SAND2022-14111C

Improving Digital Twins by Learning from a Fleet of Assets

J.D. Jakeman, D.T. Seidl, A.A. Gorodetsky

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY | Office of Science

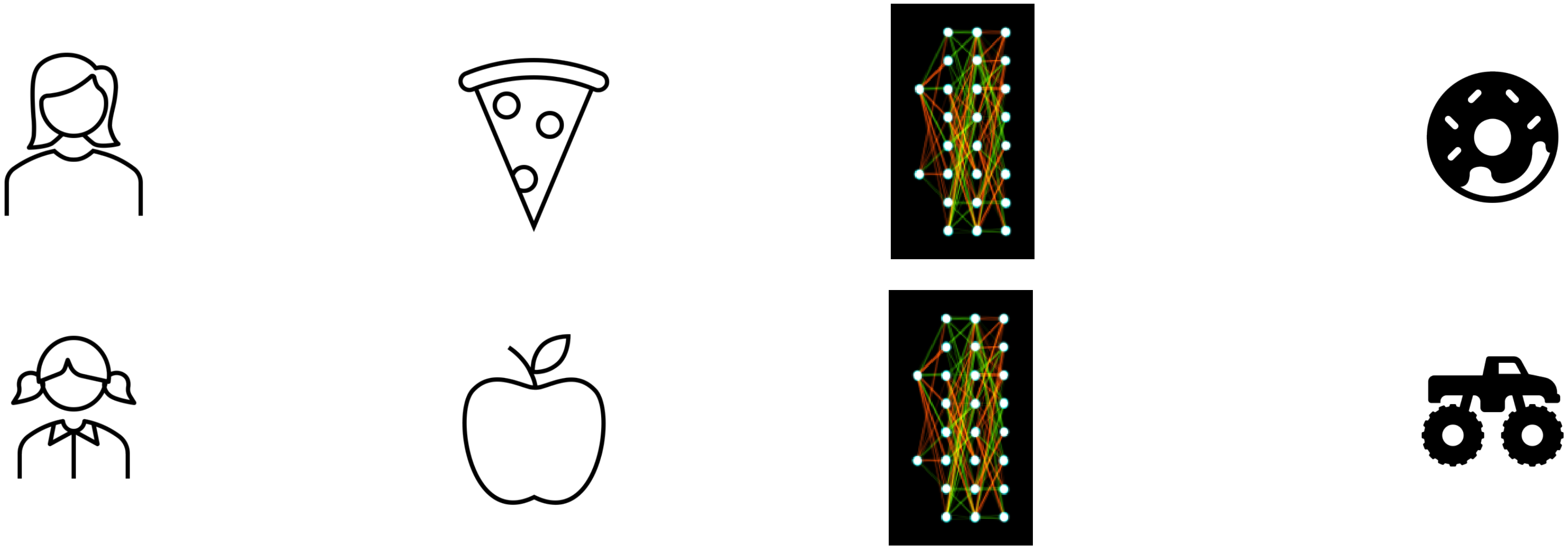
Advanced Scientific

Robust Interpretable Scalable Efficient RISE

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

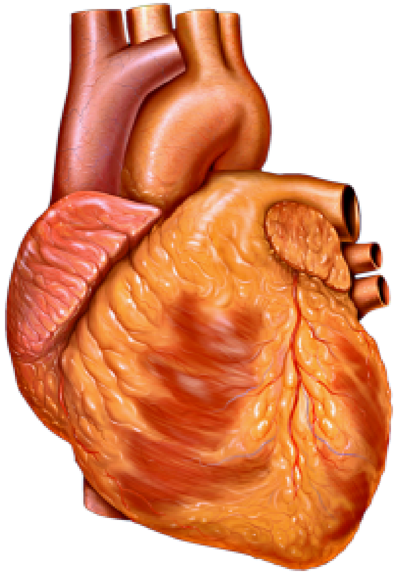
A Digital Twin

A digital twin is an evolving virtual model of a specific system or physical asset that assimilates data over its lifecycle to becomes a “patient-specific” model that can be used for intelligent automation and decision making.



Person | Data | Digital Twin | Decision making

Scientific Digital Twins



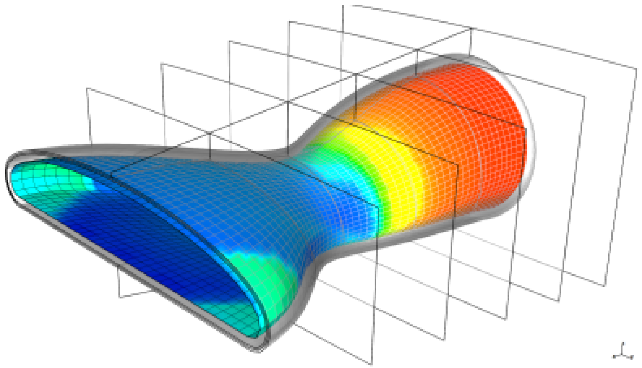
Formulating a Digital Twin

Assimilation

Want to calibrate the digital twin M to the physical asset

$$y = h(M(s, x, t;\ \theta^{\star})) + \epsilon$$

Asset



Digital Twin

$$M(s, t, x; \theta_k)$$

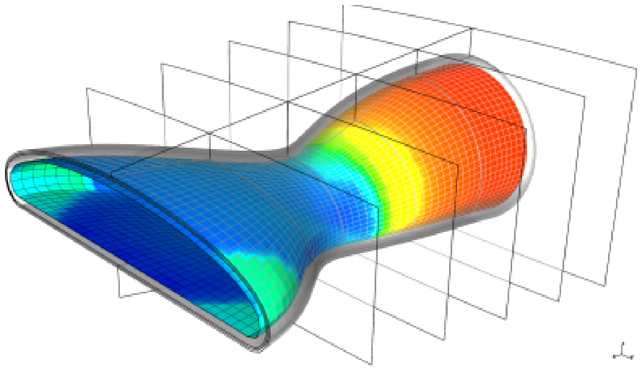
Formulating a Digital Twin

Assimilation

Want to calibrate the digital twin M to the physical asset

$$y = h(M(s, x, t;\ \theta^{\star})) + \epsilon$$

Asset

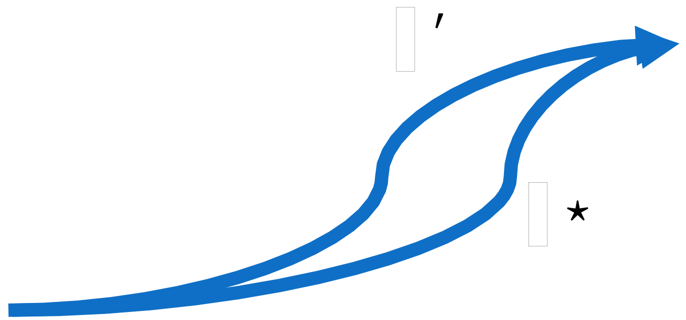


Digital Twin

$$M(s, t, x; \theta_k)$$

Prediction

Want to use the digital twin to determine how to fly asset safely given current health



Formulating a Digital Twin

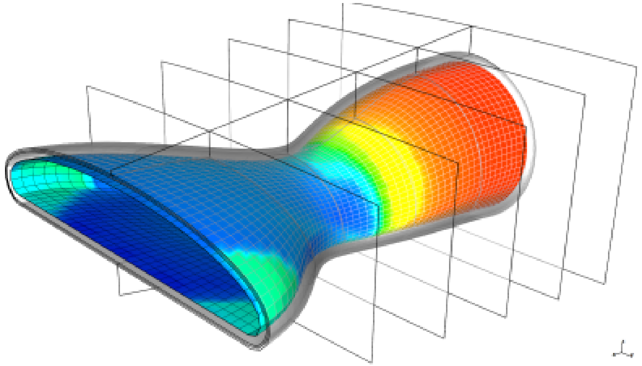
Want to predict performance under different flight scenarios $\boldsymbol{x}$

Collect data to infer health $\boldsymbol{\theta}^{\star}$

$$y = h(M(s, x, t;\ \theta^{\star})) + \epsilon$$

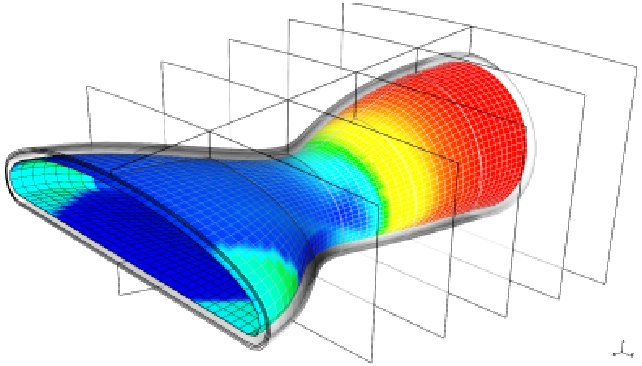
$$y = h(M(s, x, t;\ \theta^{\star})) + \epsilon$$

Asset



Event changes health of asset

$\theta_k \rightarrow \theta_{k+1}$



Digital Twin

$M(s, t, x; \theta_k)$

$M(s, t, x; \theta_{k+1})$

Time

Formulating a Digital Twin

The following process is often used to update and predict with a digital twin developed using first principles, e.g. PDE model.

Collect observations

$$y = h(s; x, \theta^{\star}) + \epsilon$$

Temperature of nozzle

Infer posterior of model variables

$$p(\theta \mid y) \propto p(y \mid \theta)p(\theta)$$

Estimate thermal conductivity due to changes in deteriorating insulation

Propagate posterior through predictive model

(,) = ((, , ,))

Predict maximum stress for specific future flight scenarios $\boldsymbol{x}$

A Data Driven Digital Twin

The following process is often used to update and predict with a purely data-driven digital twin

Want to construct approximation

$$f(x;\theta) \approx \hat{g}_N(x;\theta_n) = \sum_{n=0}^{N} \theta_n \phi_n(x)$$

Stress f vs flight scenarios $\boldsymbol{x}$

Collect observations

$$w = f(x;\theta^\star) + \epsilon$$

Observations are of stress directly

Learn approximation unknowns $\boldsymbol{\theta}$

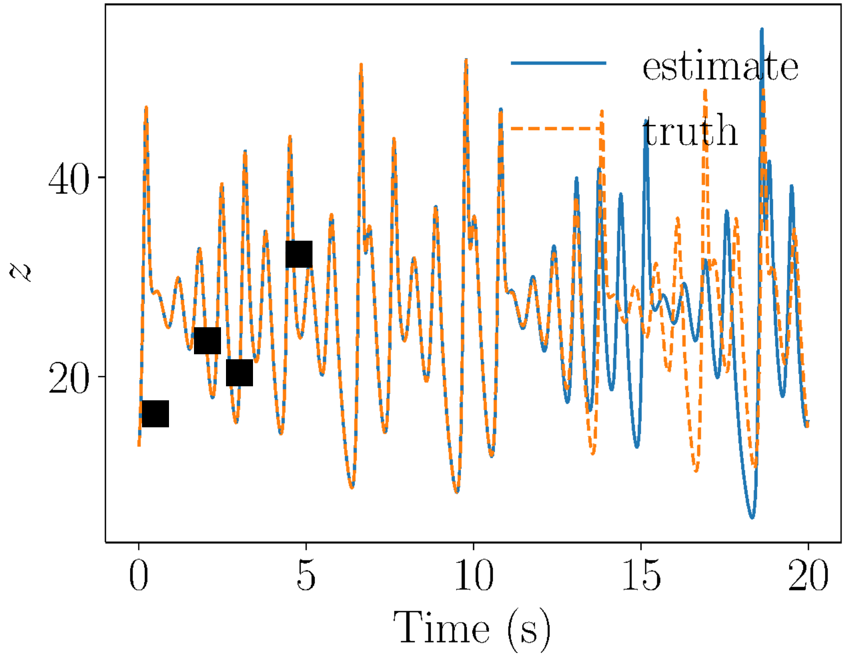
$$\text{argmin}_\theta \left|\left|w - \hat{g}_N(x,\theta)\right|\right|_{\Sigma_\epsilon}$$

E.g. Use MLE

CHALLENGE

How can we make informative inferences to enable accurate prediction when data is limited?

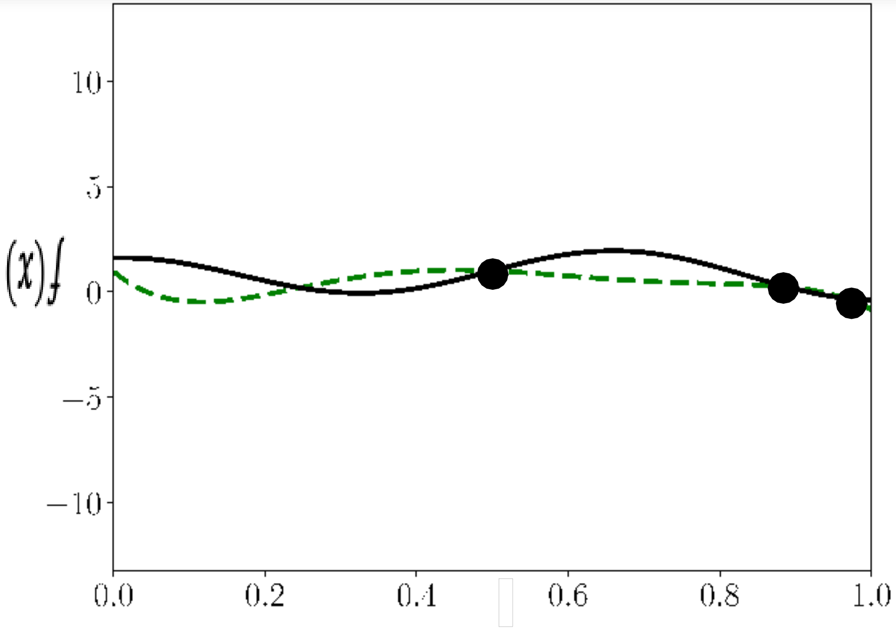
A first principles digital twin



$$\dot{x}_1 = \sigma(x_2 - x_1),$$
$$\dot{x}_2 = x_1(\rho - x_3) - x_2,$$
$$\dot{x}_3 = x_1 x_2 - \beta x_3$$

θ = [σρβ]

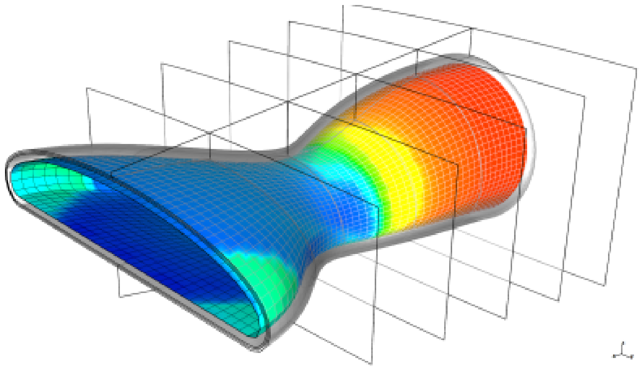
A purely data-driven digital twin



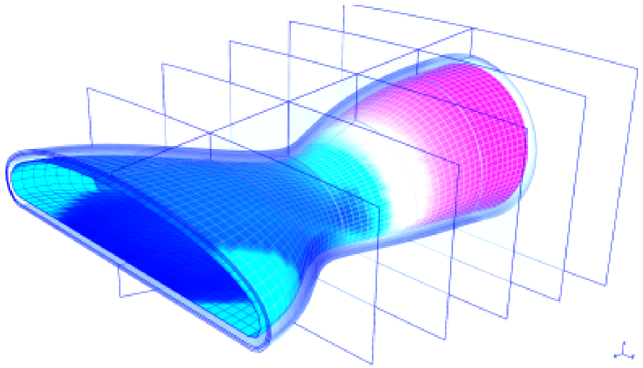
$$\hat{g}_N(x;\theta_n) = \sum_{n=0}^{N} \theta_n \phi_n(x) \quad \sim (0,1)$$

Digital Twins of Asset Classes

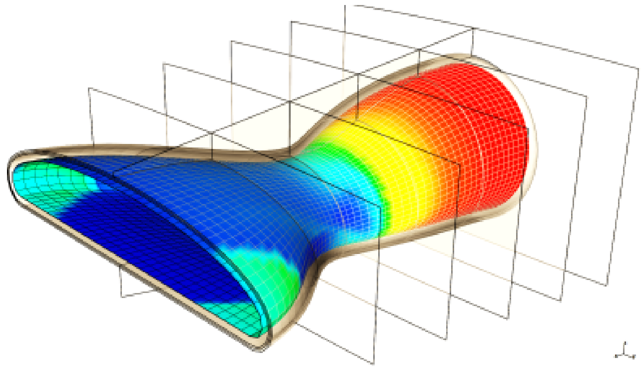
Often an asset is one of many within a class of assets. The exact health of these assets will depend on manufacturing differences (e.g. additive manufacturing) and/or the operating conditions of each individual asset.



Asset 0



Asset 1

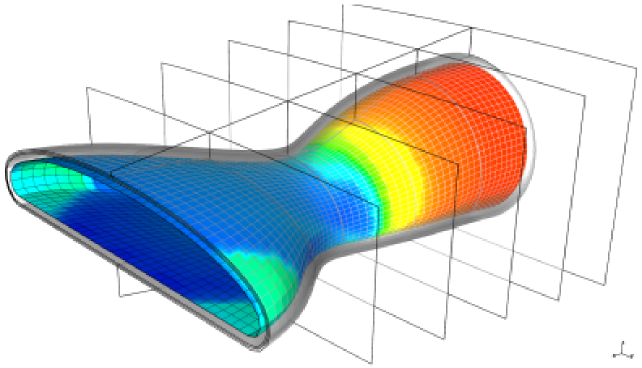


Asset 2

However, many assets are similar and we will exploit relationships in the observational data to improve the predictive capability of digital twins.

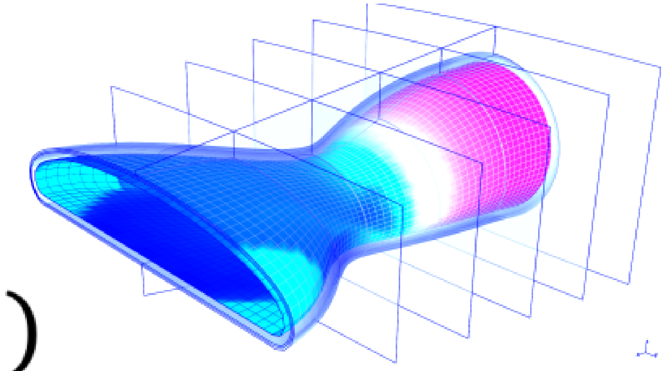
Connecting Twins Via Their Outputs

Collect data for each asset



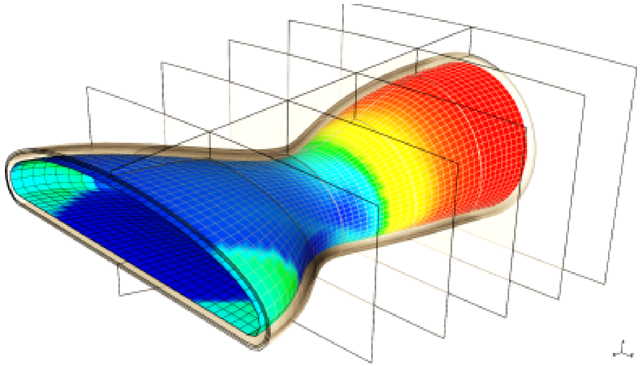
(X_0, y_0)

Asset 0



(X_1, y_1)

Asset 1



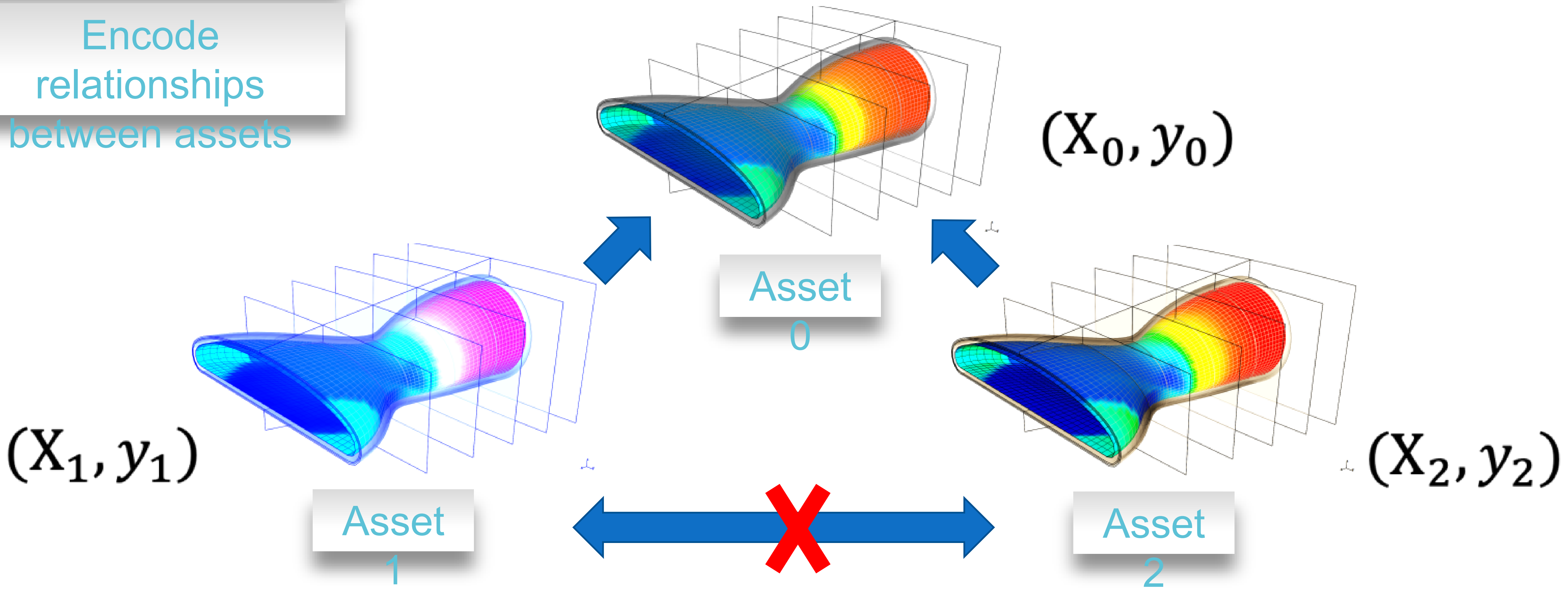
(X_2, y_2)

Asset 2

Connecting Twins Via Their Outputs

Collect data for each asset

Encode relationships between assets



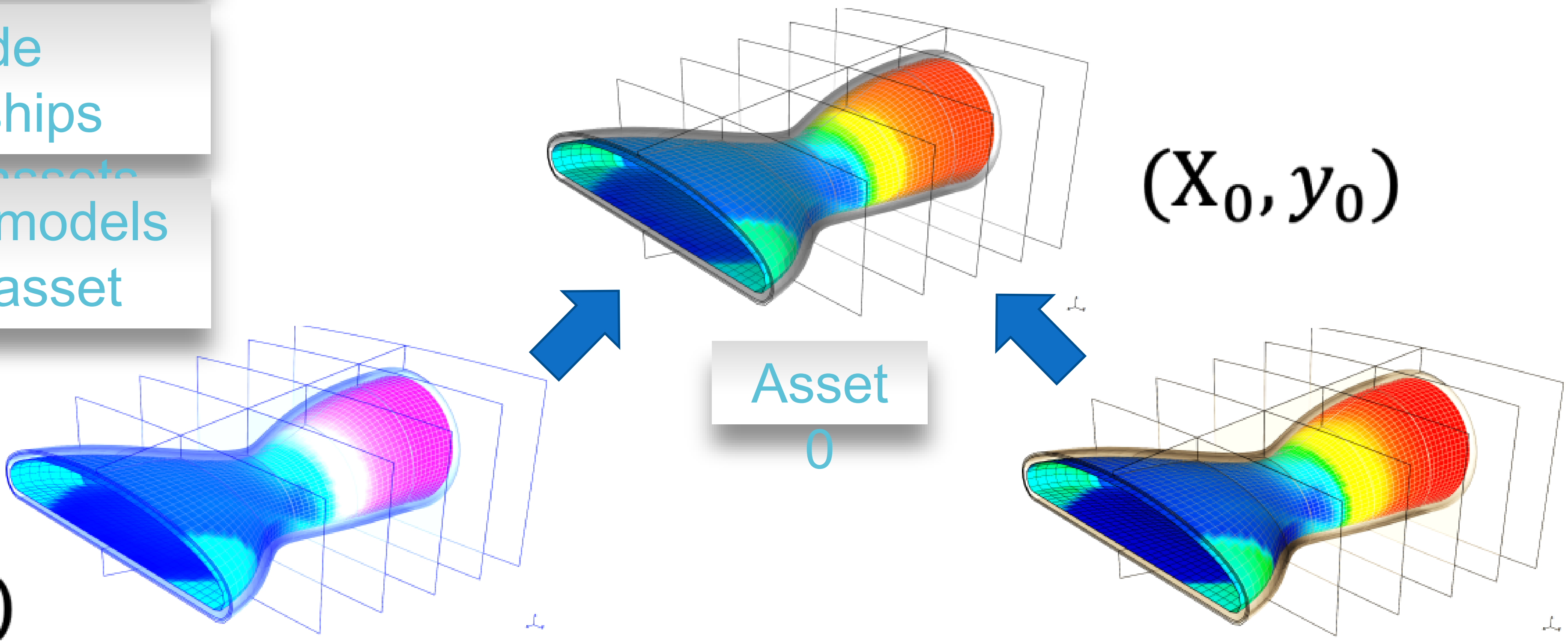
Connecting Twins Via Their Outputs

Collect data for each asset

Encode relationships between assets

Formulate models for each asset

$g_0(x, g_1(x;\theta_1), g_2(x;\theta_2);\theta_2)$



(X_0, y_0)

Asset 0

(X_1, y_1)

Asset 1

$_1(; _1)$

(X_2, y_2)

Asset 2

$_2(; _2)$

Output-Based Asset Class Learning

Place all data into G a directed acyclic graph (DAG)

$$y_G = [y_0^\top, y_1^\top, y_2^\top]^\top \qquad X_G = \{X_0, X_1, X_2\}$$

$$\theta_G = [\theta_0^\top, \theta_1^\top, \theta_2^\top]^\top$$

Use graph to formulate likelihood function

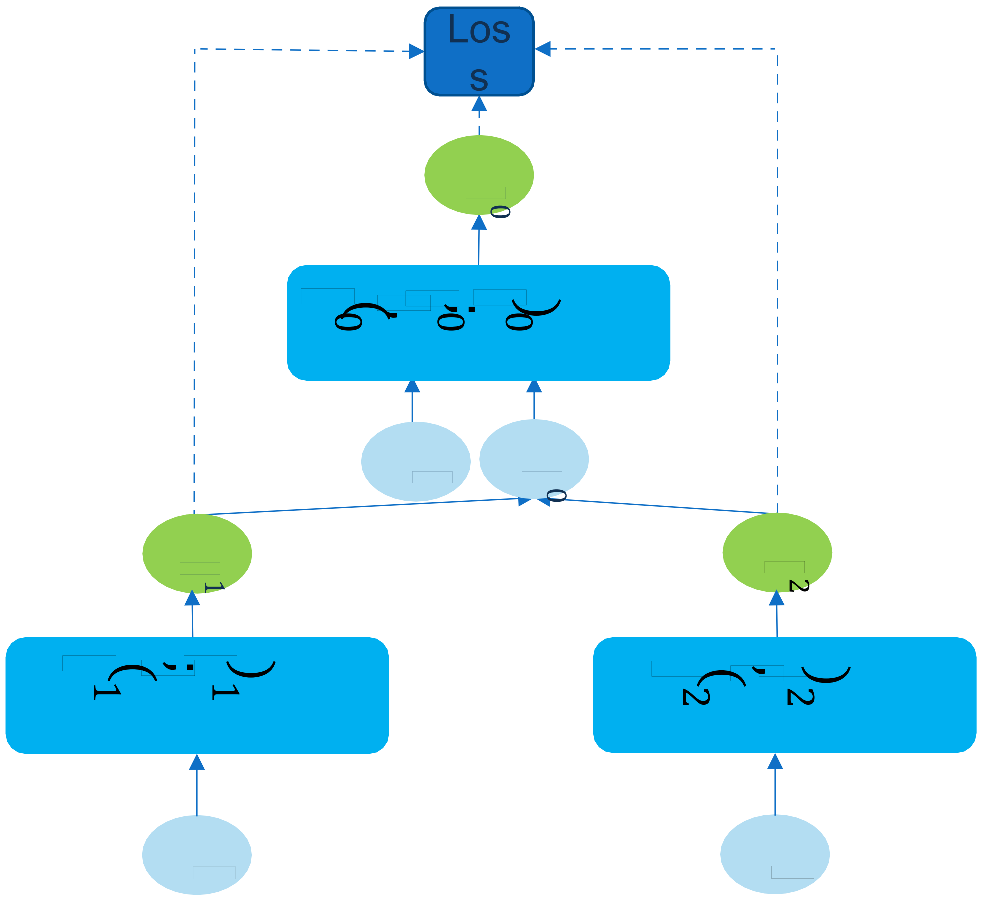
$$p(y_G \mid X_G, \theta_G) = p(y_0 \mid X_G, \theta_G) p(y_1 \mid X_1, \theta_1)\, p(y_2 \mid X_2, \theta_2)$$

For peer assets $k > 0$

$$\log p(y_k \mid X_k, \theta_k) \propto \|y_k - g_k(X_k; \theta_k)\|^2_{\Sigma_{\epsilon_k}}$$

For asset of interest $k = 0$

$$\log p(y_0 \mid X_G, \theta_G) = -\frac{N_0}{2}\log \pi - N_0 \log|\Sigma_{\epsilon_0}| - \frac{1}{2}(y_0 - g(X_0; \theta_G))^\top\ \Sigma_{\epsilon_0}^{-1}\ (y_0 - g(X_0; \theta_G))$$



Single Fidelity Gaussian Process (GPs)

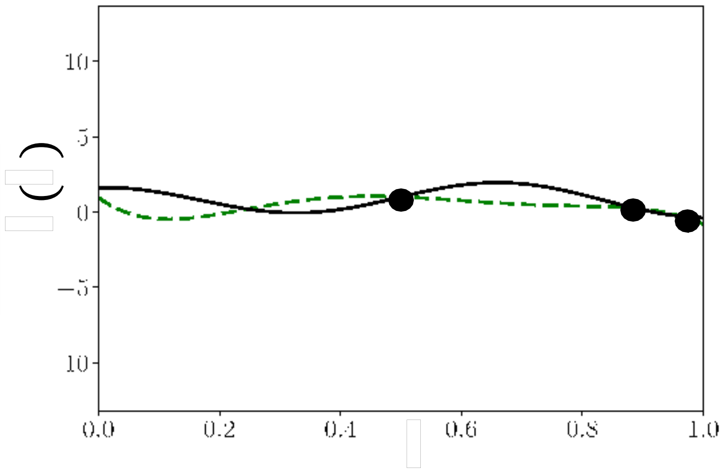
We will build multi-asset digital twins using Gaussian processes

Given data $(\boldsymbol{X}, \boldsymbol{y})$ and covariance kernel $\boldsymbol{C}$ a single assert GP posterior mean and variance are

$$m(x) = t(x)^{\top} C(X,X)^{-1} y$$

$$\sigma^2(x) = C(x,x) - t(x)^{\top} C(X,X)^{-1} t(x), \qquad t(x) = C(x,X)$$

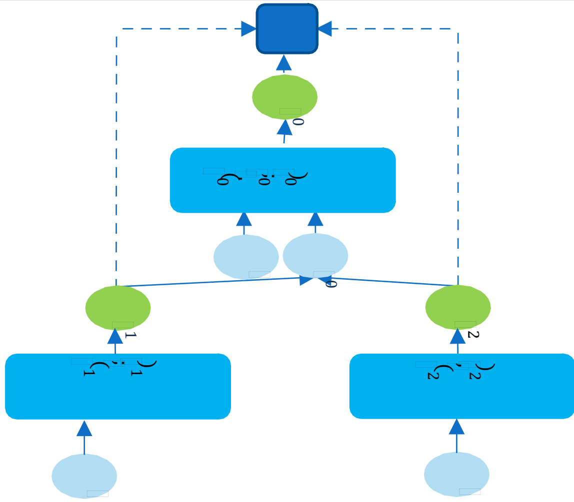
But approximation will be poor for limited data



Gaussian Process Discrepancy Modeling

Assume multiplicate and additive discrepancy

$$g_0(x, g_1(x), g_2(x); \theta_2) = \rho_1(x; \theta_2^{\rho_1})\mathrm{g}_1(\mathrm{x}) + \rho_2(x; \theta_2^{\rho_2})\mathrm{g}_2(\mathrm{x}) + \delta(x; \theta_2^{\delta})$$



Assume $\mathbf{g}_1, \mathbf{g}_2, \delta$ are Gaussian processes with correlation matrices $\mathbf{C}_0, \mathbf{C}_1, C_2$

$$C = \begin{bmatrix} C_0(X_0, X_0) + \rho_{12}^2 C_1(X_0, X_0) + \rho_{13}^2 C_2(X_0, X_0) & \rho_{12} C_1(X_0, X_1) & \rho_{13} C_2(X_0, X_2) \\ \rho_{12} C_1(X_1, X_0) & C_1(X_1, X_1) & 0 \\ \rho_{13} C_2(X_2, X_0) & 0 & C_2(X_2, X_2) \end{bmatrix}$$

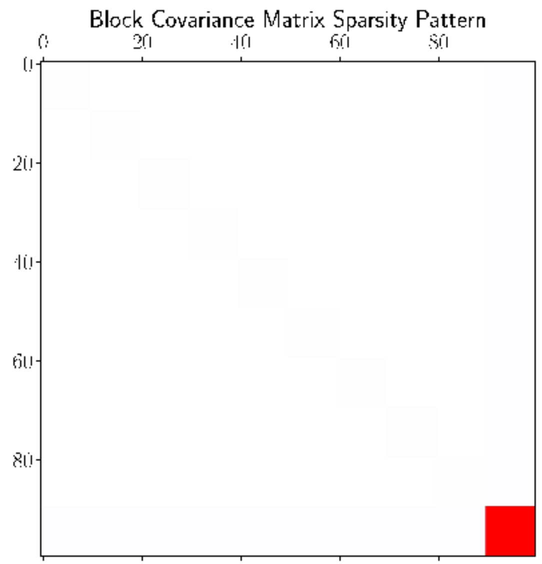
Training Multi-Asset GPs

Finding GP hyperparameters by minimizing the negative log likelihood is challenging

$$\mathrm{NLL} = \frac{1}{2}\log\left(\det \boldsymbol{C}\right) + \frac{1}{2}\boldsymbol{y}^T\boldsymbol{C}^{-1}\boldsymbol{y} + \frac{N}{2}\log\left(2\pi\right)$$

Naïve Cholesky factorization of C is expensive

$$O(N^3), \qquad N = \sum_{k=0}^{K} N_k$$



But we can exploit sparse covariance to efficiently evaluate NLL and its gradient

(left K=10 peers)

$$O(\hat{N}^3), \qquad \hat{N} = \max(N_k)$$

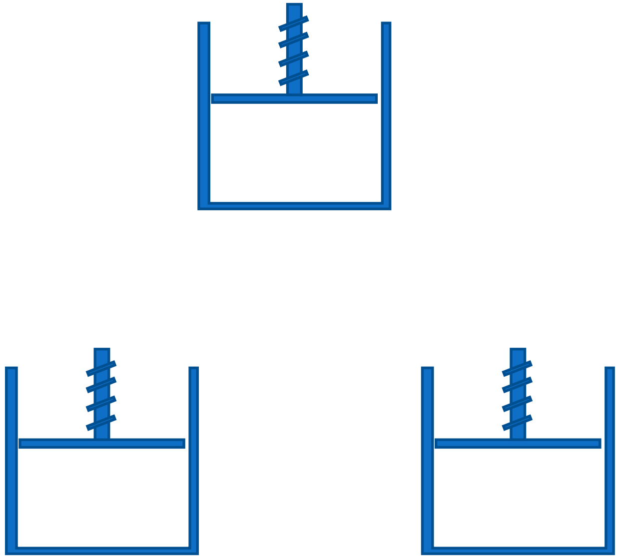
Piston Example

Predict cycle time of a piston as a function of piston weight and spring coefficient

$$C(\boldsymbol{x}) = 2\pi \sqrt{\frac{M}{k + S^2 \frac{P_0 V_0}{T_0} \frac{T_a}{V^2}}}$$

$$V = \frac{S}{2k} \left(\sqrt{A^2 + 4k \frac{P_0 V_0}{T_0} T_a} - A \right)$$

$$A = P_0 S + 19.62 M - \frac{k V_0}{S}$$



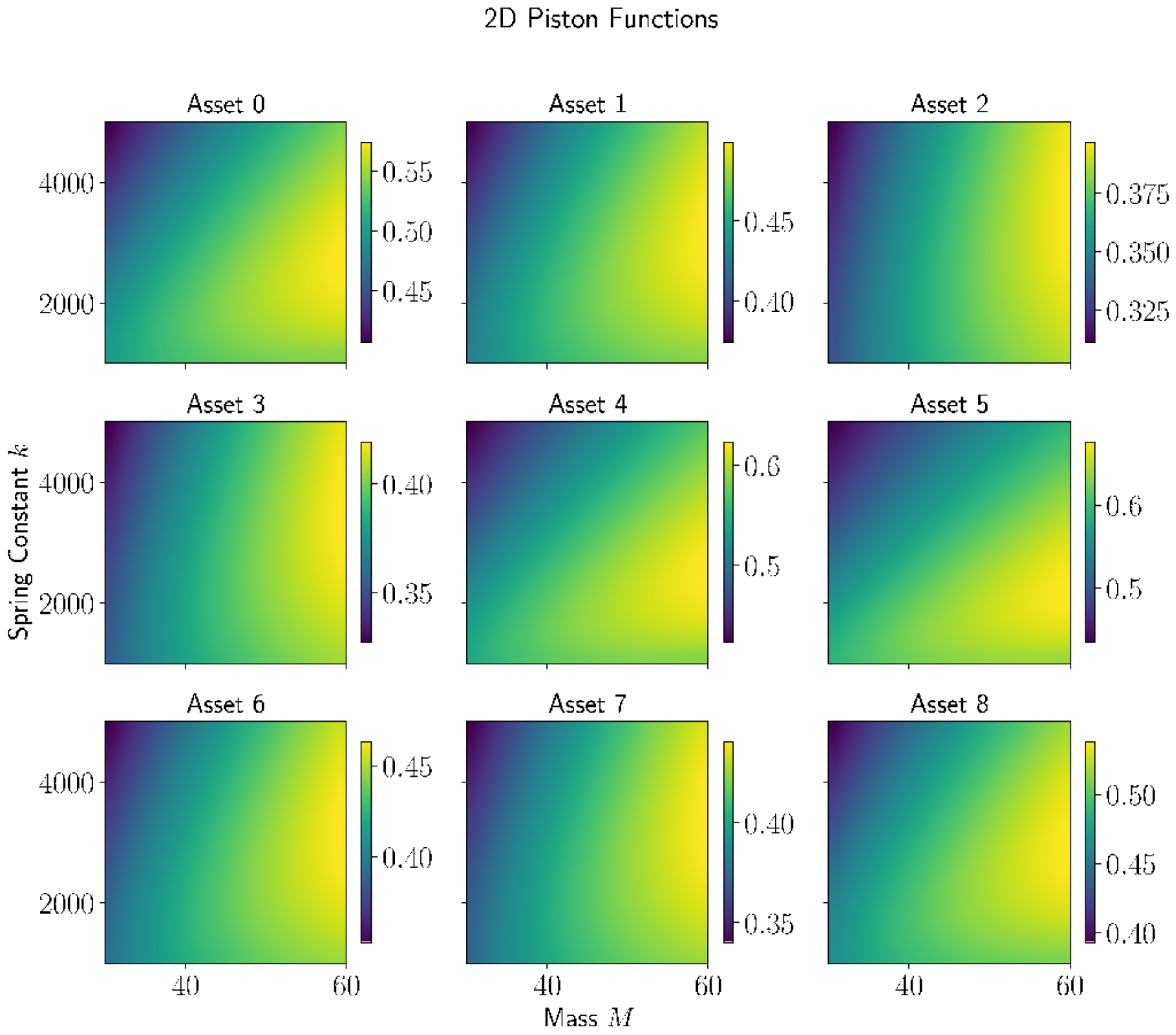
Input variables x

Latent variables

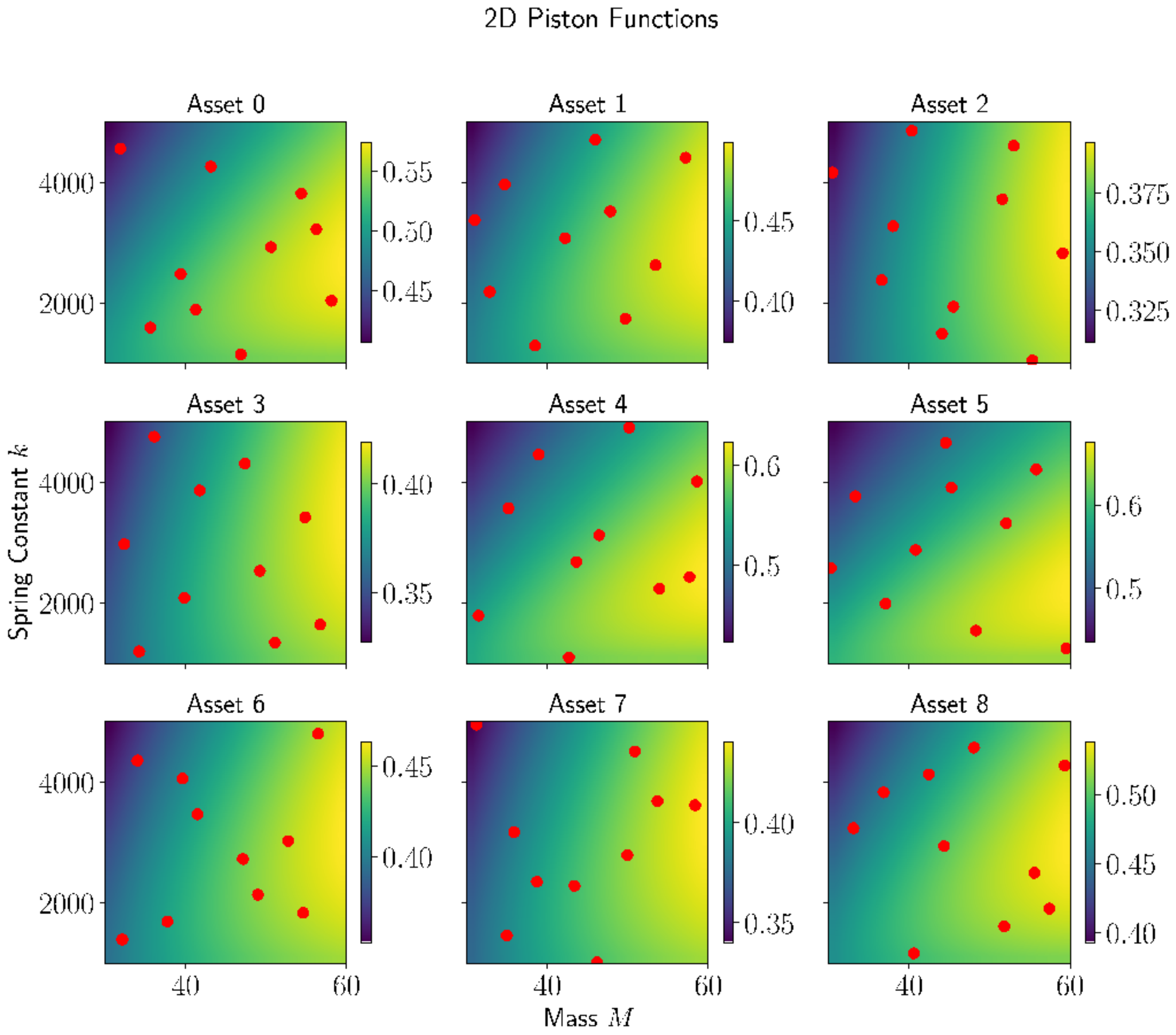
$M \in [30, 60]$	piston weight (kg)
$S \in [0.005, 0.020]$	piston surface area (m^2)
$V_0 \in [0.002, 0.010]$	initial gas volume (m^3)
$k \in [1000, 5000]$	spring coefficient (N/m)
$P_0 \in [90000, 110000]$	atmospheric pressure (N/m^2)
$T_a \in [290, 296]$	ambient temperature (K)
$T_0 \in [340, 360]$	filling gas temperature (K)

Benchmark problem for surrogate modeling:

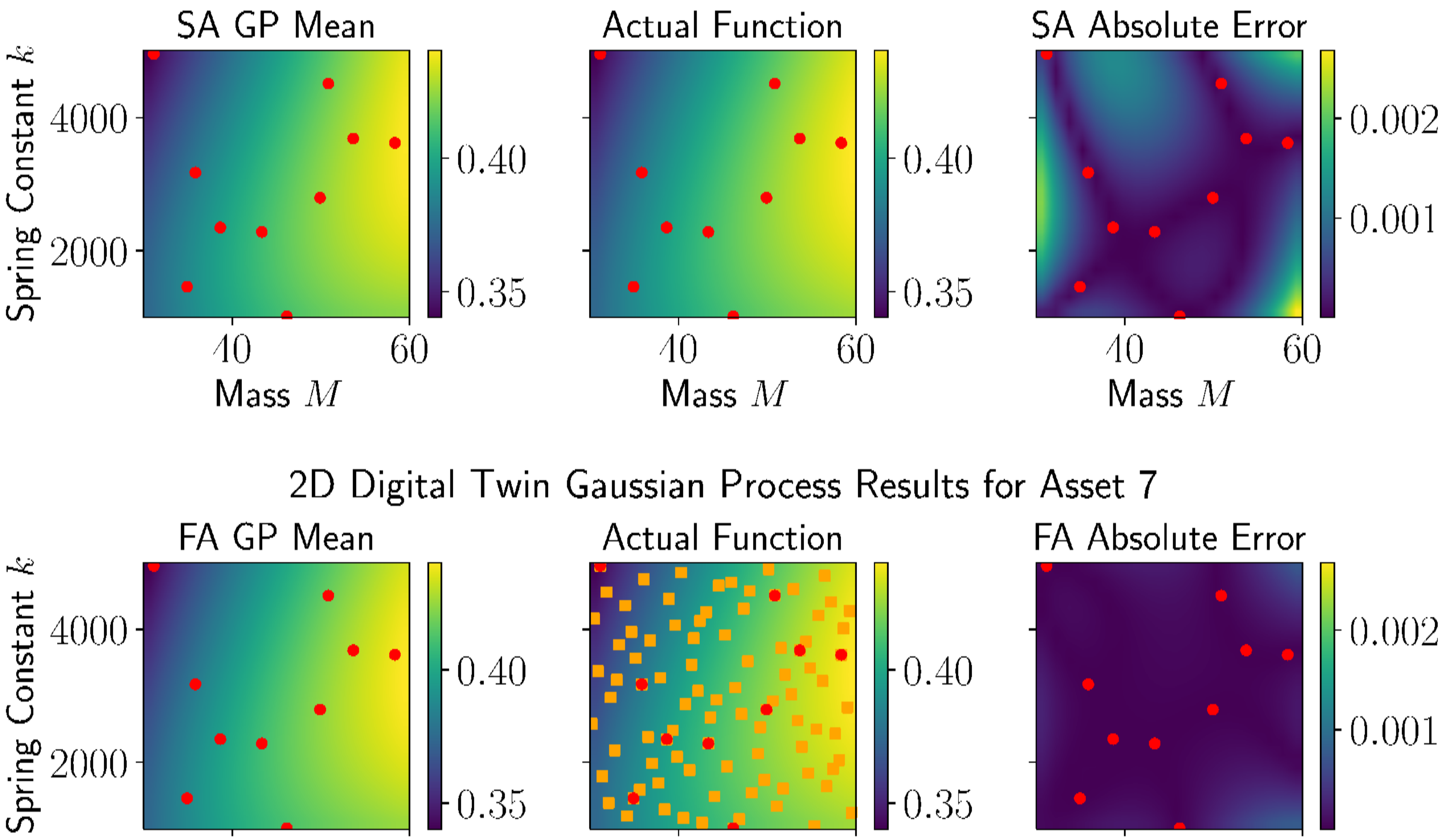
Piston Example: Response Surfaces



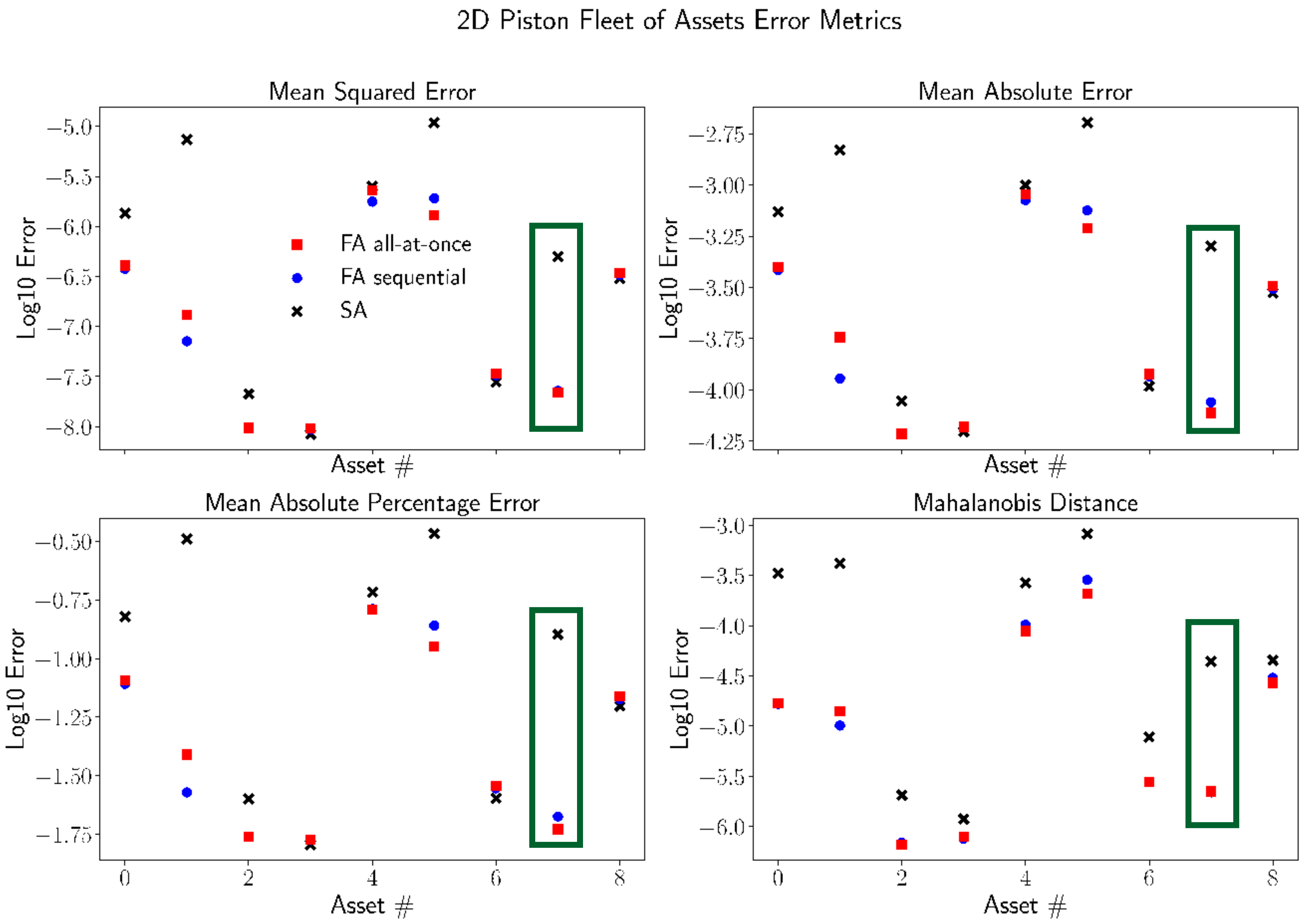
Piston Example: Training Data



PISTON EXAMPLE: ERROR FOR SINGLE ASSET



Piston Example: Error For All assets

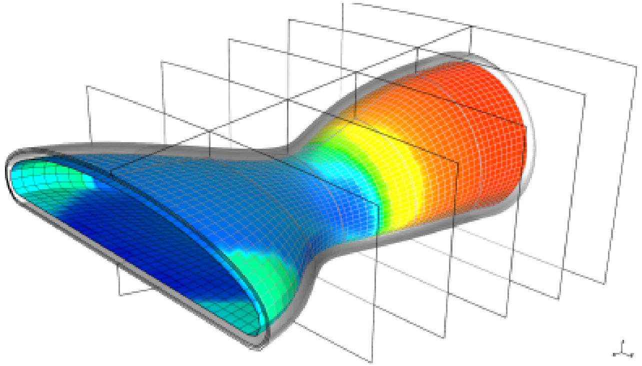


Connecting Twins Via Parameters

Collect data for each asset

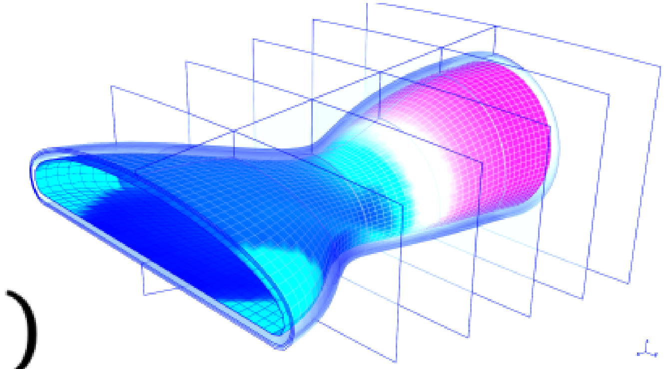
Formulate model for each asset

$g_0(x; \theta_1, \theta_2)$



(X_0, y_0)

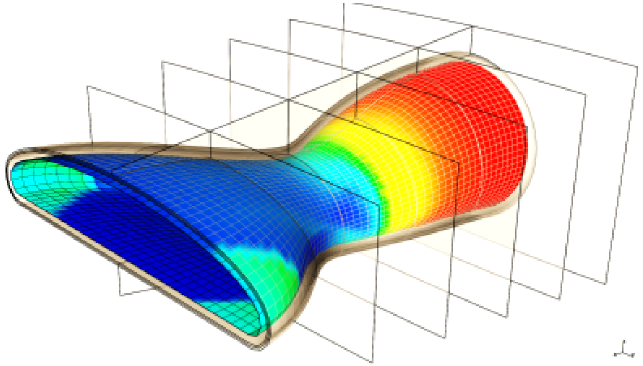
Asset 0



(X_1, y_1)

Asset 1

$_1(\; ; \; _1)$



(X_2, y_2)

Asset 2

$_2(\; ; \; _2)$

Latent Variable-Based Asset Class Learning

Assume variables are related by latent variables $\boldsymbol{\theta}$

$$\theta_G = [\theta_0, \theta_{\sim 0}^{\top}]^{\top} \quad \theta_0 = A\theta_{\sim 0} + b + \nu$$

Using hierarchical priors posterior is given by

$$p(\theta_G, A, b \mid y) \propto p(y \mid \theta_G, A, b)\, p(\theta_G, A, b) = p(y \mid \theta_G)\, p(\theta_{\sim 0}) p(A) p(b)$$

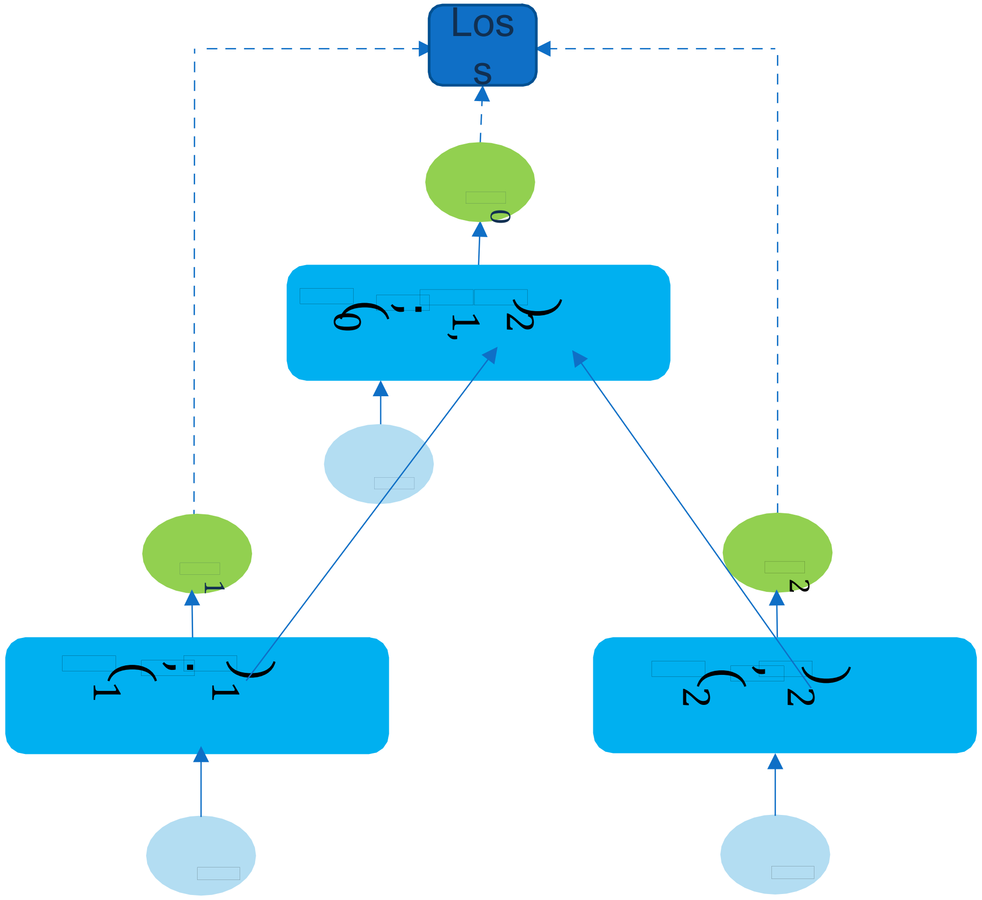
Log likelihood is

$$\log p(y|\theta_G) = \log p(y_0|\theta_G) + \sum_{k=1}^{K} \log p(y_k|\theta_k)$$

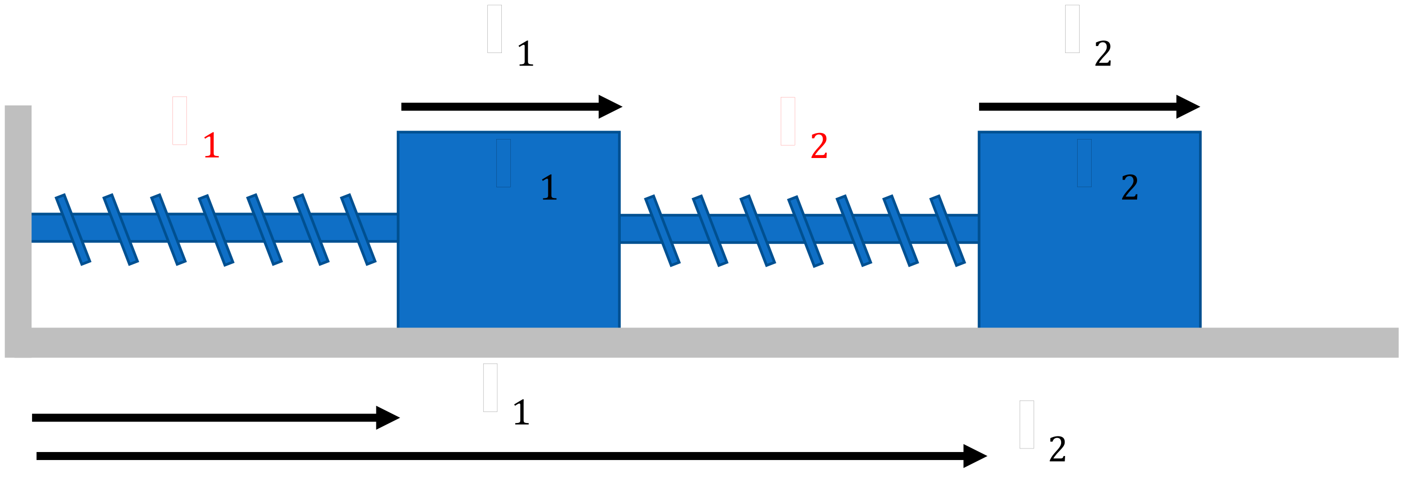
For asset of interest $k = 0$

$$\log p(y_0 \mid X_G, \theta_G) = -\frac{N_0}{2} \log \pi - N_0 \log |\Sigma_{\epsilon_0 \nu}| - \frac{1}{2} (y_0 - g_0(X_0; \gamma_G))^{\top} \Sigma_{\epsilon_0 \nu}^{-1} (y_0 - g_0(X_0; \gamma_G))$$

$\Sigma_{\epsilon_0 \nu} = \Sigma_{\epsilon_0} + \Phi \Sigma_\nu \Phi^{\top}$ Assuming linear model $g_0 = \Phi(X)\theta_0$



Spring System Example



$$m_1 x_1'' + b_1 x_1' + {\color{red}k_1}(x_1 - L_1) - {\color{red}k_2}(x_2 - x_1 - L_2) = 0$$
$$m_2 x_2'' + b_2 x_2' + {\color{red}k_2}(x_2 - x_1 - L_2) = 0$$

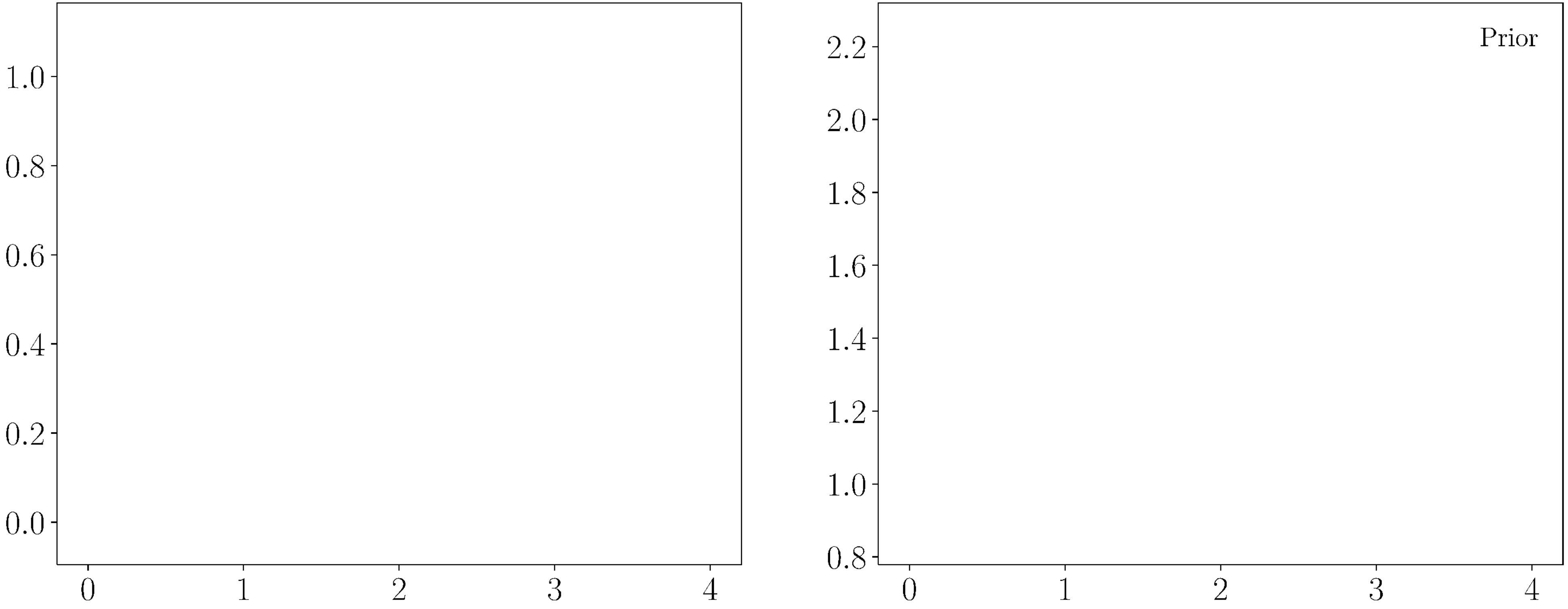
Assume spring coefficient has deteriorated differently for each asset

Assign a prior encoding prior knowledge

$$p(\theta_{\sim 0}) \sim N(\mu_{\sim 0}, \Sigma_{\sim 0})$$

$$A = \text{diag}(a), \qquad p(a) \sim N(1/K, \Sigma_a)$$

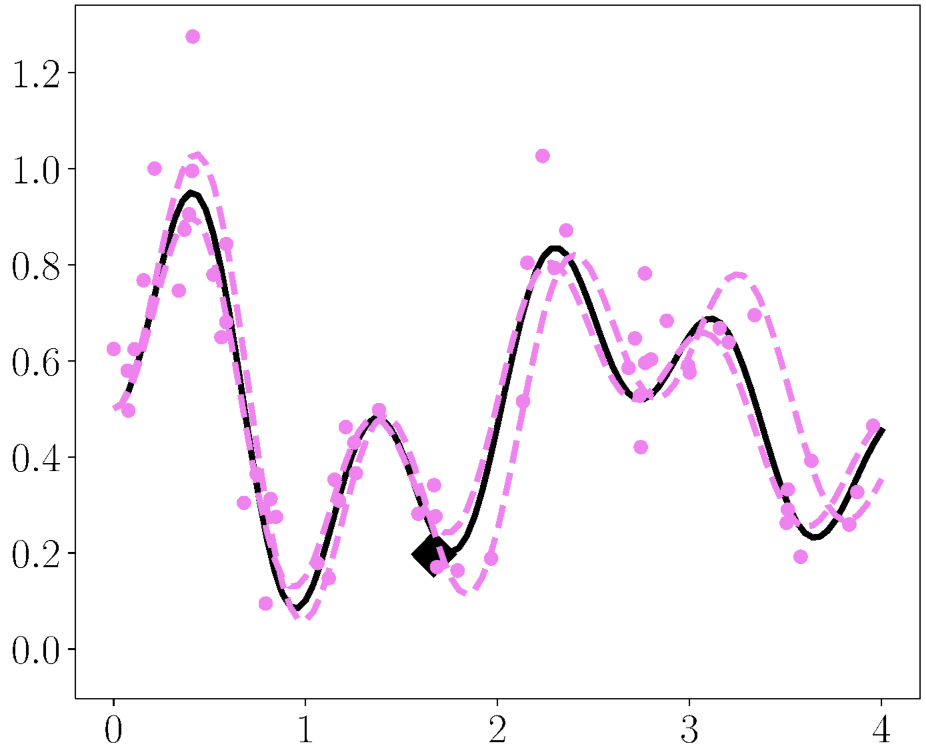
$$p(b) \sim N(0, \Sigma_b)$$

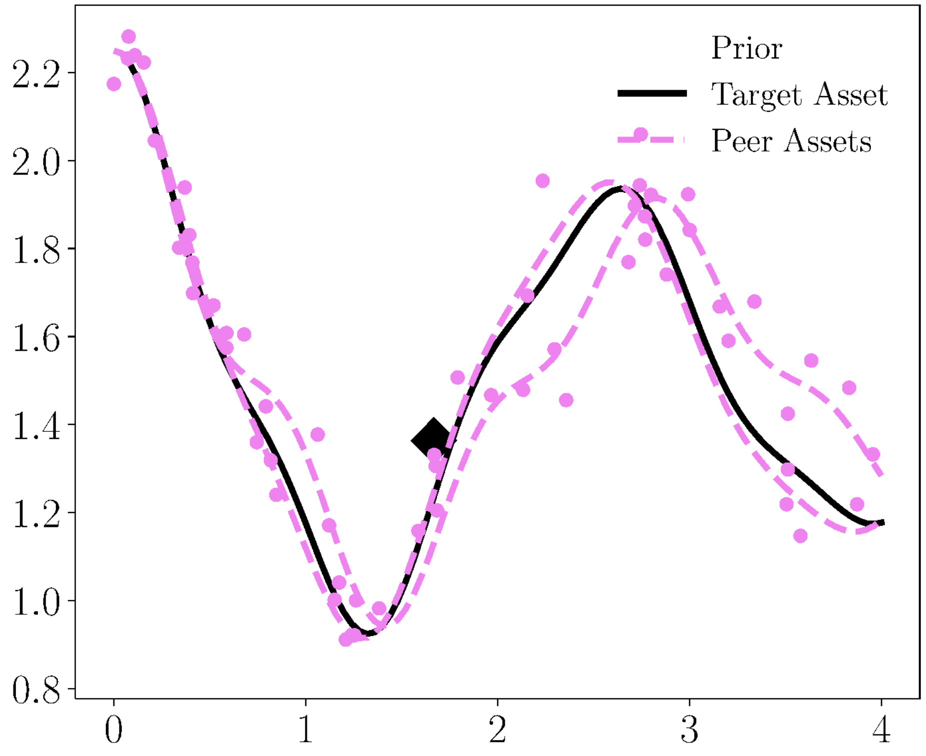


Prior predictive for target asset

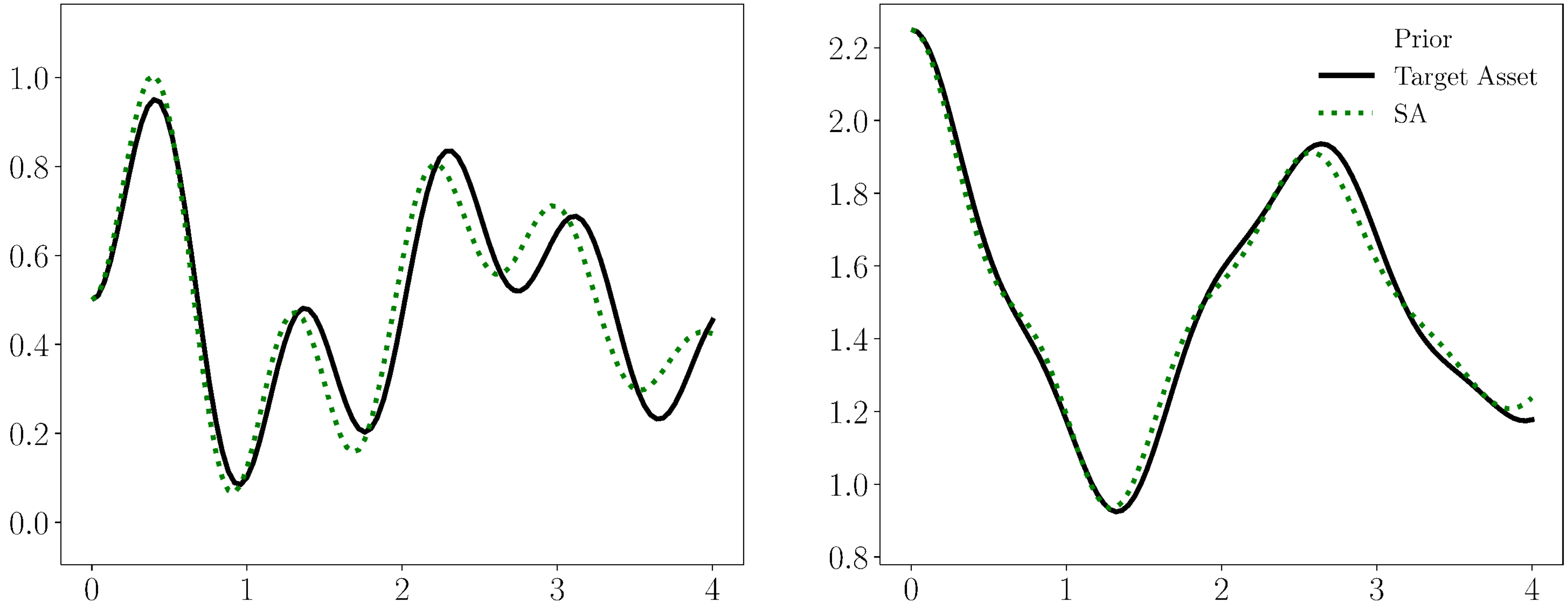
Collect data from all assets in the fleet

Using all assets we have a rich data set





USING ONLY DATA FROM A SINGLE ASSET PRODUCES A POOR DIGITAL TWIN



USING DATA FROM THE FLEET IMPROVES PERFORMANCE OF A SINGLE TWIN DRAMATICALLY