On solving/learning differential equations with kernels

Houman Owhadi

FAU DCN-AvH Seminar
Fri Oct 8, 2021

Problem

$$\mathcal{X} \xrightarrow{\quad f^\dagger \quad} \mathbb{R}$$

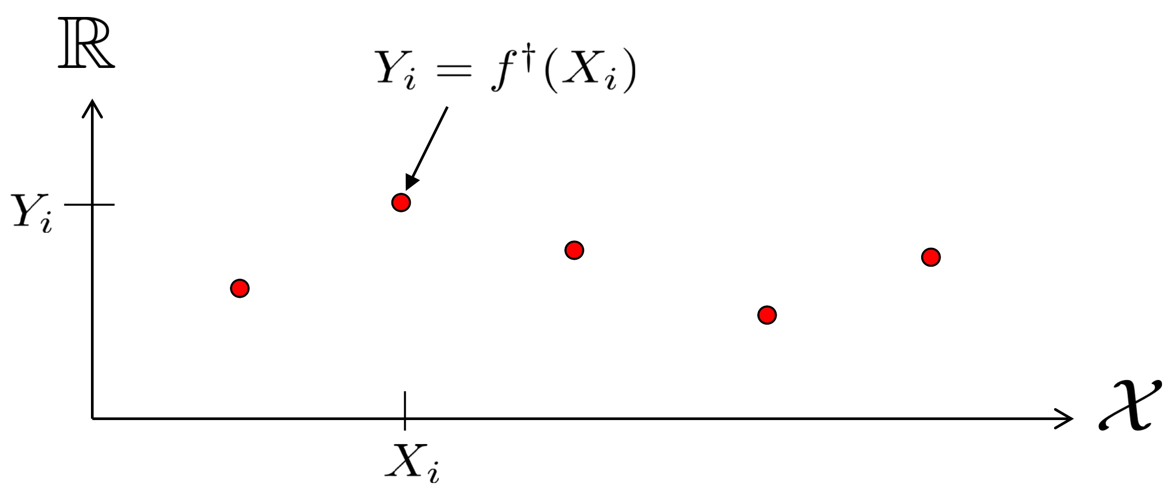
$f^\dagger$: Unknown

Given $f^\dagger(X) = Y$ with $(X, Y) \in \mathcal{X}^N \times \mathbb{R}^N$ approximate $f^\dagger$

$$X := (X_1, \ldots, X_N) \in \mathcal{X}^N$$

$$f^\dagger(X) := \big(f^\dagger(X_1), \ldots, f^\dagger(X_N)\big) \in \mathbb{R}^N$$

$$Y := (Y_1, \ldots, Y_N) \in \mathbb{R}^N$$



What is a kernel?

Kernel: $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

For all $m \geq 1$, $x_1, \ldots, x_m \in \mathcal{X}$ the $m \times m$ matrix with entries $K(x_i, x_j)$ is symmetric positive

⇕

Feature map:

$\exists$ a Hilbert space $\mathcal{F}$ and a map $\psi : \mathcal{X} \to \mathcal{F}$ such that

$$K(x, x') = \big\langle \psi(x), \psi(x') \big\rangle_{\mathcal{F}}$$

⇕

RKHS space: $\exists$ a Hilbert space $\mathcal{H} := \{f : \mathcal{X} \to \mathbb{R}\}$ such that

$f(x) = \langle f, K(x, \cdot) \rangle_{\mathcal{H}}$ for $x \in \mathcal{X}$, $f \in \mathcal{H}$

Write $\|f\|_K^2 := \|f\|_{\mathcal{H}}^2$

⇕

GP: $\exists$ a Gaussian process, $\xi : \mathcal{X} \to$ Gaussian space, such that

$$K(x, x') = \mathbb{E}\big[\xi(x)\xi(x')\big]$$

Write $\xi \sim \mathcal{N}(0, K)$

Kernel method solution

Kernel: Approximate $f^\dagger$ with

$$f(x) = K(x,X)K(X,X)^{-1}Y$$

$K(X,X)$: $N \times N$ matrix with entries $K(X_i, X_j)$

$K(x,X)$: $1 \times N$ vector with entries $K(x, X_i)$

Feature map: Approximate $f^\dagger$ with

$$f(x) = \langle \psi(x), c \rangle_{\mathcal{F}}$$

$c \in \mathcal{F}$ such that $f(X) = Y$ and $\|c\|_{\mathcal{F}}$ is minimal

RKHS space: Approximate $f^\dagger$ with minimizer of

Optimal recovery

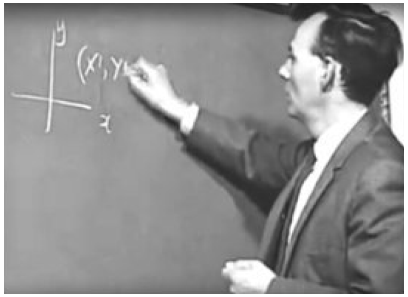
$$\begin{cases} \text{Minimize} & \|f\|_K \\ \text{subject to} & f(X) = Y \end{cases}$$

GPR: Approximate $f^\dagger$ with

$$f(x) = \mathbb{E}\left[\xi(x) \middle| \xi(X) = Y\right]$$

Most numerical approximation methods are kernel interpolation methods

Sard (1963)



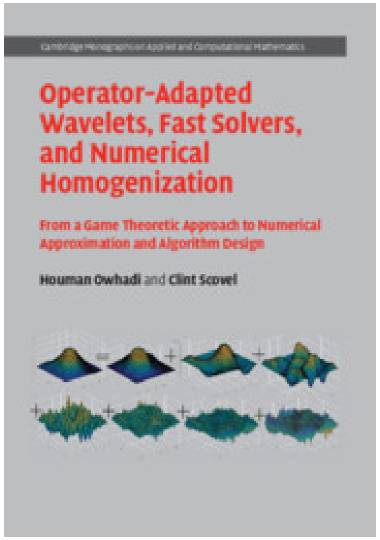
Larkin (1972)

Diaconis (1986)

See also: Sul'din (1959). Kimeldorf and Wahba (1970).

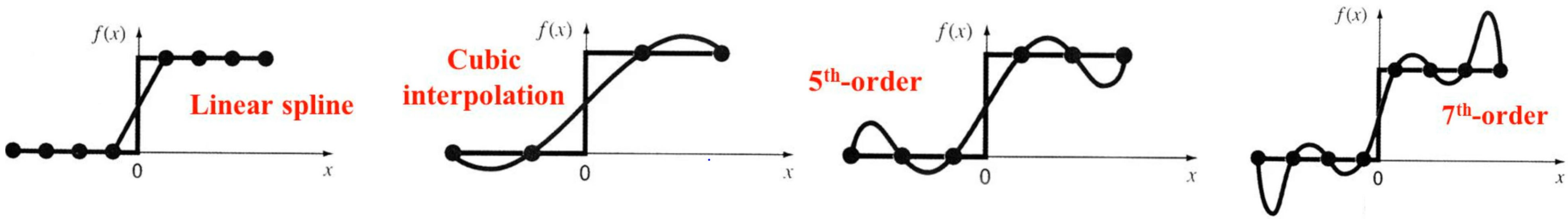
Survey: "Statistical Numerical Approximation", O., Scovel, Schäfer, 2019

Book: Cambridge University Press, O., Scovel, 2019



Cardinal splines

[Schoenberg, 1973]

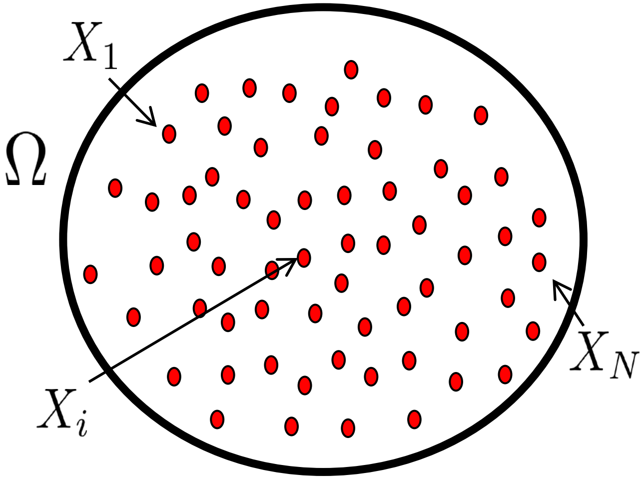


https://slideplayer.com/slide/4635359/

Cardinal spline interpolants are optimal recovery (kernel interpolants) splines

Polyharmonic splines [Harder and Desmarais, 1972], [Duchon, 1977]

$$\begin{cases} -\Delta f^\dagger = g, & x \in \Omega, \\ f^\dagger = 0, & x \in \partial\Omega, \end{cases} \qquad g \in L^2(\Omega)$$



$\Omega \subset \mathbb{R}^d$

$d \leq 3$

Problem: Given $f^\dagger(X)$ recover $f^\dagger$

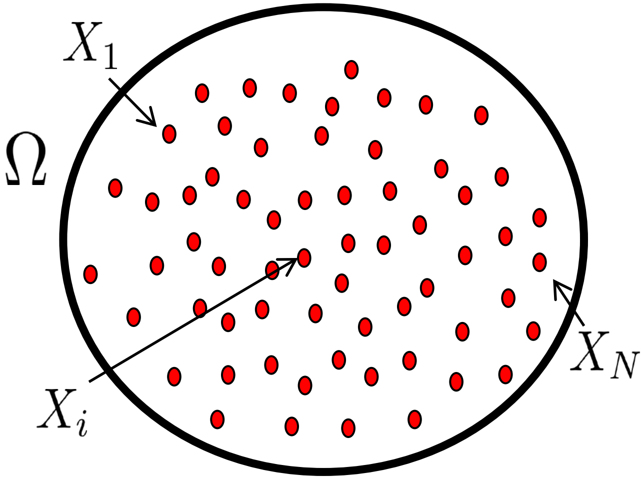
$$\begin{cases} \text{Minimize} & \int_\Omega |\Delta f|^2 \\ \text{subject to} & f(X) = Y \end{cases}$$

$$\|f^\dagger - f\|_{L^2(\Omega)} \lesssim N^{-\frac{2}{d}} \|g\|_L^2$$

The convergence can be arbitrarily bad if the kernel is not adapted

$$\begin{cases} -\operatorname{div}(a\nabla f^\dagger) = g, & x \in \Omega, \\ f^\dagger = 0, & x \in \partial\Omega, \end{cases}$$

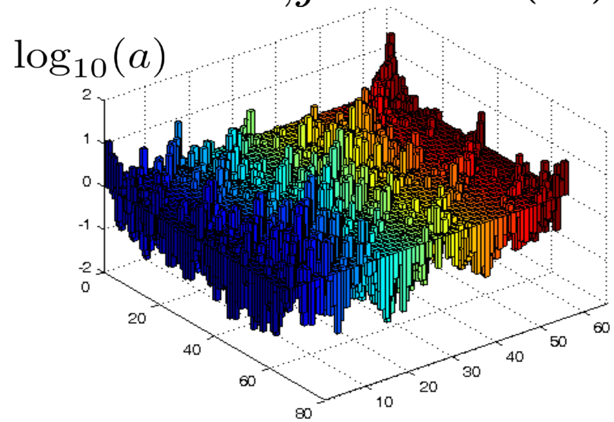
$g \in L^2(\Omega)$



$\Omega \subset \mathbb{R}^d$

$d \leq 3$

$a_{i,j} \in L^\infty(\Omega)$



$$\begin{cases} \text{Minimize} & \int_\Omega |\Delta f|^2 \\ \text{subject to} & f(X) = Y \end{cases}$$

$\|f^\dagger - f\|_{L^2(\Omega)} \geq \chi(N)$

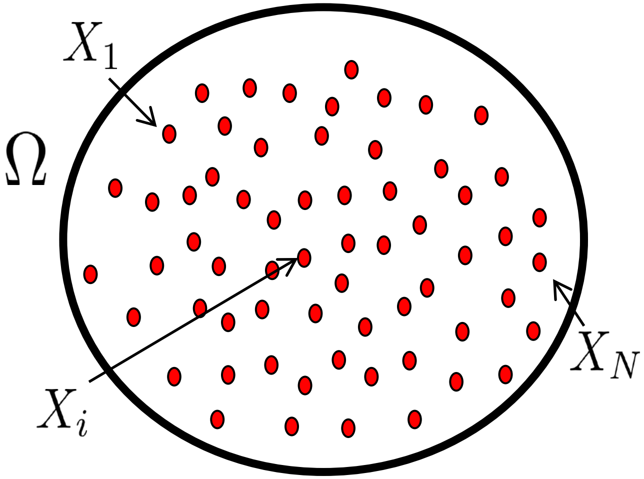
The convergence of $\chi(N)$ towards zero can be arbitrarily slow

[Babuška, Osborn, 2000]: Can a finite element method perform arbitrarily badly?

PDE adapted kernel

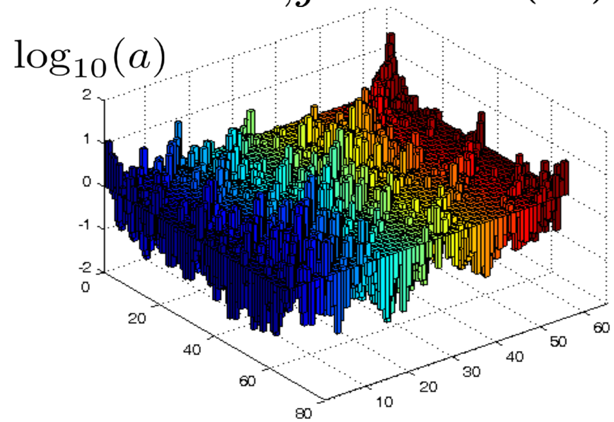
$$\begin{cases} -\operatorname{div}(a\nabla f^\dagger) = g, & x \in \Omega, \\ f^\dagger = 0, & x \in \partial\Omega, \end{cases}$$

$$g \in L^2(\Omega)$$



$$\Omega \subset \mathbb{R}^d$$
$$d \leq 3$$

$$a_{i,j} \in L^\infty(\Omega)$$



$$\begin{cases} \text{Minimize} & \int_\Omega \left|\operatorname{div}(a\nabla f)\right|^2 \\ \text{subject to} & f(X) = Y \end{cases}$$

$$\|f^\dagger - f\|_{L^2(\Omega)} \lesssim N^{-\frac{2}{d}} \|g\|^2_L$$

[O., Berlyand, Zhang, 2014]: Rough polyharmonic splines

PDE adapted Gaussian prior

$$\begin{cases} -\operatorname{div}(a\nabla f^\dagger) = g, & x \in \Omega, \\ f^\dagger = 0, & x \in \partial\Omega, \end{cases} \qquad g \in L^2(\Omega)$$

$$\begin{cases} -\operatorname{div}(a\nabla \xi) = \zeta, & x \in \Omega, \\ \xi = 0, & x \in \partial\Omega, \end{cases} \qquad \zeta \sim \mathcal{N}(0, \delta(x-x'))$$

$$f(x) = \mathbb{E}\big[\xi(x) \big| \xi(X) = Y\big] \qquad \|f^\dagger - f\|_{L^2(\Omega)} \lesssim N^{-\frac{2}{d}}$$

[O., 2014]: Bayesian Numerical Homogenization

[Malqvist, Peterseim, 2012-2014]: Local Orthogonal Decomposition.

[O., 2015], [O., Zhang, 2016], [O., Scovel, 2019], [Schäfer, Sullivan, O., 2017]: Gamblets

[Feischl, Peterseim, 2020]

[Schäfer, Katzfuss and O., 2020]

Learning methods for solving PDEs

ANNs

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations.
M. Raissi, P.Perdikaris, G.E. Karniadakis, JCP 2019

GPs

Gamblets: Bayesian Numerical Homogenization. H. Owhadi. SIAM MMS, 2015.
Multigrid with rough coefficients and Multiresolution operator decomposition from Hierarchical Information Games. H. Owhadi, SIREV, 2017
Operator adapted wavelets, fast solvers, and numerical homogenization from a game theoretic approach to numerical approximation and algorithm design. H. Owhadi and C. Scovel. Cambridge University Press, Cambridge Monographs on Applied and Computational Mathematics, 2019

Time dependent: Numerical Gaussian processes for time-dependent and nonlinear partial differential equations M Raissi, P Perdikaris, GE Karniadakis, SISC 2018

Probabilistic numerics: Cockayne, C. Oates, T. Sullivan, and M. Girolami, 2017
RBF collocation methods: R. Schaback and H. Wendland, 2006
Interplays with numerical approximation: Sard, Larkin, Diaconis, Suldin, Kimeldorf and Wahba

GPs: **More theoretically well-founded and with a long history of interplays with numerical approximation**
but were limited to linear/quasi-linear/time-dependent PDEs

Generalization of GP methods to arbitrary nonlinear PDEs

Solving and Learning Nonlinear PDEs with Gaussian Processes.
Y. Chen, B. Hosseini, H. Owhadi, AM. Stuart.
Journal of Computational Physics, Volume 447, 2021,
https://arxiv.org/abs/2103.12959

Properties

- Provably convergent for forward problems
- Interpretable and amenable to numerical analysis
- Solve forward and inverse problems
- Inherit the complexity of SOA solvers for dense kernel matrices
- Could be used to develop a theoretical understanding of ANN based methods

A simple prototypical non-linear PDE

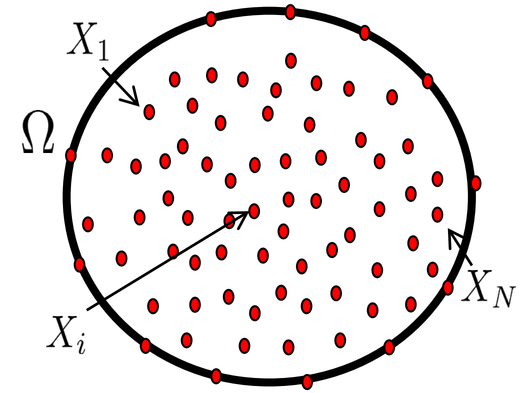
$$\begin{cases} -\Delta u^\dagger + \tau(u^\dagger) = f, & x \in \Omega, \\ u^\dagger = g, & x \in \partial\Omega, \end{cases}$$

$f : \Omega \to \mathbb{R}$, $g : \partial\Omega \to \mathbb{R}$ and $\tau : \mathbb{R} \to \mathbb{R}$: given continuous functions.

τ: Such that the PDE has a unique strong solution

Generalizes to arbitrary non-linear PDEs

$$\begin{cases} -\Delta u^\dagger + \tau(u^\dagger) = f, & x \in \Omega, \\ u^\dagger = g, & x \in \partial\Omega, \end{cases}$$



The method

$K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$: Given kernel.

$X_1, \ldots, X_N$: Collocation points on Ω and $\partial\Omega$

Approximate $u^\dagger$ with the minimizer u of

$$\begin{cases} \text{Minimize} & \|u\|_K^2 \\ \text{subject to} & -\Delta u(X_i) + \tau(u(X_i)) = f(X_i), \ \ X_i \in \Omega, \\ \text{and} & u(X_i) = g(X_i), \ \ X_i \in \partial\Omega, \end{cases}$$

Theorem

Assume that

- K is chosen so that
  - $\mathcal{H} \subset H^s(\Omega)$ for some $s > s^*$, where $s^* = \frac{d}{2} +$ order of PDE (order of PDE$= 2$)
  - $u^\dagger \in \mathcal{H}$
- Fill distance of $\{X_1, \ldots, X_N\}$ goes to zero as $N \to \infty$

Then, as $N \to \infty$

- $u \to u^\dagger$ pointwise in $\bar{\Omega}$
- $u \to u^\dagger$ in $H^t(\Omega)$ for $t < s$

$\mathcal{H}$: RKH space defined by kernel K

Implementation

$$\begin{cases} \text{Minimize} & \|u\|_K^2 \\ \text{subject to} & -\Delta u(X_i) + \tau(u(X_i)) = f(X_i), \quad X_i \in \Omega, \\ \text{and} & u(X_i) = g(X_i), \quad X_i \in \partial\Omega, \end{cases}$$

$\Updownarrow$

$$\begin{cases} \min_{z^{(1)},z^{(2)}} \begin{cases} \min_u \|u\|_K^2 \\ \text{s.t. } u(X_i) = z_i^{(1)} \text{ and } -\Delta u(X_i) = z_i^{(2)} \end{cases} \\ z_i^{(2)} + \tau(z_i^{(1)}) = f(X_i) \text{ for } X_i \in \Omega \\ z_i^{(1)} = g(X_i) \text{ for } X_i \in \partial\Omega \end{cases}$$

$\Updownarrow$

Reduction theorem

$$u(x) = K(x,\phi)K(\phi,\phi)^{-1} z$$

$$\begin{cases} \min_{z^{(1)},z^{(2)}} z^T K(\phi,\phi)^{-1} z \\ z_i^{(2)} + \tau(z_i^{(1)}) = f(X_i) \text{ for } X_i \in \Omega \\ z_i^{(1)} = g(X_i) \text{ for } X_i \in \partial\Omega \end{cases}$$

$$z = (z^{(1)}, z^{(2)})$$

$$\phi = (\phi^{(1)}, \phi^{(2)})$$

$$\phi_i^{(1)} = \delta_{X_i}$$

$$\phi_i^{(2)} = \delta_{X_i} \circ \Delta$$

$$\big(K(x,\phi)\big)_i = \int K(x,y)\phi_i(y)\, dy$$

$$\big(K(\phi,\phi)\big)_{i,j} = \int \phi_i(x) K(x,y) \phi_j(y)\, dx\, dy$$

$$\begin{cases} \min_{z^{(1)},z^{(2)}} z^T K(\phi,\phi)^{-1} z \\ z_i^{(2)} + \tau(z_i^{(1)}) = f(X_i) \text{ for } X_i \in \Omega \\ z_i^{(1)} = g(X_i) \text{ for } X_i \in \partial\Omega \end{cases}$$

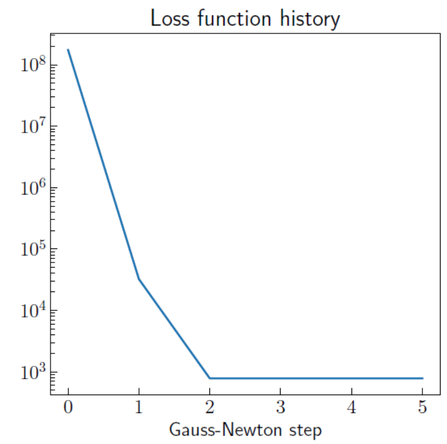
Eliminate $z^{(2)}$

$$\min_{z^{(1)}} \left(z_i^{(1)}, g(X_i), f(X_i) - \tau(z_i^{(1)})\right)^T K(\phi,\phi)^{-1} \left(z_i^{(1)}, g(X_i), f(X_i) - \tau(z_i^{(1)})\right)$$

Gauss-Newton Iteration

$$z_i^{(1),n+1} = z_i^{(1),n} + \delta z_i^{(1),n}$$

$$\min_{\delta z^{(1)}} Z^T K(\phi,\phi)^{-1} Z^T$$



$$Z = \left(z_i^{(1),n} + \delta z_i^{(1),n}, g(X_i), f(X_i) - \tau(z_i^{(1),n}) - \delta z_i^{(1),n} \nabla\tau(z_i^{(1),n})\right)$$

Converges in 2 to 7 steps

Inherits the complexity of fast linear solvers for $K(\phi,\phi)$

[Schäfer, Katzfuss and O., 2020]: $\mathcal{O}\left(N \log^{2d}(\frac{N}{\epsilon})\right)$ complexity

Gauss-Newton Iteration ⟺ Successive linerization of the PDE

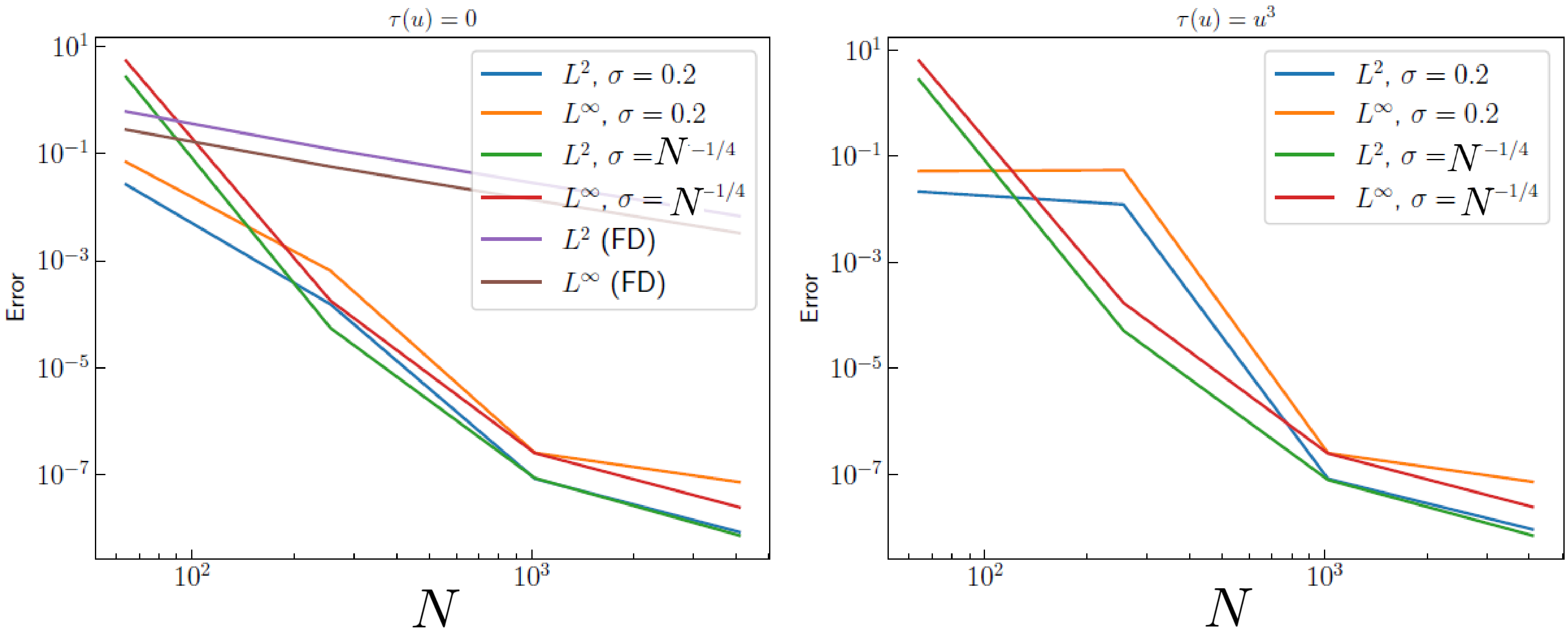
$$\begin{cases} -\Delta u^\dagger + \tau(u^\dagger) = f, & x \in \Omega, \\ u^\dagger = g, & x \in \partial\Omega, \end{cases}$$

$u^{n+1} = u^n + \delta u^n$

Given u^n solve for δu^n

$$\begin{cases} -\Delta(u^n + \delta u^n) + \tau(u^n) + \delta u^n \nabla \tau(u^n) = f, & x \in \Omega, \\ u^n + \delta u^n = g, & x \in \partial\Omega, \end{cases}$$

Numerical experiments



$$K(x, x') = \exp\left(- \frac{|x - x'|^2}{\sigma^2} \right)$$

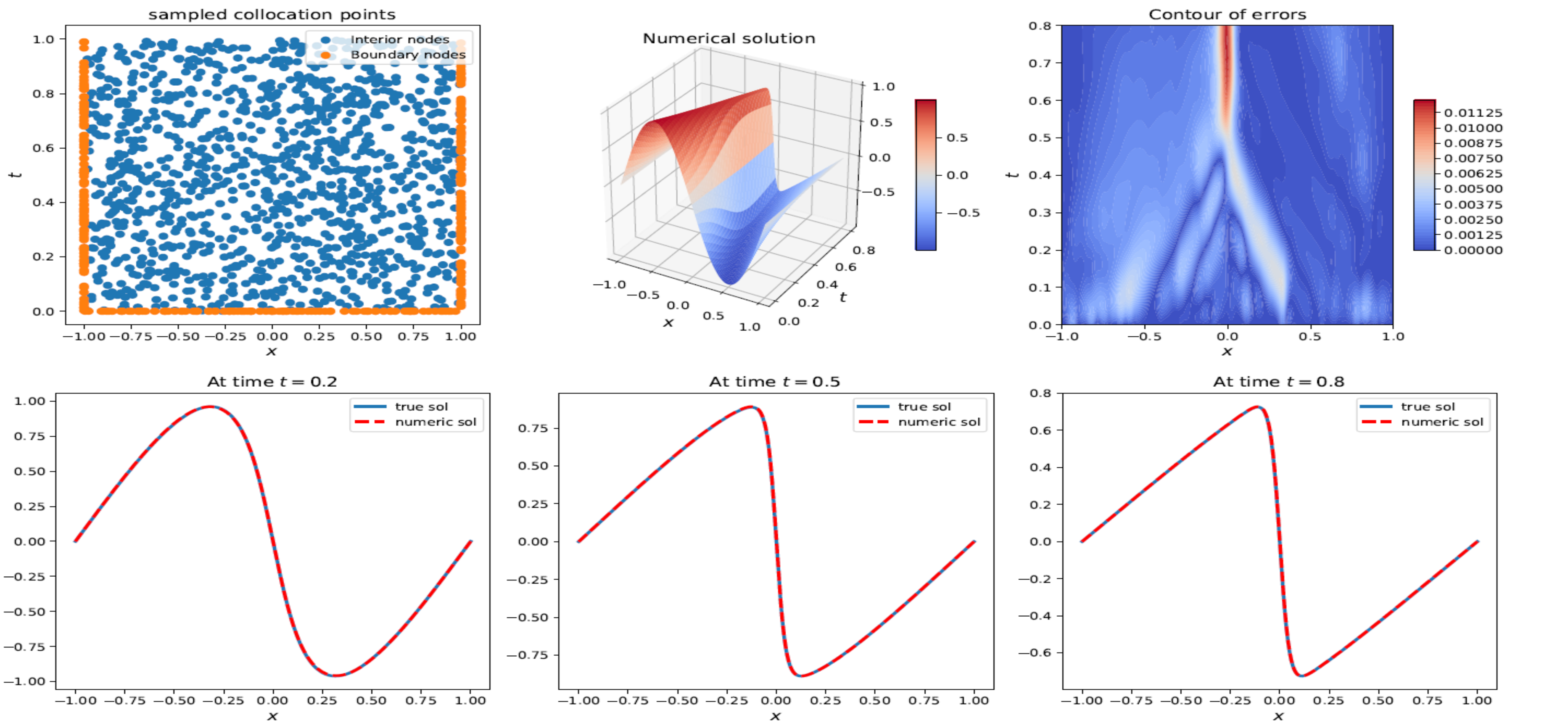
FD: Finite difference

Burger's

$$\partial_t u + u\partial_s u - \nu\partial_s^2 u = 0, \qquad \forall (s,t) \in [-1,1] \times [0,\infty),$$
$$u(s,0) = -\sin(\pi x),$$
$$u(-1,t) = u(1,t) = 0.$$

$$K\big((x,t),(x',t')\big) = \exp\big(-20|x-x'|^2 - 3|t-t'|^2\big)$$

N	600	1200	2400	4800
L^2 error	1.75e-02	7.90e-03	8.65e-04	9.76e-05
L^∞ error	6.61e-01	6.39e-02	5.50e-03	7.36e-04

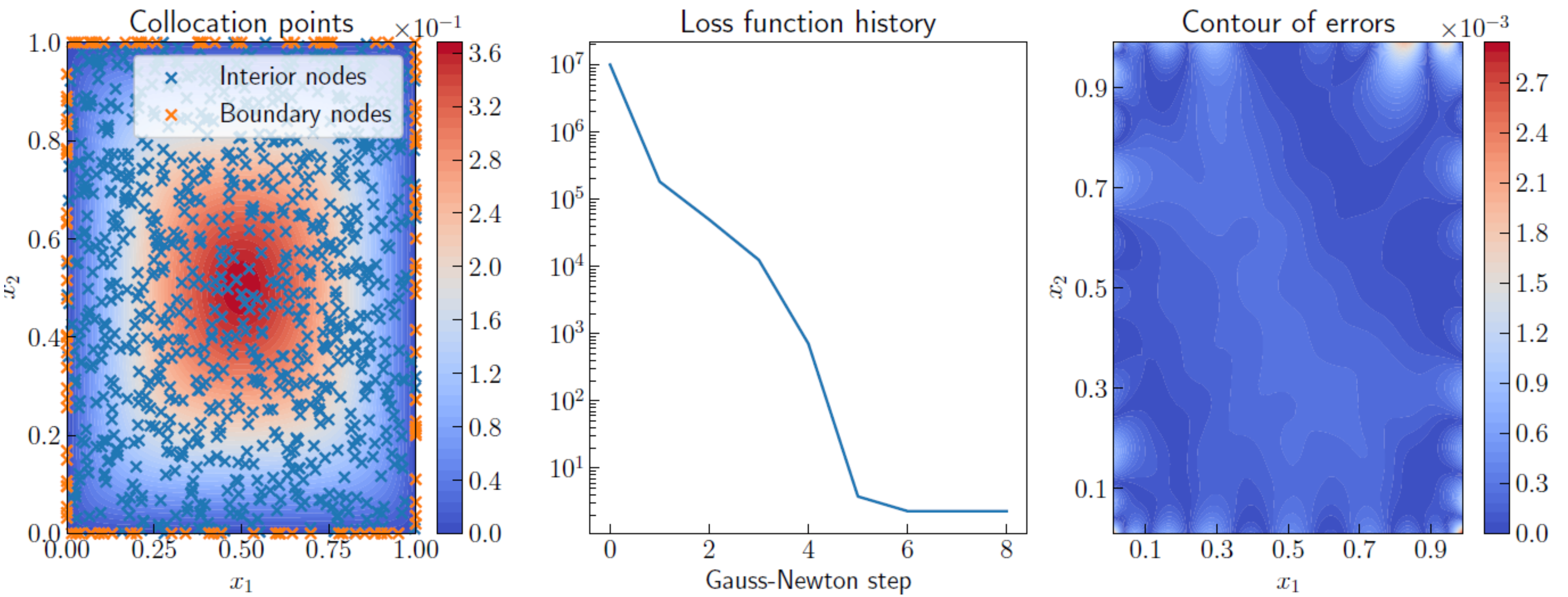


Eikonal

$$\begin{cases} \|\nabla u(x)\|^2 = f(x)^2 + \epsilon \Delta u(x), & \forall x \in \Omega\,, \\ u(x) = 0, & \forall x \in \partial\Omega\,, \end{cases}$$

$$K(x, x') = \exp\left(- \frac{|x-x'|^2}{\sigma^2} \right)$$

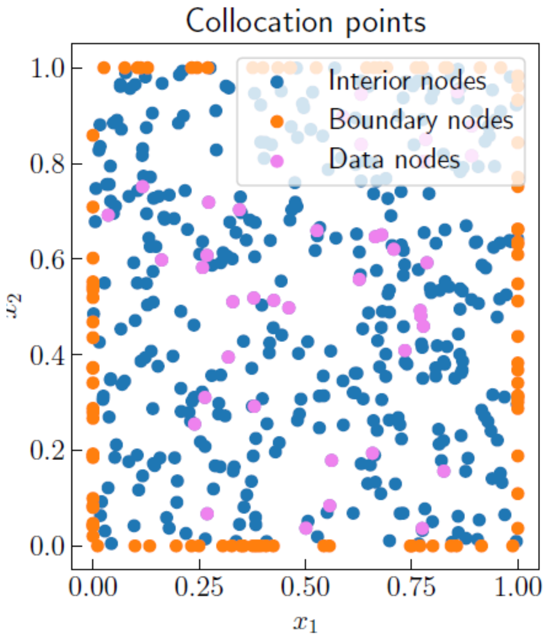
N	1200	1800	2400	3000
L^2 error	3.7942e-04	1.3721e-04	1.2606e-04	1.1025e-04
L^∞ error	5.5768e-03	1.4820e-03	1.3982e-03	9.5978e-04



Inverse Problem

$$\begin{cases} -\operatorname{div}\left(\exp(a)\nabla u\right)(x) = f(x), & x \in \Omega, \\ u(x) = 0, & x \in \partial\Omega. \end{cases}$$

a, u: Unknown. u observed at pink points.
Problem: Recover a and u.



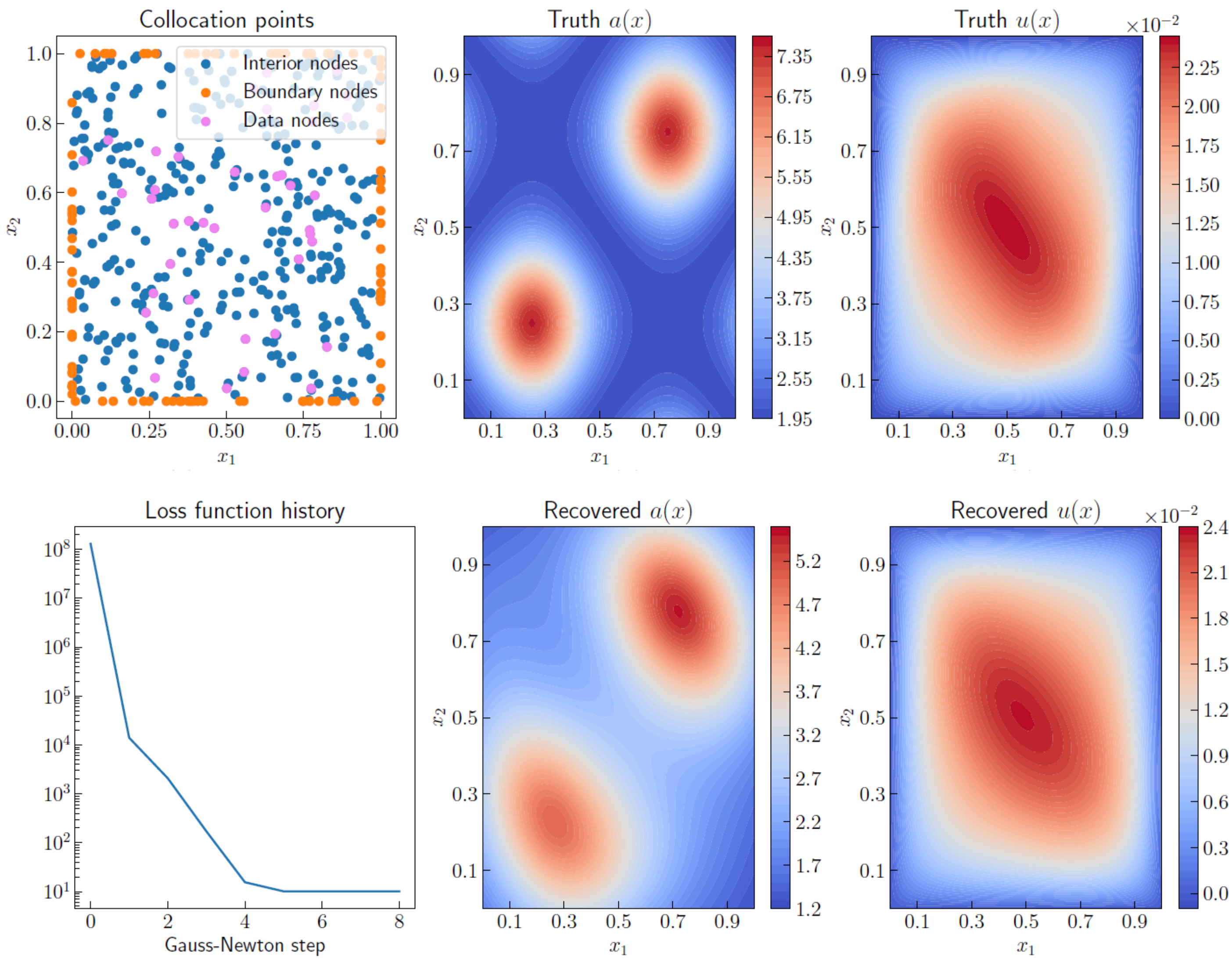
$$\begin{cases} \text{Minimize} & \|u\|_K^2 + \|a\|_\Gamma^2 \\ \text{subject to} & -\operatorname{div}\left(\exp(a)\nabla u\right)(X_i) = f(X_i), \;\; X_i \in \Omega, \\ \text{and} & u(X_i) = Y_i, \;\; (X_i, Y_i) \text{ is data point}, \\ \text{and} & u(X_i) = 0, \;\; X_i \in \partial\Omega, \end{cases}$$

Inverse Problem

$$\begin{cases} -\operatorname{div}\left(\exp(a)\nabla u\right)(x) = f(x), & x \in \Omega, \\ u(x) = 0, & x \in \partial\Omega. \end{cases}$$

a, u: Unknown. u observed at pink points.

Problem: Recover a and u.



Which kernel do we pick?

Kernel Flows: from learning kernels from data into the abyss.
H. Owhadi and G. R. Yoo, arXiv:1808.04475.
Journal of Computational Physics, 2019

Consistency of Empirical Bayes And Kernel Flow For Hierarchical Parameter Estimation. Y. Chen, H. Owhadi, A. M. Stuart.
Mathematics of Computation 2021, arXiv:2005.11375

Interpolation problem

$$\text{Recover } f^\dagger : D \subset \mathbb{R}^d \to \mathbb{R}$$

$$\text{Given } f^\dagger(X_1), \ldots, f^\dagger(X_N)$$

Family of kernels

$$K_\theta : D \times D \to \mathbb{R}$$

θ: Hierarchical parameter

Kernel/GP interpolant

$$\boxed{f(\cdot, \theta, X) = K_\theta(\cdot, X) K_\theta(X, X)^{-1} f^\dagger(X)}$$

$$f^\dagger(X) := \big(f^\dagger(X_1), \ldots, f^\dagger(X_N)\big) \in \mathbb{R}^N$$

$K_\theta(X, X)$: $N \times N$ matrix with entries $K_\theta(X_i, X_j)$

$K_\theta(x, X)$: $1 \times N$ vector with entries $K_\theta(x, X_i)$

Question

Which θ do we pick?

Empirical Bayes answer

Place a prior on θ

Assume that $f^\dagger|\theta \sim \mathcal{N}(0, K_\theta)$

Select the θ maximizing the marginal probability of θ subject to conditionning on $f^\dagger(X)$

Uninformative prior on θ

⬇

Maximum Likelihood Estimate

$$\theta^{EB} = \underset{\theta}{\text{argmin}}\, L^{EB}(\theta, X, f^\dagger)$$

$$L^{EB}(\theta, X, f^\dagger) = f^\dagger(X)^T K_\theta(X,X)^{-1} f^\dagger(X) + \log\det K_\theta(X,X)$$

Kernel Flow answer (Variant of cross-validation, O., Yoo, 2019)

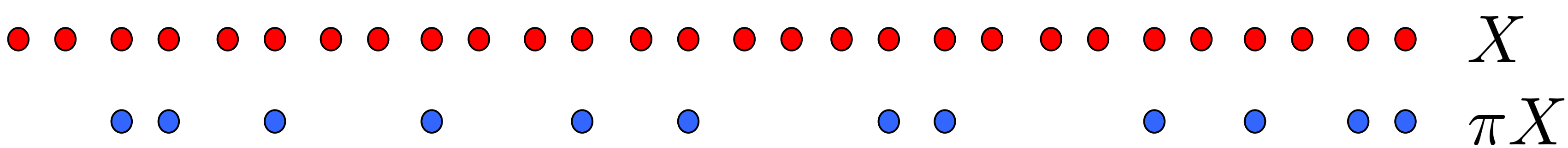
Pick a θ such that subsampling the data does not influence the interpolant much

$$\theta^{KF} = \underset{\theta}{\operatorname{argmin}}\, L^{KF}(\theta, X, \pi X, f^\dagger)$$

$$L^{KF}(\theta, X, \pi X, f^\dagger) = \frac{\left\| f(\cdot,\theta,X) - f(\cdot,\theta,\pi X) \right\|_{K_\theta}^2}{\left\| f(\cdot,\theta,X) \right\|_{K_\theta}^2}$$

$$f(\cdot, \theta, X) = K_\theta(\cdot, X) K_\theta(X, X)^{-1} f^\dagger(X)$$

π: subsampling operator, πX is a subvector of X



$\| \cdot \|_{K_\theta}$: RKHS norm determined by K_θ

A kernel is good if subsampling the data does not influence the interpolant much

Question

How do θ^{EB} and θ^{KF} behave as # of data $\to \infty$

Model

- Domain $D = \mathbb{T}^d = [0,1]^d_{\text{per}}$
- Lattice data $X_q = \{j \cdot 2^{-q}, j \in J_q\}$
  where $J_q = \{0, 1, \ldots, 2^q - 1\}^d$, # of data 2^{qd}
- Kernel $K_\theta = (-\Delta)^{-\theta}$
- Subsampling in KF: $\pi X_q = X_{q-1}$

Theorem

(Chen, O., Stuart, 2020)

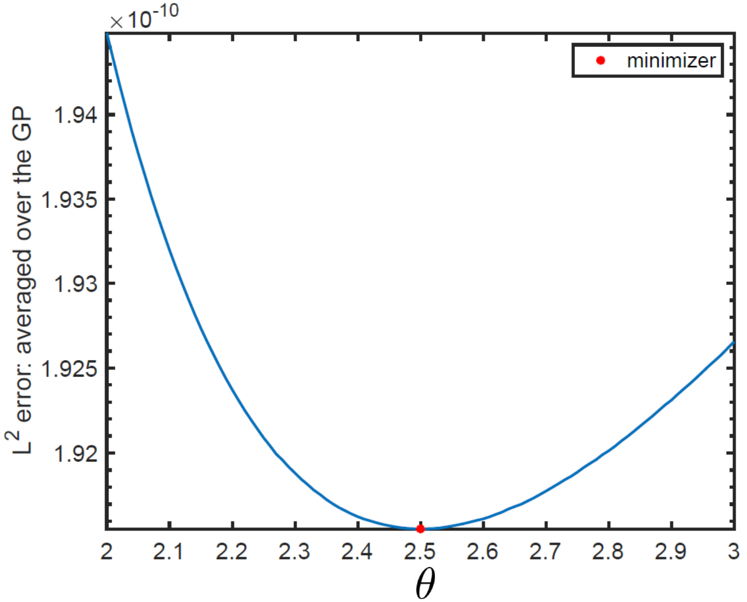
If $f^\dagger \sim \mathcal{N}(0, (-\Delta)^{-s})$ for some $s > d/2$, then as $q \to \infty$

$\theta^{EB} \to s$ and $\theta^{KF} \to \frac{s - \frac{d}{2}}{2}$ in probability

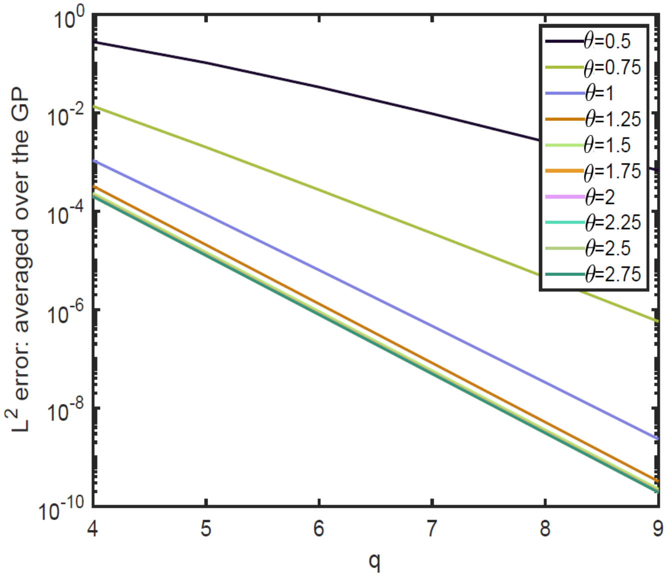
Question? How are the limits s and $\frac{s-\frac{d}{2}}{2}$ special?

Experiment $d = 1,\ s = 2.5,\ \#\text{ of data } N = 2^9$

L^2 error vs θ



L^2 error vs log(# data points)



- $s\ (= 2.5)$ is the θ that minimizes the mean squared error
- $\frac{s-\frac{d}{2}}{2}\ (= 1)$ is the smallest θ that suffices to achieve fastest rate in L^2

Takeaway message

- EB selects the θ that minimizes the mean squared error.
- KF selects the smallest θ that suffices for the fastest rate of convergence in mean squared error.

More comparisons

- EB may be brittle (not robust) to model misspecification
- KF has some degree of robustness to model misspecification

G. Wahba and J. Wendelberger. Some new mathematical methods for variational objective analysis using splines and cross validation. 1980.

M. L. Stein. A comparison of generalized cross validation and modified maximum likelihood for estimating the parameters of a stochastic process. 1990.

F. Bachoc. Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. 2013.

Chen, O., Stuart. Consistency of Empirical Bayes And Kernel Flow For Hierarchical Parameter Estimation. 2020.

Extrapolation problem

Given time series $z_1, \ldots, z_N$

predict $z_{N+1}, z_{N+2}, z_{N+3}, \ldots$

Assumption

$$z_{k+1} = f^\dagger(z_k, \ldots, z_{k-\tau^\dagger+1})$$

$f^\dagger$, $\tau^\dagger$ unknown

Fundamental problem

[Box, Jenkins, 1976]: Time Series Analysis

Mezíc, Klus, Budišić, R. Mohr,...: Koopman operator

[Alexander, Giannakis, 2020]: Operator theoretic framework

[Bittracher et al, 2019]: kernel embeddings of transition manifolds

[Brunton, Proctor, Kutz, 2016]: SINDy

Brian, Hunt, Ott, Pathak, Lu, Hunt, Girvan, Ott,...: Reservoir computing

Ralaivola, Chattopadhyay,...: LSTM

Schneider, Stuart, Wu: ensemble Kalman inversion

Simplest solution

Approximate $f^\dagger$ with Kernel interpolant f

$$\boxed{f(z_k, \dots, z_{k-\tau^\dagger+1}) = z_{k+1}} \qquad k = \tau^\dagger, \tau^\dagger + 1, \dots, N-1$$

$$\boxed{f(x) = K(x, X)K(X, X)^{-1}Y}$$

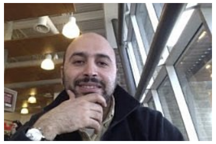
$$X_k = (z_k, \dots, z_{k-\tau^\dagger+1})$$

$$Y_k = z_{k+1} = f^\dagger(X_k)$$

Predict future values of the time series by simulating the dynamical system

$$\boxed{s_{k+1} = f(s_k, \dots, s_{k-\tau^\dagger+1})}$$

Learning dynamical systems from data: a simple cross-validation perspective. B. Hamzi and H. Owhadi. 2020. arXiv:2007.05074
Physica D nonlinear phenomena, 2021



Boumediene Hamzi

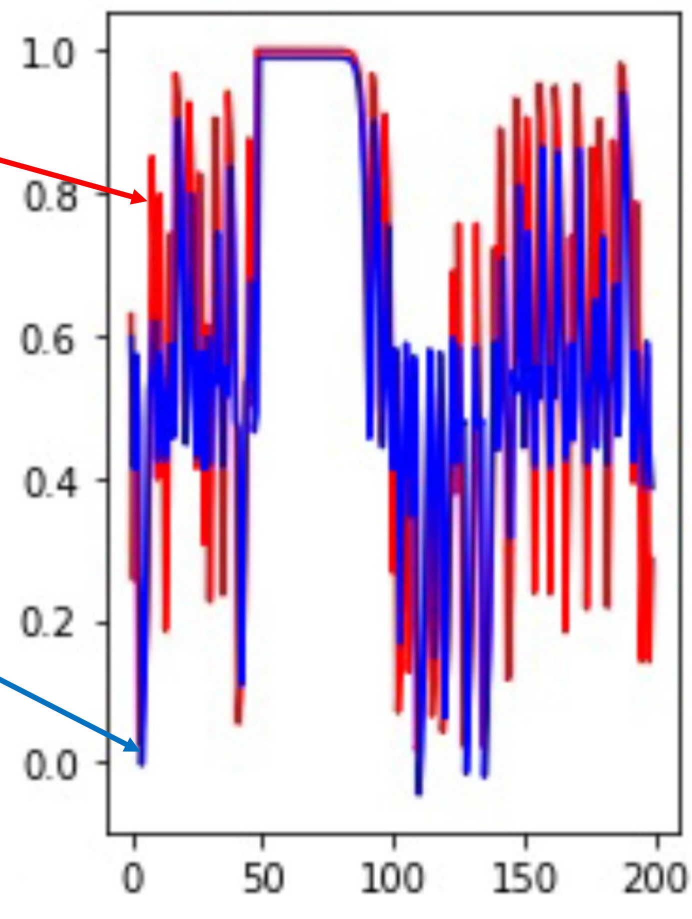
Example: Bernoulli map

$$z_{k+1} = 2z_k \mod 1$$

$$K(x, x') = e^{-\|x-x'\|^2}$$

True dynamic

Predicted dynamic



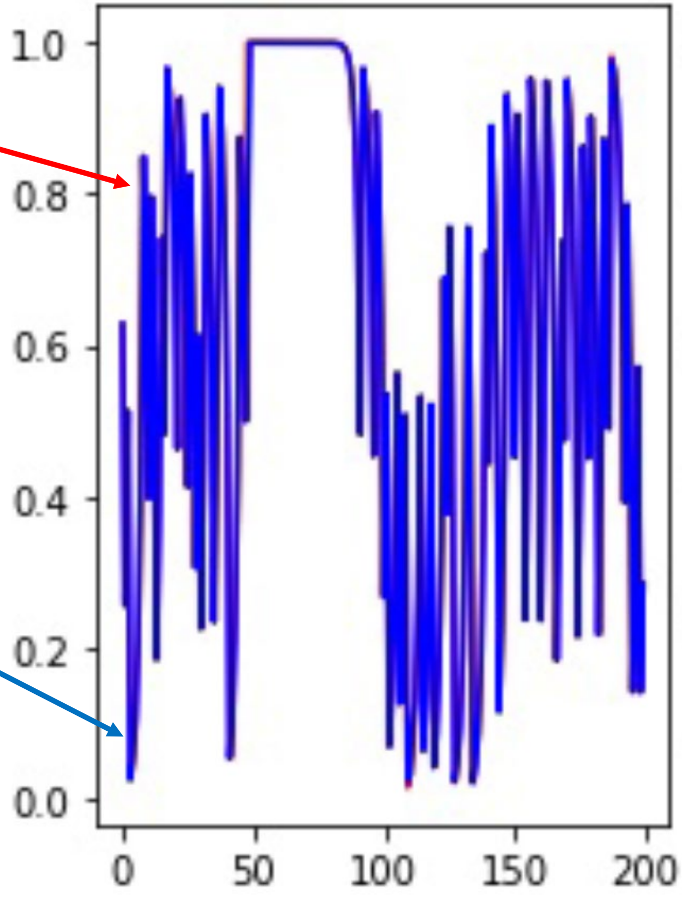
Example: Bernoulli map

$$z_{k+1} = 2z_k \mod 1$$

$$K(x,x') = \alpha_0 \max\{0, 1 - \frac{\|x-x'\|^2}{\sigma_0}\} + \alpha_1 e^{-\frac{\|x-x'\|^2}{\sigma_1^2}}$$

True dynamic

Predicted dynamic



$\alpha_0, \sigma_0, \alpha_1, \sigma_1^2$:
Learned parameters
(using Kernel Flows)

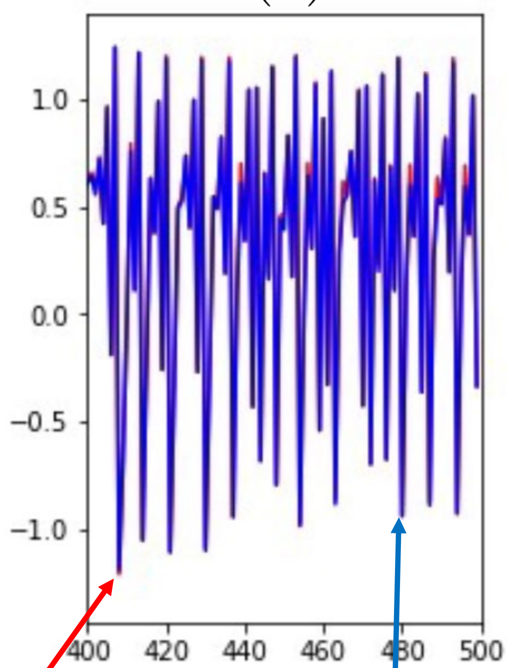
Example: Hénon map

$$
\begin{aligned}
x(k+1) &= 1 - ax(k)^2 + y(k) \\
y(k+1) &= bx(k)
\end{aligned}
$$

$$
K(x, x') = \begin{pmatrix} k_1(x, x') & 0 \\ 0 & k_2(x, x') \end{pmatrix}
$$

$$
k_i(x,y) = \alpha_i + (\beta_i + ||x-y||_2^{\kappa_i})^{\sigma_i} + \delta_i e^{-||x-y||_2^2/\mu_i^2}
$$

$x(k)$



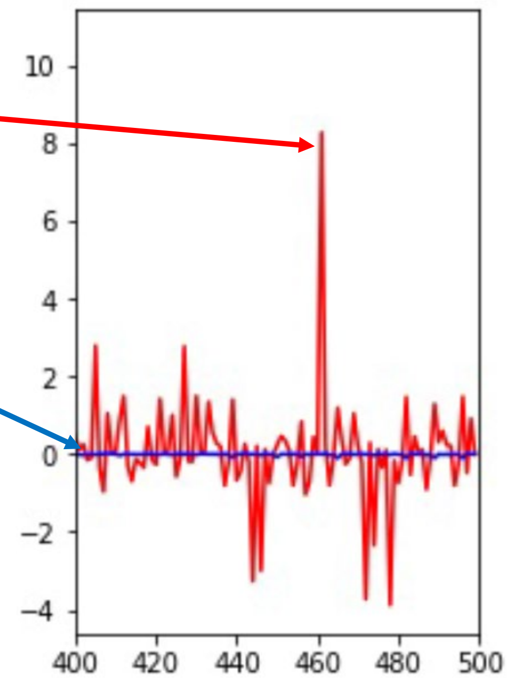
True dynamic　　Predicted dynamic

Learned kernel

Prediction error

Initial kernel

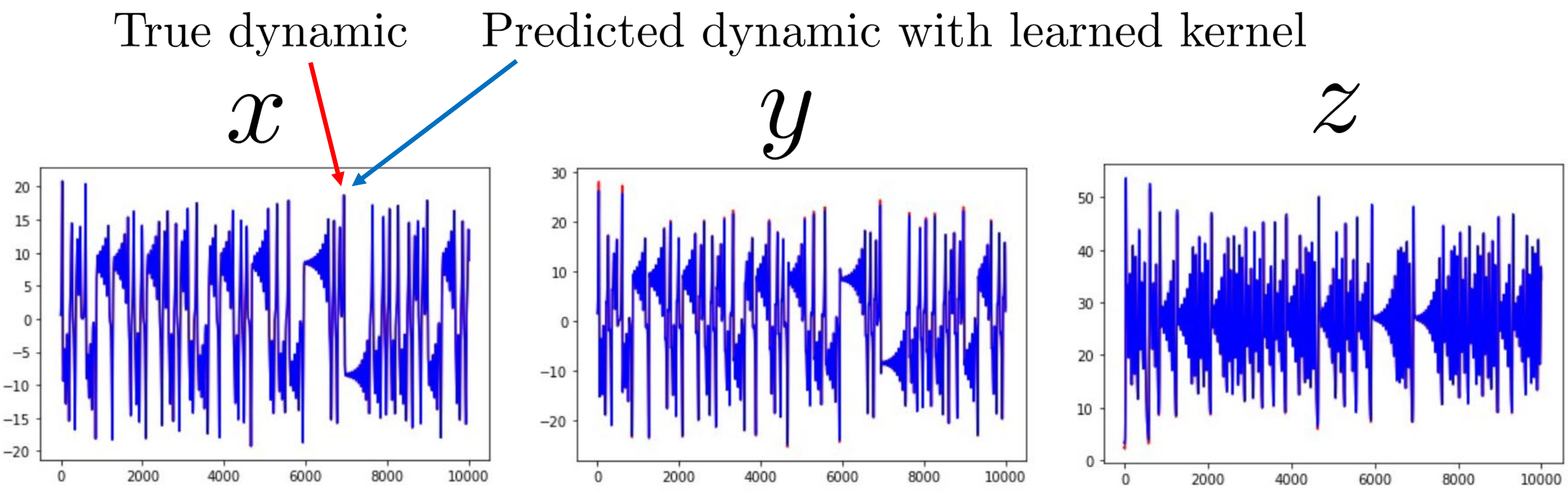
Learned kernel



Example: Lorenz system

$$\frac{dx}{dt} = s(y-x)$$
$$\frac{dy}{dt} = rx - y - xz$$
$$\frac{dz}{dt} = xy - bz$$

$$k_i(x,y) = \alpha_i + (\beta_i + ||x-y||_2^{\kappa_i})^{\sigma_i} + \delta_i e^{-||x-y||_2^2/\mu_i^2}$$



Data-driven geophysical forecasting

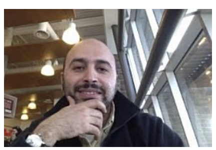
HYCOM: 800 core-hours per day of forecast on a Cray XC40 system

CESM: 17 million core-hours on Yellowstone, NCAR's high-performance computing resource

Architecture optimized LSTM: 3 hours of wall time on 128 compute nodes of the Theta supercomputer.

Our method: 40 seconds to train on a single node machine (laptop) without acceleration

Romit Maulik (ANL)



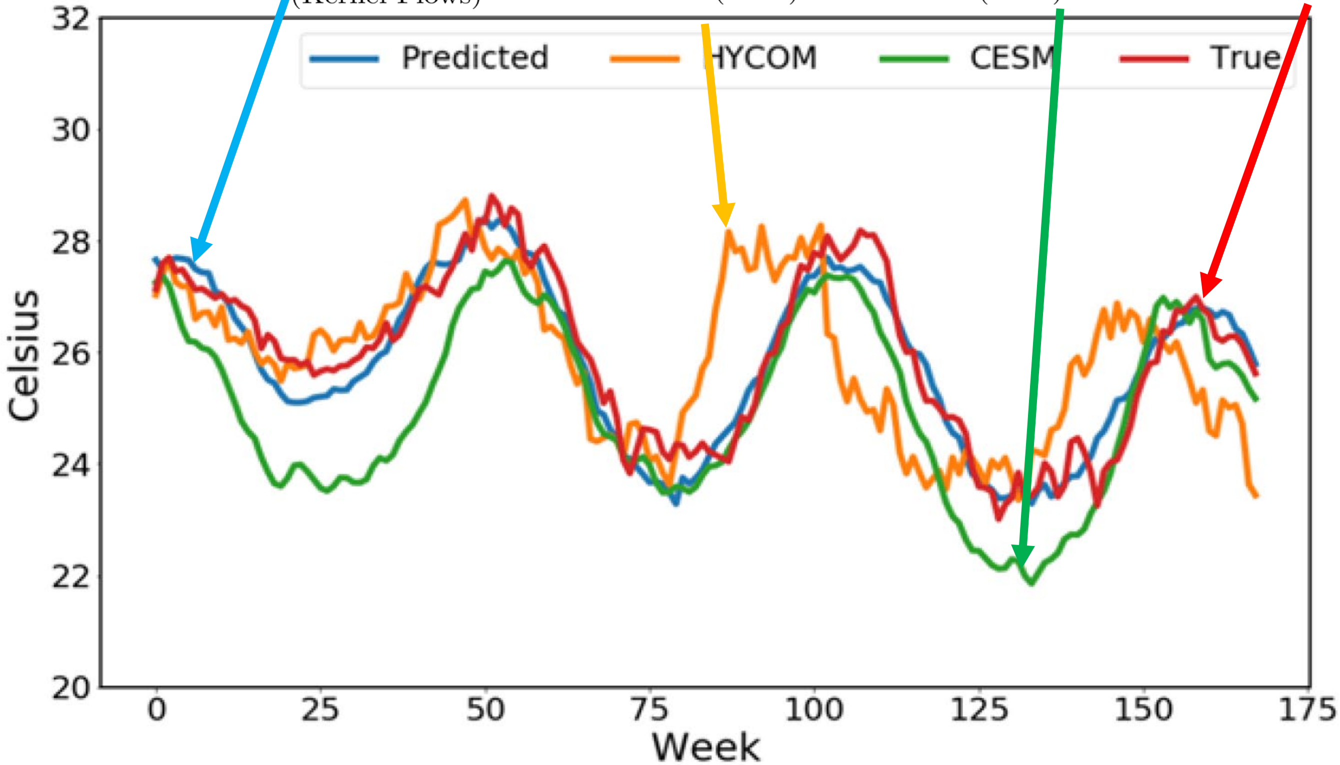
Boumediene Hamzi Imperial/Caltech

Predicted (Kernel Flows)

HYCOM (PDE)

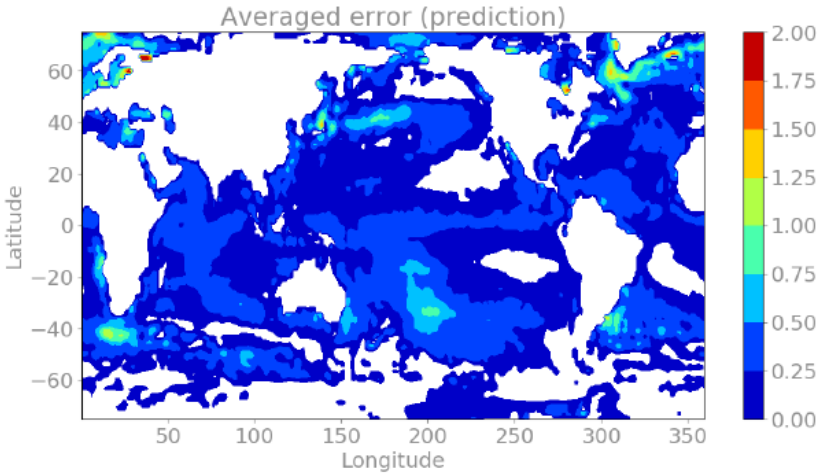
CESM (PDE)

True (NOAA)

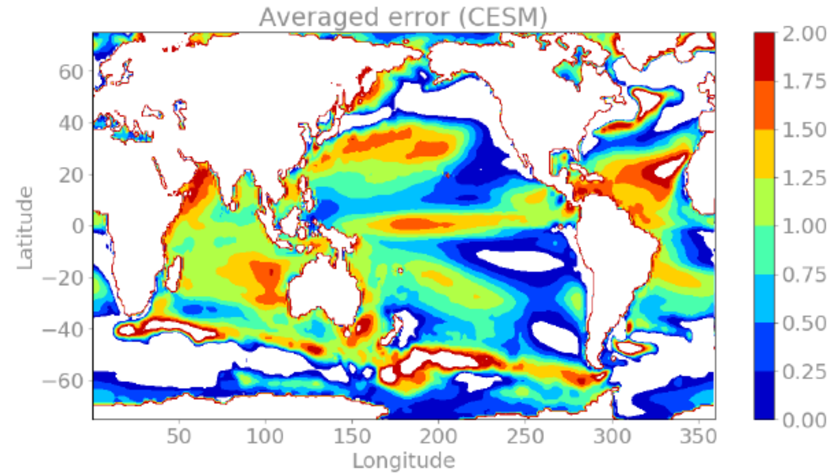


Data-driven geophysical forecasting: Simple, low-cost, and accurate baselines with kernel methods. **Proceedings of the Royal Society A**. 2021 Hamzi, Maulik, O.

NOAA-SST data set (low noise dataset)



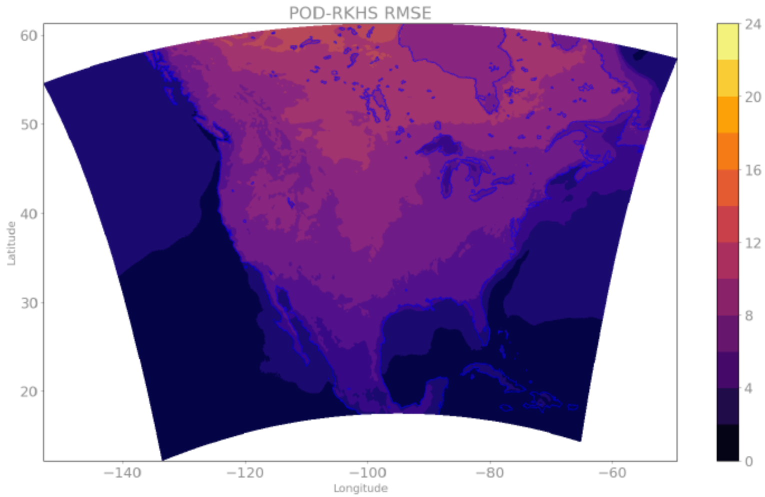
(a) Prediction error



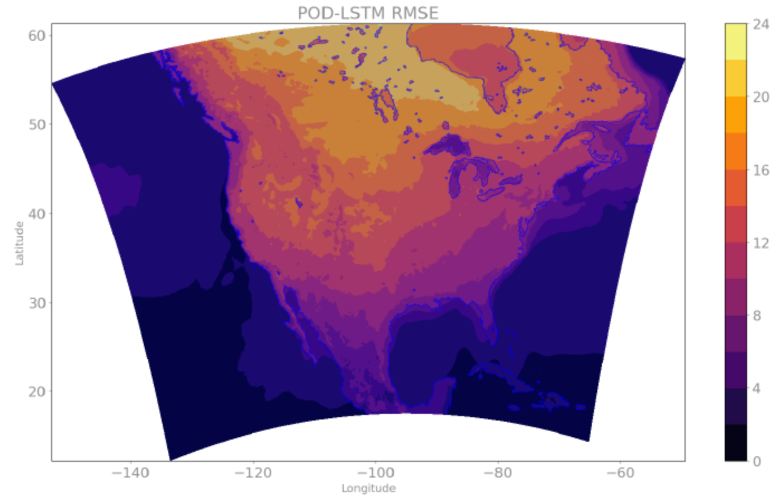
(b) CESM error

	RMSE (°Celsius)							
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
NAS-LSTM	0.62	0.63	0.64	0.66	0.63	0.66	0.69	0.65
CESM	1.88	1.87	1.83	1.85	1.86	1.87	1.86	1.83
HYCOM	0.99	0.99	1.03	1.04	1.02	1.05	1.03	1.05
Predicted	0.76	0.67	0.66	0.69	0.69	0.72	0.77	0.76

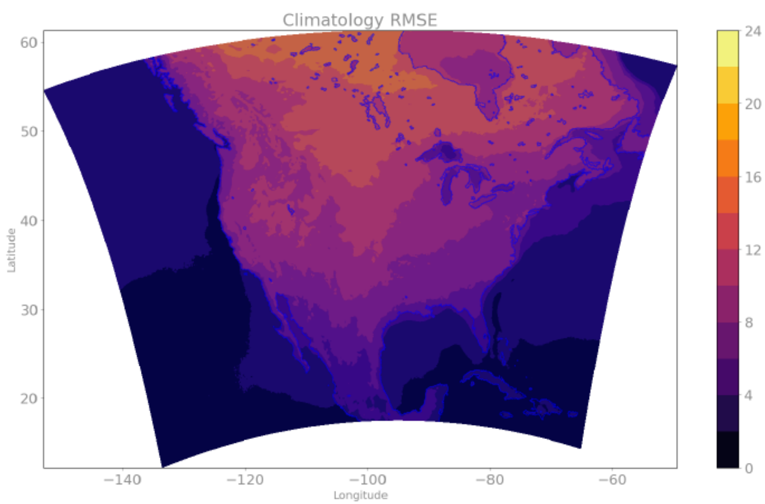
NAM (North American Mesoscale Forecast System) dataset (high noise dataset)



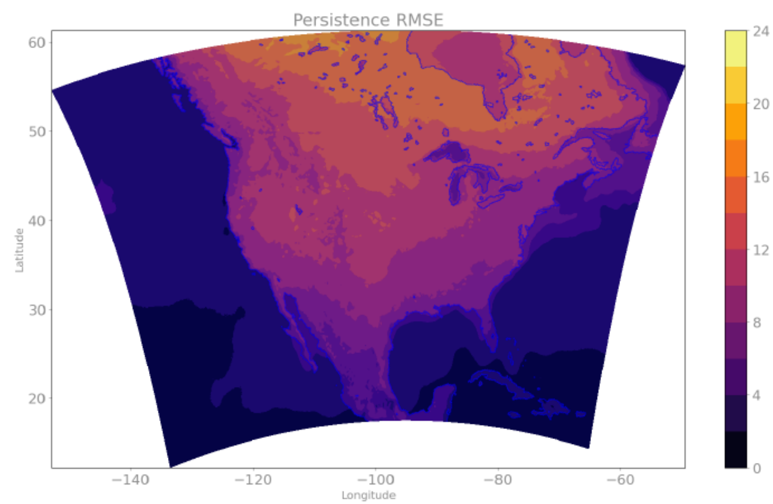
(a) POD-RKHS Prediction



(b) POD-LSTM



(c) Climatology



(d) Persistence

Thank you