal parameters

Our method: gradient ascent on approximate likelihood calculated using EnKF

Stochastic filtering & prediction

- **Goal**: estimate the current state of an evolving dynamical system observed via indirect measurements
- **Example**: tracking
  - State is position and velocity at time t
  - But we only observe noisy location at each time t
  - **Filtering**: At each time, given past observations, estimate true location and velocity
  - **Prediction**: At each time, given past observations, predict future location and velocity

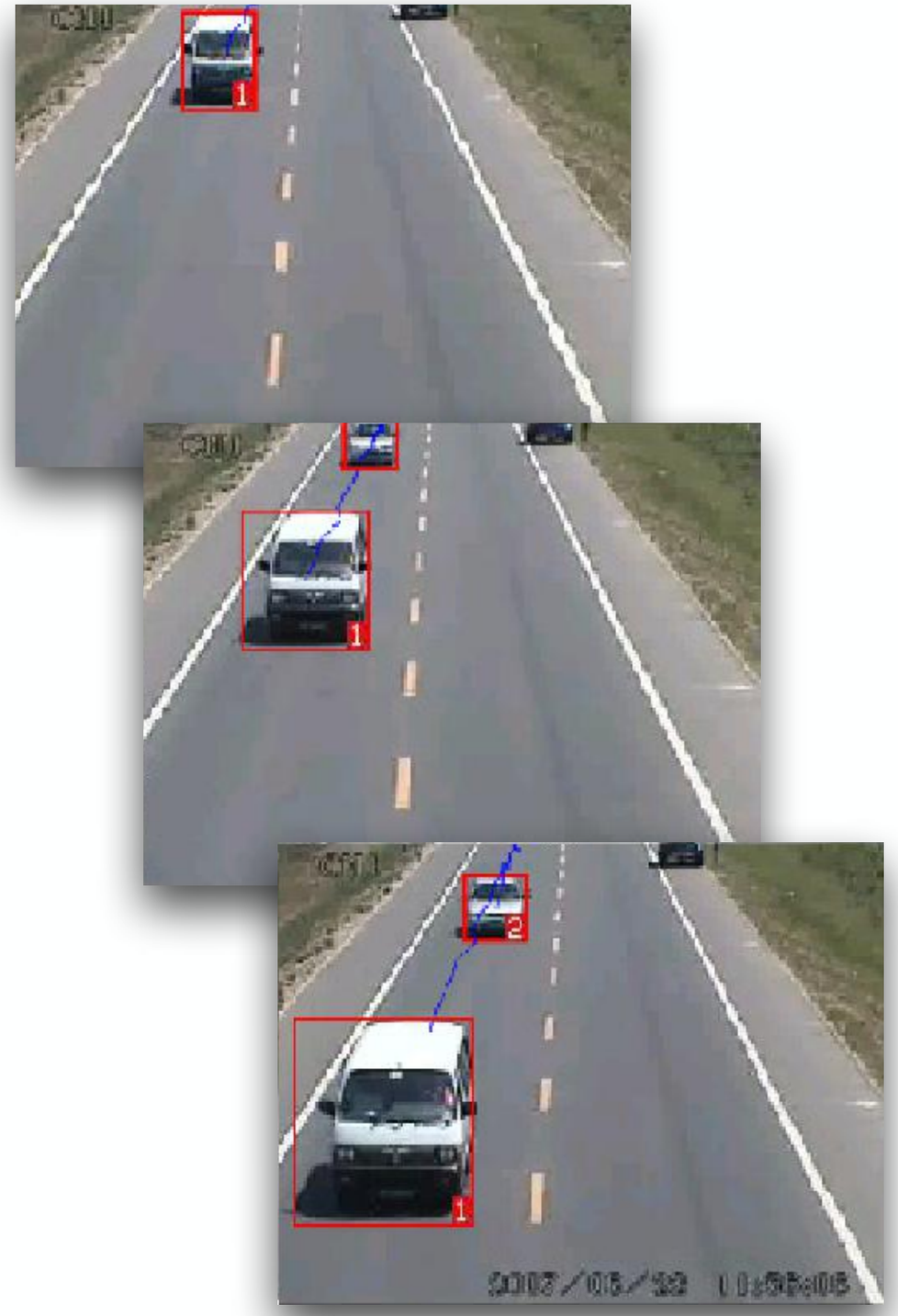
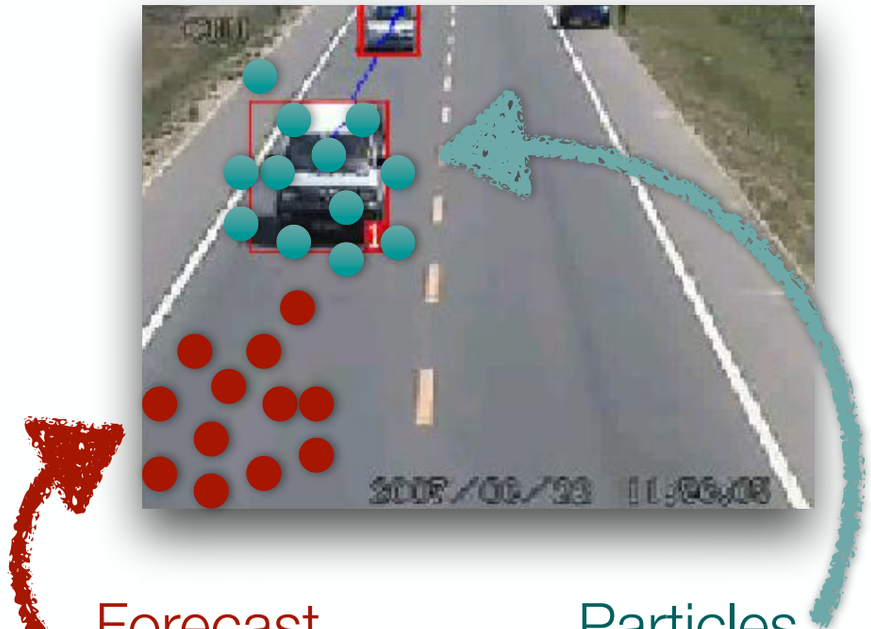


Image from Li, Wang, Wang, & Li 2010

Ensemble Kalman Filter — Forecast step

Let x_{t-1}^n be the value of the n^{th} particle at time $t-1$. **First, for each particle, we predict where it will be at the next time given dynamics** (F_α, Q_β):

$$\hat{x}_t^n = F_\alpha(x_{t-1}^n) + Q_\beta^{1/2}\xi_t^n, \qquad \xi_t^n \sim \mathcal{N}(0,I)$$



Forecast particles, $\{\hat{x}_t^n\}_{n=1}^N$

Particles from past, $\{x_{t-1}^n\}_{n=1}^N$

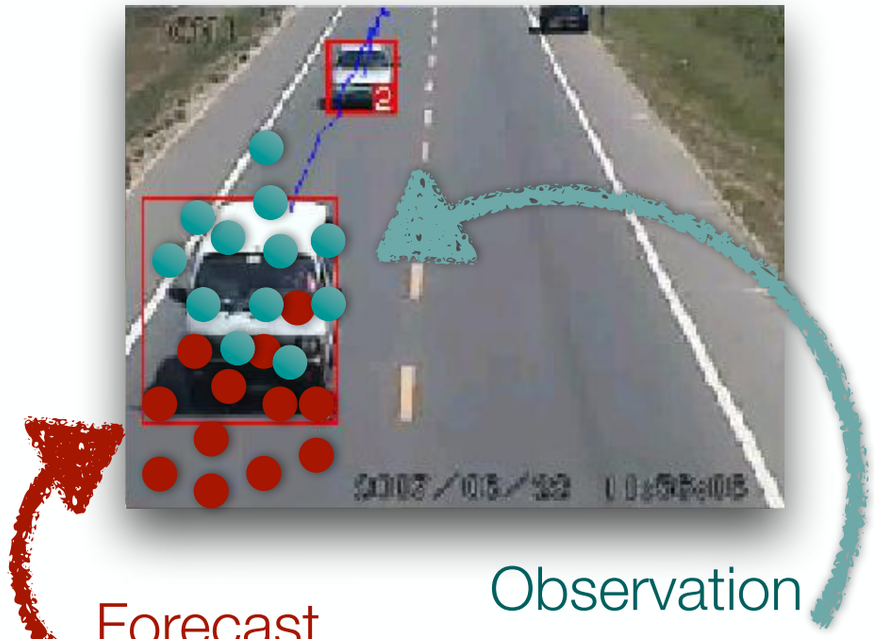
Evensen, 2009

Ensemble Kalman Filter — Analysis step

Next, we observe y_t and use this observation plus knowledge of H to improve our estimate of each x_t^n:

$$x_t^n = \hat{x}_t^n + \hat{K}_t(y_t + R^{1/2}\gamma_t^n - H\hat{x}_t^n),$$

$$\gamma_t^n \sim \mathcal{N}(0, I)$$



where

$$\hat{K}_t = \hat{C}_t H^\top (H\hat{C}_t H^\top + R)^{-1}$$

is called the Kalman gain

Evensen, 2009

The **Ensemble Kalman Filter** sequentially estimates filtering distributions

$$p_\theta(x_{1:t} \mid y_{1:t}),\ 1 \leq t \leq T.$$

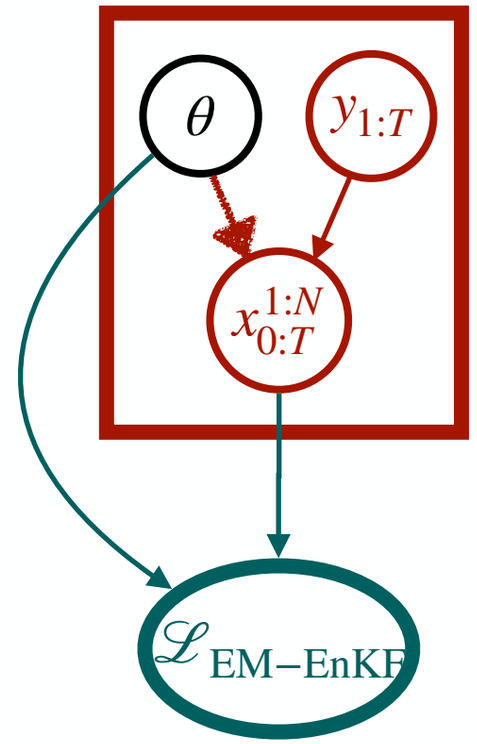
The distributions are represented using a collection of N particles.

Using ensemble mean and covariance, we can approximate the likelihood $\mathscr{L}(\theta)$

Expectation Maximization

Brajard, Carassi, Bocquet, & Bertino, 2020

- for $k = 1,2,\ldots$
  - $x_{0:T}^{1:N} = \mathrm{EnKF}(\theta^k, y_{1:T})$
  - $\theta^{k+1} = \theta^k + \eta \nabla_\theta \left[\sum_{n,t} \log \mathcal{N}\left(y_t^n; F_{\theta^k}(x_{t-1}^n), Q_{\theta^k}\right) \right]$



Key insight

In contrast to the EM method, we

- treat the particles as *functions of* θ and
- compute likelihood gradients that reflect this dependence
- by leveraging **automatic differentiation**

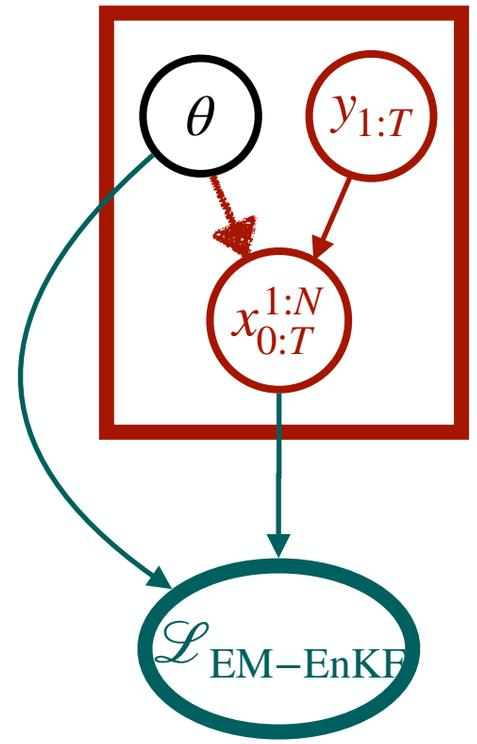
Automatic differentiation is different from numerical differentiation

- autodiff uses compositions of elementary functions whose derivatives are known
- autodiff to compute gradients incurs negligible extra computational cost compared to evaluation of likelihoods
- finite difference approximations cause discretization errors

Expectation Maximization

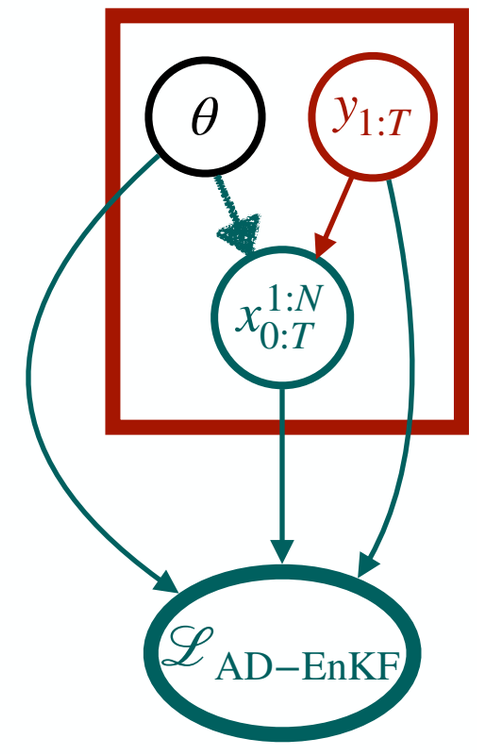
Brajard, Carassi, Bocquet, & Bertino, 2020

- for $k = 1,2,\ldots$
  - $x_{0:T}^{1:N} = \mathrm{EnKF}(\theta^k, y_{1:T})$
  - $\theta^{k+1} = \theta^k + \eta \nabla_\theta \left[\sum_{n,t} \log \mathcal{N}\left(y_t^n; F_{\theta^k}(x_{t-1}^n), Q_{\theta^k}\right) \right]$



Our Autodiff-EnFK approach

- for $k = 1,2,\ldots$
  - $x_{0:T}^{1:N}(\theta^k) = \mathrm{EnKF}(\theta^k, y_{1:T})$
  - $\theta^{k+1} = \theta^k + \eta \nabla_\theta \left[\sum_{n,t} \log \mathcal{N}\left(y_t^n; F_{\theta^k}(x_{t-1}^n(\theta^k)), Q_{\theta^k}\right) \right]$



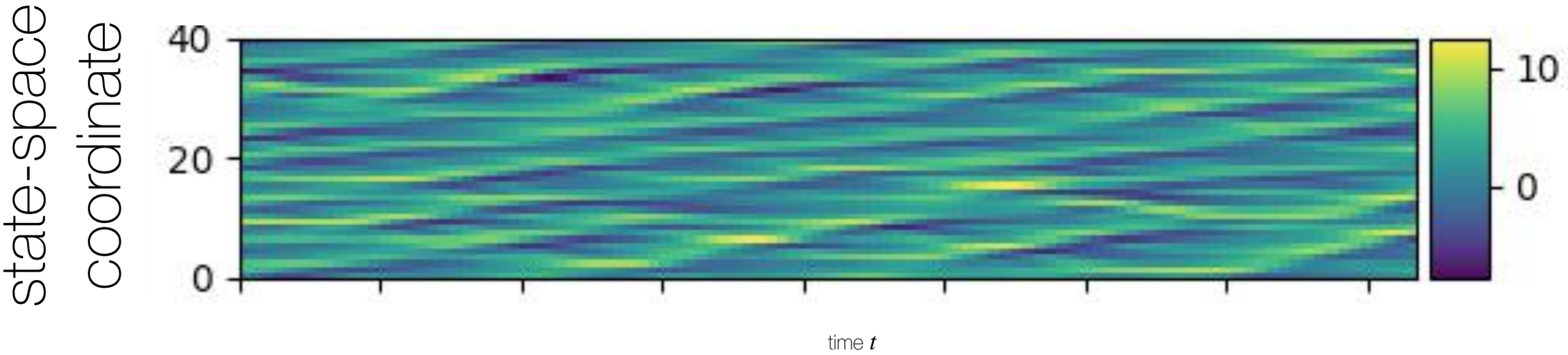
Compute accurate gradients using autodifferentiation

Lorenz-96 system

Let F^* be flow map of vector field:

$$\frac{dx}{ds} = f^*(x), \qquad F^* : x(s) \mapsto x(s + \Delta_s)$$

with $f^{*(i)}(x) = -x^{(i-1)}(x^{(i-2)} - x^{(i+1)}) - x^{(i)} + 8,\ i = 1,\ldots,40$

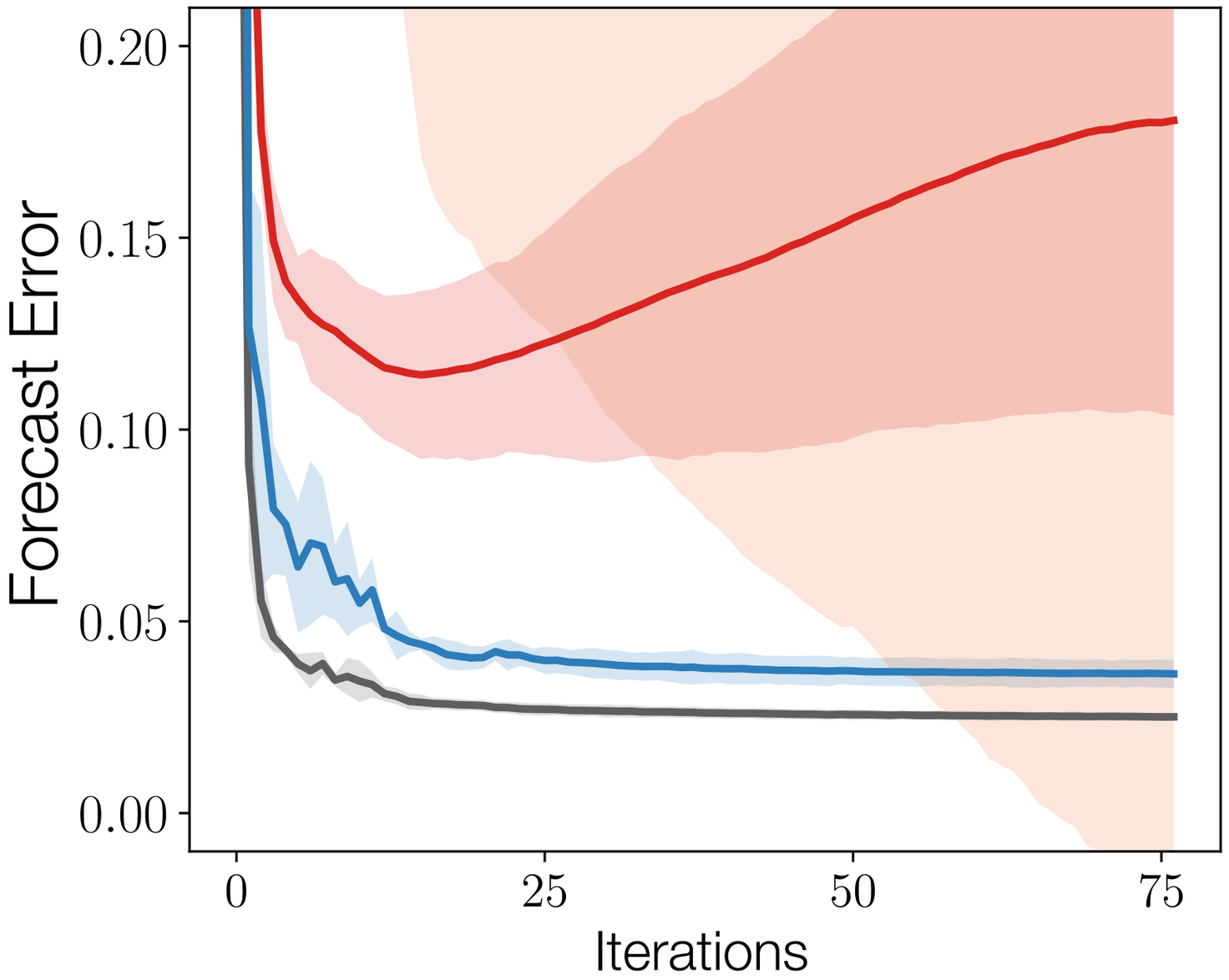


- Common test model for filtering algorithms and low-frequency climate models
- Prototypical turbulent dynamical system; our setting: strong chaotic turbulence
- Dynamics exhibit strong energy-conserving non-linearities
- Can be defined for any desired state space dimension d_x

Majda & Harlim, 2012

Example on turbulent dynamics with partial obs.

We only observe 66% of the state and use a neural network to estimate the underlying dynamics



Our method with ZERO knowledge of dynamics

Our method correcting inaccurate model of dynamics

Competing method with ZERO knowledge of dynamics

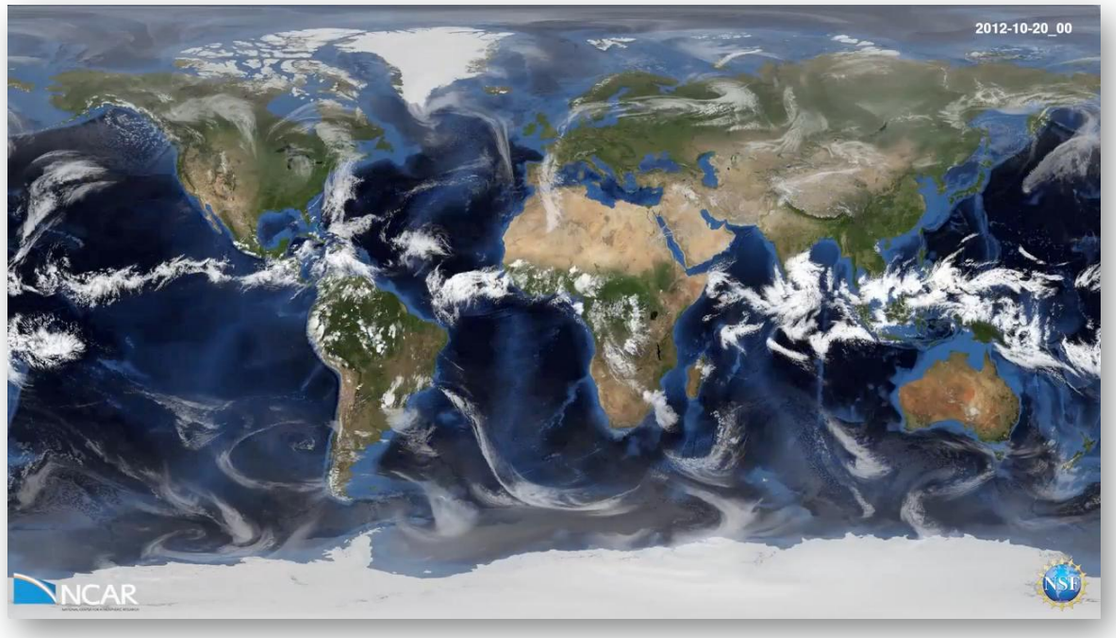
Competing method correcting inaccurate model of dynamics

Our method is more accurate and more stable

Example — estimating simulation parameters

- Climate simulator takes parameters x and outputs simulation $y = H(x)$
- Given observations y_{obs}, what are the corresponding parameters x?

- Similar to previous inverse problem settings, but now
  - H is nonlinear
  - we don't have an explicit form for H, can only access through simulations
  - we want to quantify uncertainty about x



Past approaches

Classical method: For some predefined moment function m,

$$\hat{x} = \arg\min_{x} \|m(y) - m(H(x))\|^2_{\Sigma[m(y)]}.$$

- need expert knowledge to choose m,
- requires repeated (slow) runs of H for each new observation y,
- gradient-free optimization methods like ensemble Kalman inversion highly dependent on prior p_x.

Supervised regression: Learn a neural network f_θ so that

$$\hat{x} = f_\theta(y);$$

difficult to get accurate uncertainty estimates.

Schneider, Tapio, Lan, Shiwei, Stuart, Andrew, et al., 2017

Standard Emulator Approach

low-dimensional parameter x

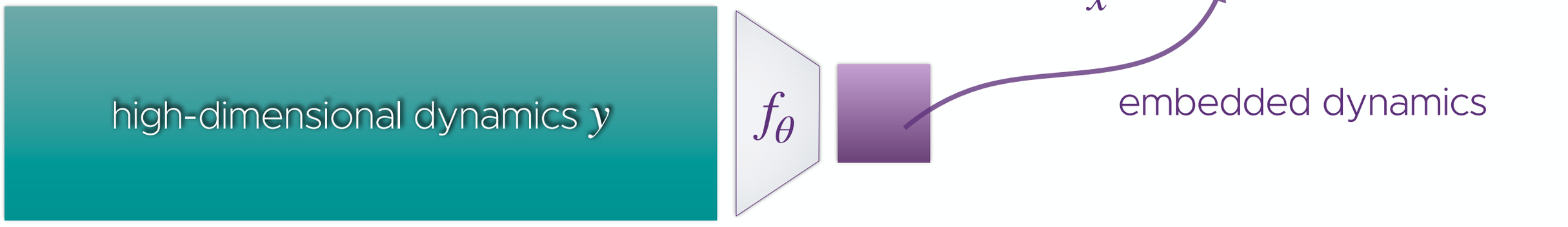
$$\hat{x} = \arg\min_x \|m(y) - m(\hat{H}_\theta(x))\|_{\Sigma[m(y)]}$$

Embed & Emulate (ours)

low-dimensional parameter x

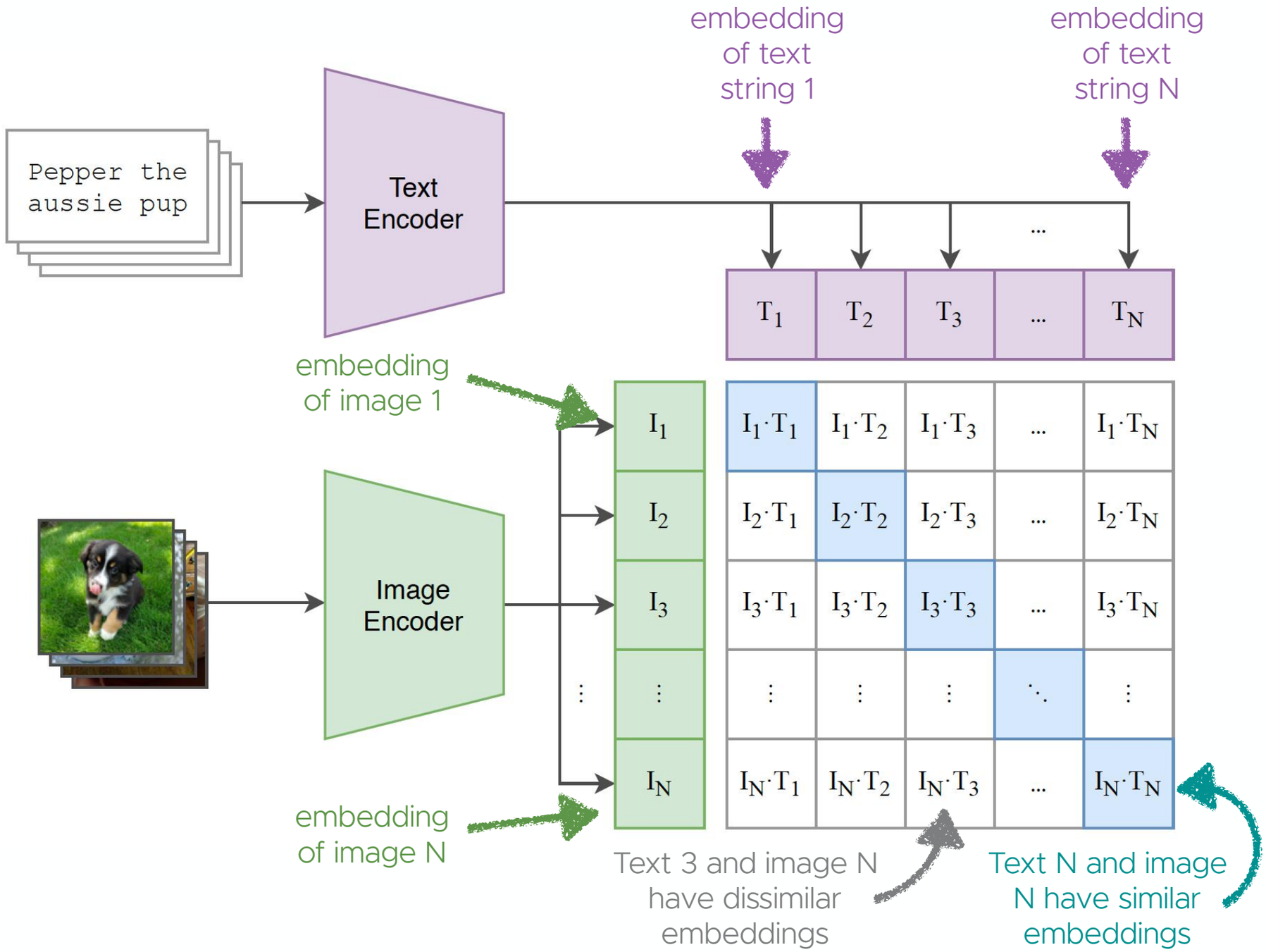
embedded parameters and dynamics $\approx f_\theta \circ H$

$$\hat{x} = \arg\min_x \|f_\theta(y) - \hat{g}_\theta(x)\|_2$$



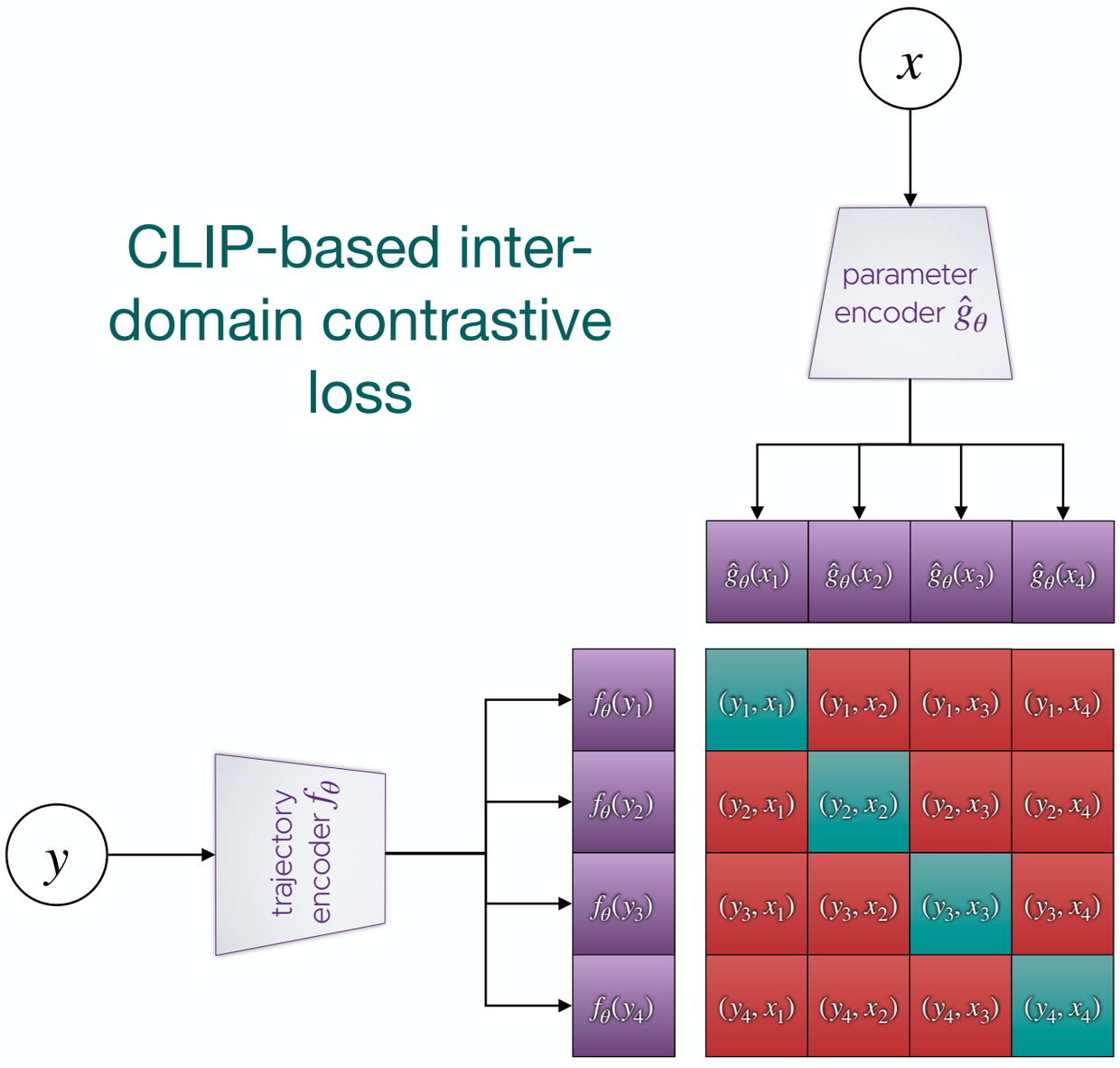
Leveraging ideas from computer vision

- CLIP = Contrastive Language-Image Pre-Training
- Learns embedding so that
  - if an image and text go together, their embeddings are similar
  - if an image and text are unrelated, their embeddings are far apart



Embed & Emulate key ideas

CLIP-based inter-domain contrastive loss

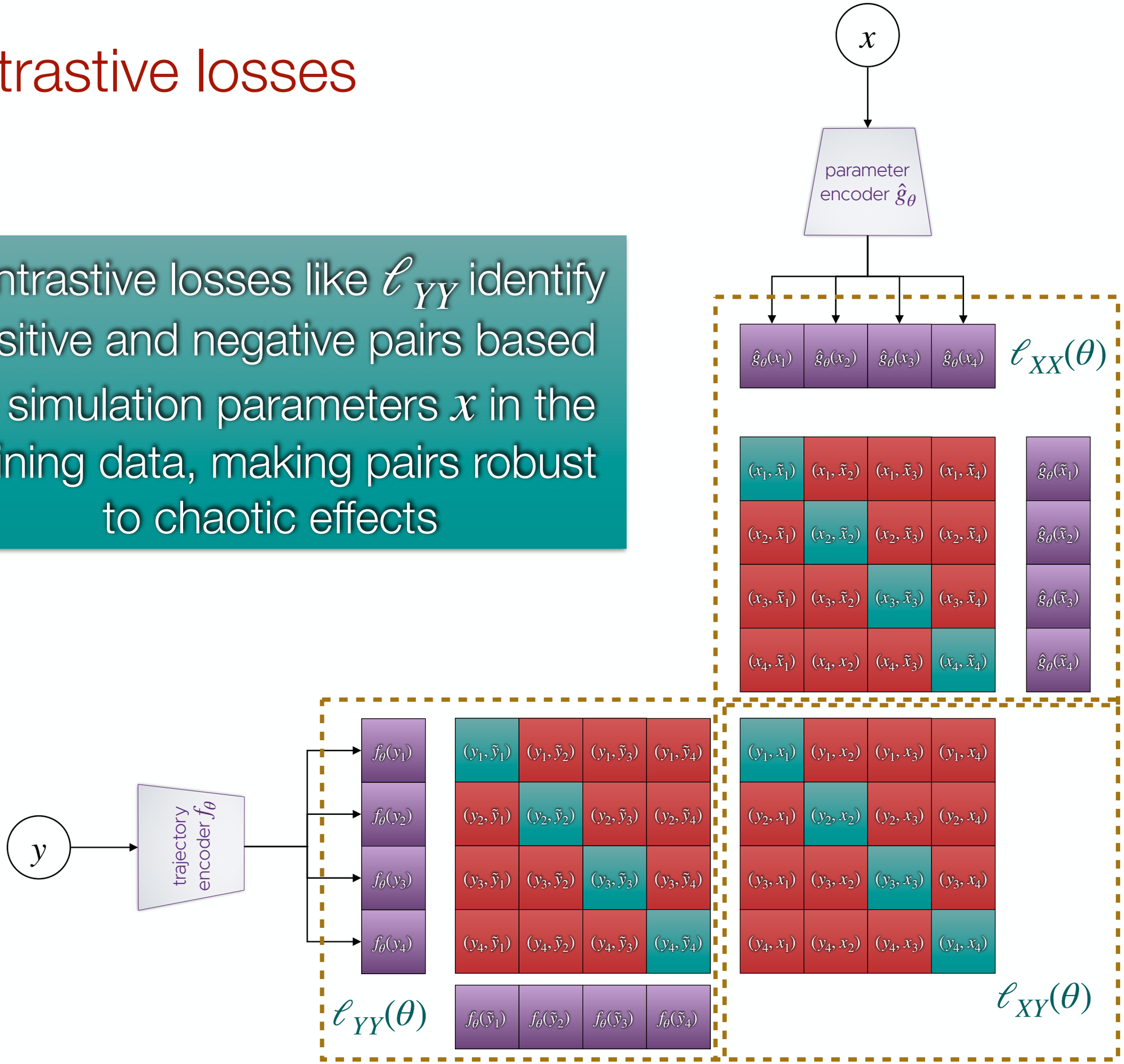


Inter-domain contrastive learning scheme: Diagonals are dot products between representations of "positive" pairs (x_i, y_i) .

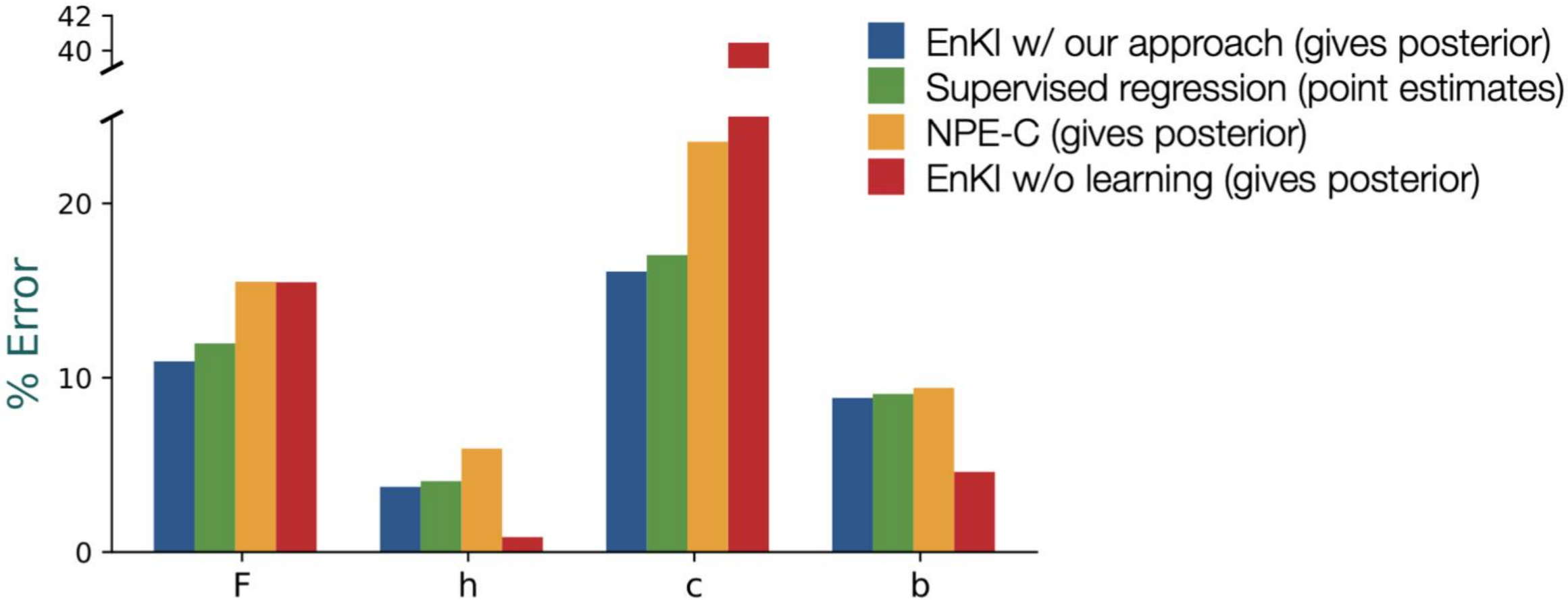
- We design an "emulator" that fits well in the context of parameter estimation problem.
- We use CLIP-wise loss to align the metric space of the "emulator" and the embedding network.
- We use contrastive loss to capture intra-domain structural information to learn meaningful embeddings.

Contrastive losses

Contrastive losses like ℓ_{YY} identify positive and negative pairs based on simulation parameters x in the training data, making pairs robust to chaotic effects



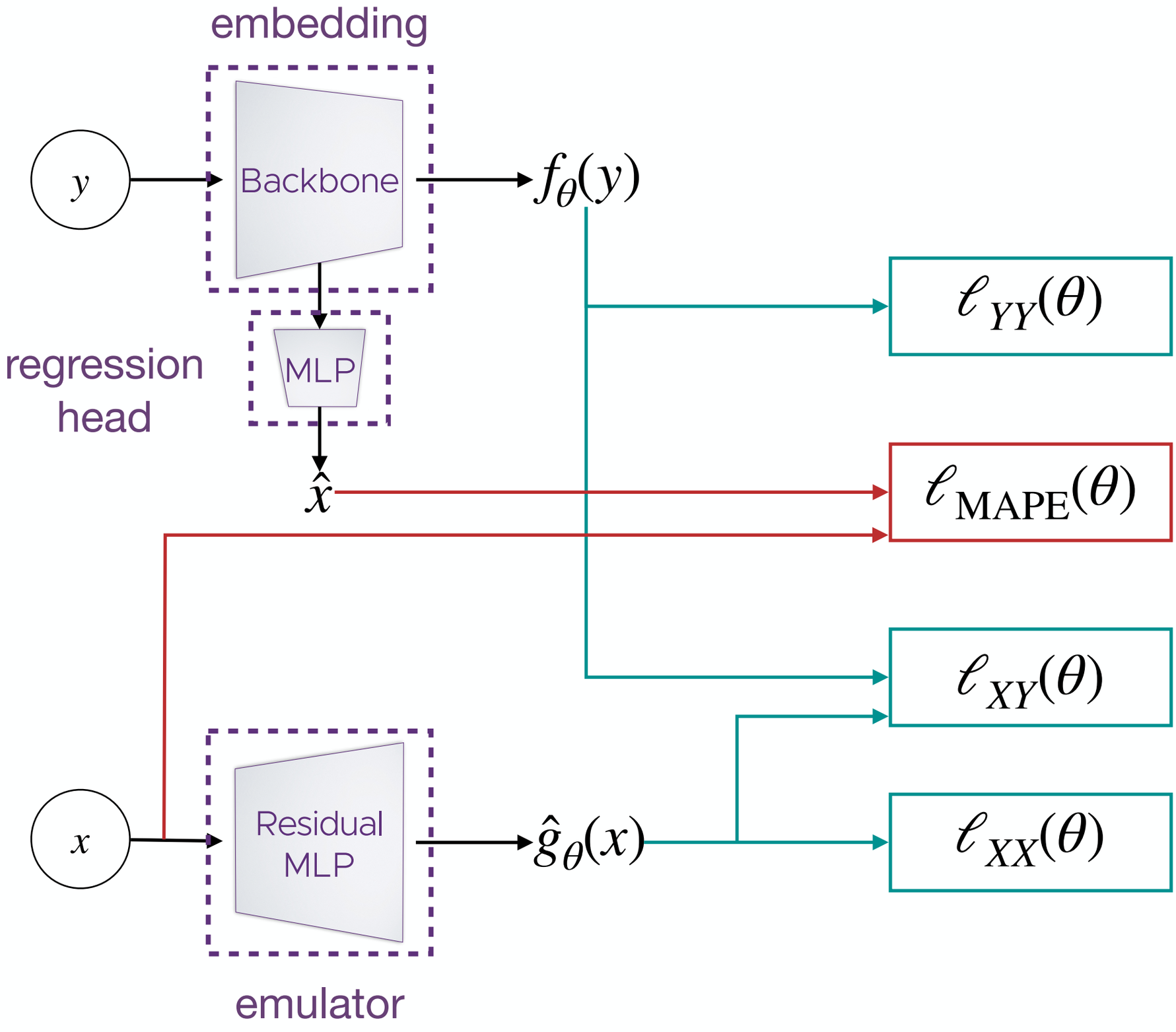
Example estimating Lorenz-96 parameters



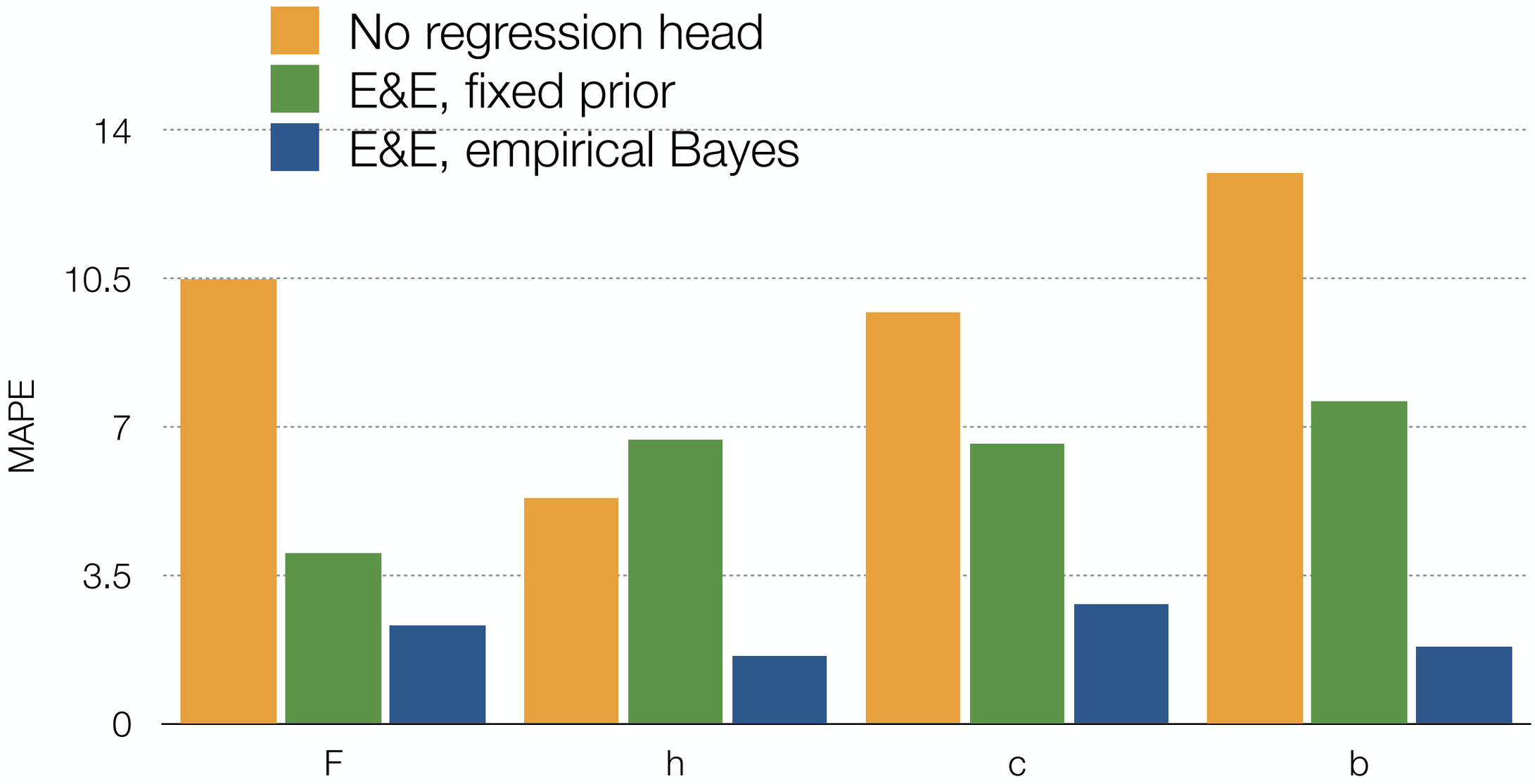
Method	EnKI w/ our approach	Supervised Regression	NPE-C	EnKI w/o learning
Time in minutes	52.0 (0.87 h)	43.0 (0.72 h)	52.0 (0.87 h)	8,000 (5.5 d)

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

Embed & Emulate architecture



Regression head informs EnKI prior



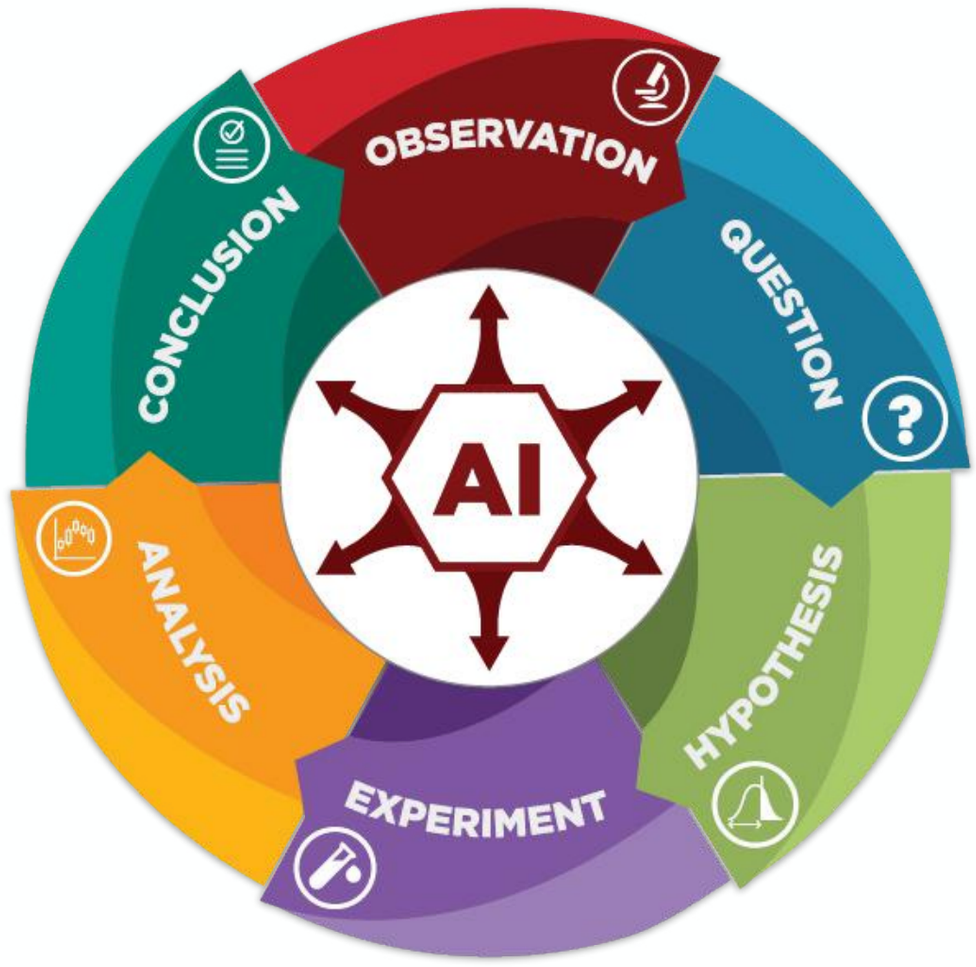
Regression head helps guide contrastive learning

Using regression head to set prior used by EnKI further reduces errors

Physics-based models can inform neural network architectures and training, improving the reliability and interpretability of ML systems.

AI & Science @ UChicago

We envision **AI as an integral component of the scientific method**, guiding the construction of hypotheses, designing sequences of experiments, and analyzing data to develop new hypotheses, while advancing core AI principles.



THE UNIVERSITY OF CHICAGO DATA SCIENCE INSTITUTE

Argonne NATIONAL LABORATORY

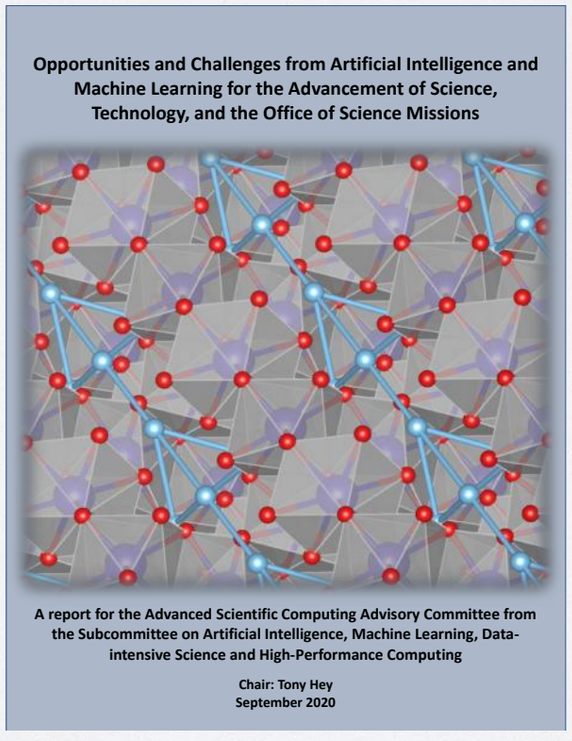
Fermilab

THE UNIVERSITY OF CHICAGO MARINE BIOLOGICAL LABORATORY

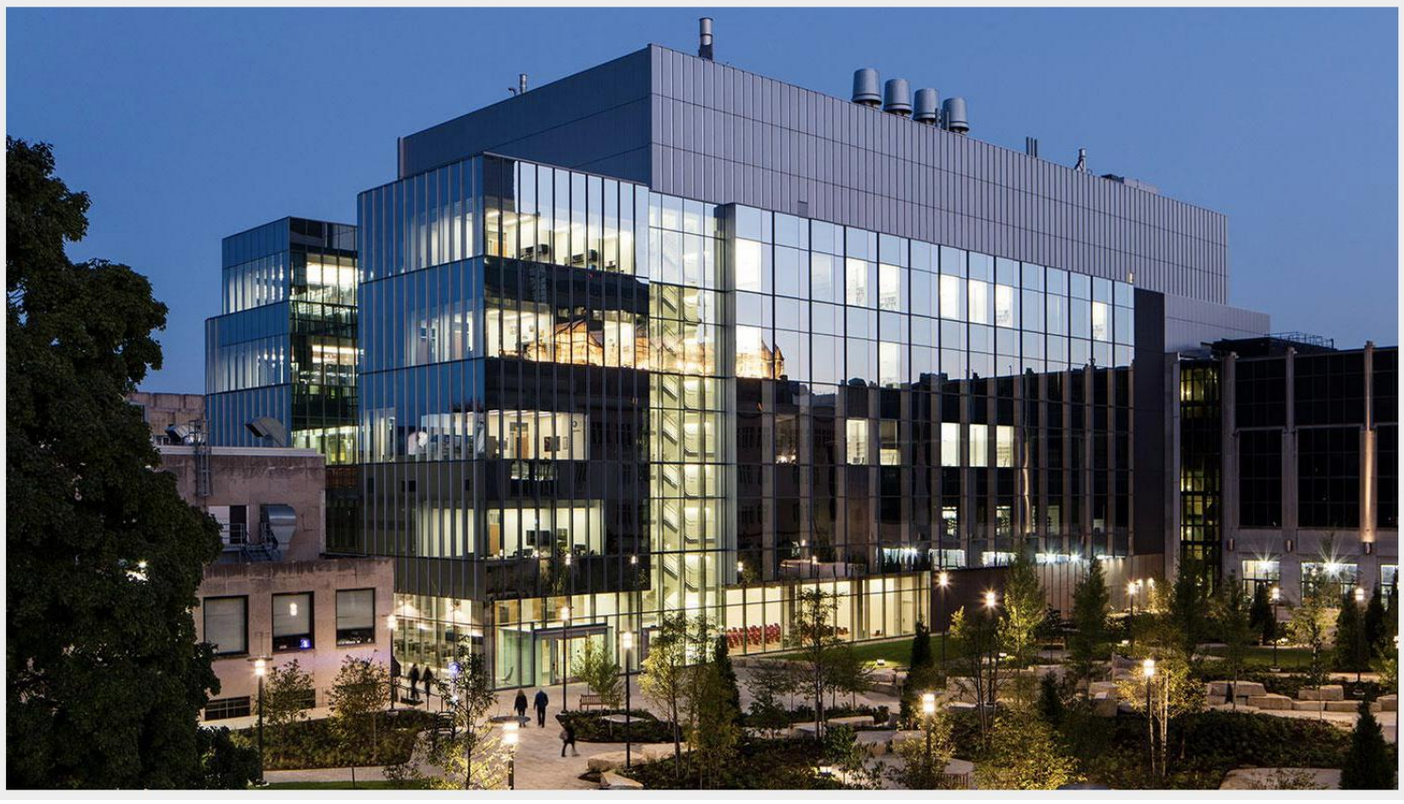
Key challenges of AI in science

- Effective ML training for **small or sparse datasets**
- Incorporation of **physical models** into AI structure
- Creation of **surrogates**
- Dimensional reduction, data synthesis and compression, and **reduced order models**
- Control and AI-enabled **experimental design**
- **Operation** of experimental facilities
- Robustness, **inference**, and calibration
- Real-time **decision-making**
- Predictive maintenance and **event prediction**
- Training data variability and **noise**
- **Interpretable** models and algorithms
- Coupling **simulations and experiment**

2020 DOE Report "Opportunities and Challenges from Artificial Intelligence and Machine Learning for the Advancement of Science, Technology, and the Office of Science Missions"



New Schmidt Futures fellowship at UChicago to foster next generation of AI-driven scientists



SCHMIDT **FUTURES**

https://aiscience.uchicago.edu/

Thank you!

Roxie Jiang, UChicago

Davis Gilton, Microsoft

Greg Ongie, Marquette

Yuming Chen, UChicago

Daniel Sanz-Alonso, UChicago