Active Learning for Scientific Computing

Nick Dexter[†]

[†]Assistant Professor, Department of Scientific Computing, Florida State University, Tallahassee, FL, USA

FLORIDA STATE
UNIVERSITY

FSU DSC ML Seminar - DSC 499
Friday March 29th, 2024 - Tallahassee, FL

Collaborators on works referenced in this talk

Simon Fraser University:

- Ben Adcock
- Sebastian Moraga (PhD student)

University of Colorado Boulder:

- Juan Cardenas (postdoc)

Concordia University:

- Simone Brugiapaglia

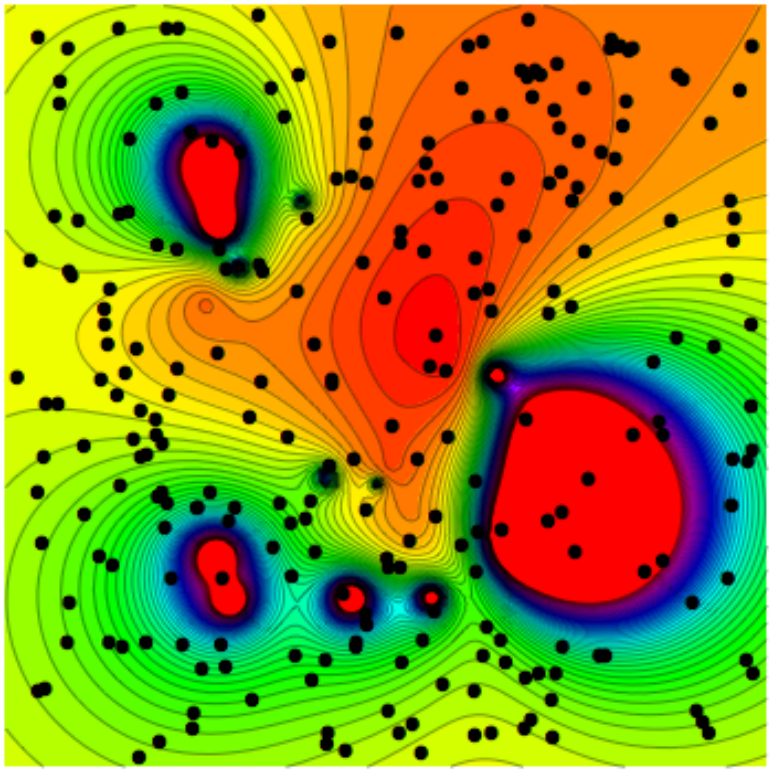
Active learning in regression

Standard active learning setup:

- $\mathcal{U} \subseteq \mathbb{R}^d$ be a domain,
- ρ be a measure on $\mathcal{U}$,
- $\mathbb{U} \subset L^2_\rho(\mathcal{U})$ an approximation space (or model class, hypothesis set).

Goal: Approximate $f : \mathcal{U} \to \mathcal{V}$ with an element $\hat{f} \in \mathbb{U}$ from m (noisy) samples

$$d_i = f(\boldsymbol{y}_i) + n_i, \quad i = 1, \ldots, m.$$



Question:

Assuming we can sample f (the oracle) at any point in $\mathcal{U}$, how should we best choose the sample points $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$?

Machine learning in computational science

Recently, much focus on applying Machine Learning (ML), in particular, Deep Learning (DL), techniques to problems in CSE.

Examples:

- DL for PDEs via Physics-Informed Neural Networks (PINNs)
- DL for parametric PDEs in UQ and operator learning
- ML for computational imaging
- ML for discovering dynamics of physical systems
- ....

Many of these applications are highly data starved and therefore stand to benefit from active learning.

The need for generalizations

However, in many such applications the training data is not in the form of pointwise samples. Indeed, we often encounter

- Transform data: E.g. Fourier or Radon transform.
- Multimodal data: $C > 1$ different types of data.
- Vector-valued data: Each sample is a vector or function.
- Plus combinations thereof.

Example: gradient-augmented regression

Standard setup: In gradient-augmented regression the training data is

$$(x_i, f(x_i), \nabla f(x_i)), \quad i = 1, \ldots, m,$$

(i.e., vector-valued data)

Applications: parametric PDEs in UQ, seismology, PINNs for PDEs, Sobolev training in DL,...

Partial gradient samples: It may be too costly to acquire full gradient samples. Instead, if $m = m_1 + m_2$, the training data is

$$(x_i, f(x_i), \nabla f(x_i)), \ i = 1, \ldots, m_1, \qquad (x_i, f(x_i)), \ i = 1, \ldots, m_2.$$

(i.e., multimodal data)

Example: MRI

Standard setup: Let $n \in \mathbb{N}$ and $X \in \mathbb{C}^{n \times \cdots \times n}$ be a discrete d-dimensional image with Fourier transform $\widehat{X}$.

Single-coil MRI: The training data is

$$(\omega_i, \widehat{X}(\omega_i)), \quad i = 1, \ldots, m,$$

for frequencies $\omega_1, \ldots, \omega_m$. (i.e., transform data)

Parallel MRI: $C > 1$ coils simultaneously acquire data. The training data is

$$(\omega_i, \widehat{X \odot S_c}(\omega_i)), \quad i = 1, \ldots, m_c, \ c = 1, \ldots, C,$$

where $S_c \in \mathbb{C}^{n \times \cdots \times n}$ is the sensitivity profile. (i.e., multimodal data)

Sampling: In MRI, it is impractical to sample individual frequencies. Samples are acquired along piecewise smooth curves in frequency space. (i.e., vector-valued data)

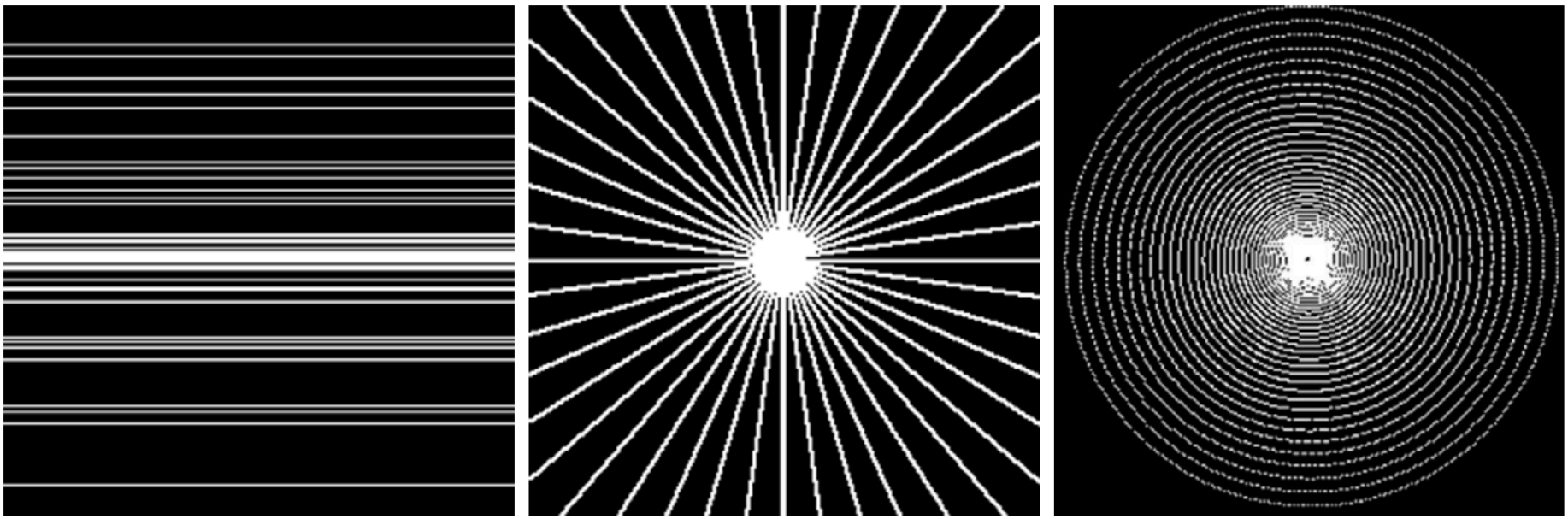


Figure 4.2 (a) Horizontal Cartesian sampling with lines chosen randomly from the distribution $\pi_\omega \propto 1/\max\{1, |\omega|\}$. (b) Radial sampling. (c) Spiral sampling.

Adcock & Hansen, Compressive Imaging: Structure, Sampling, Learning, (2021)

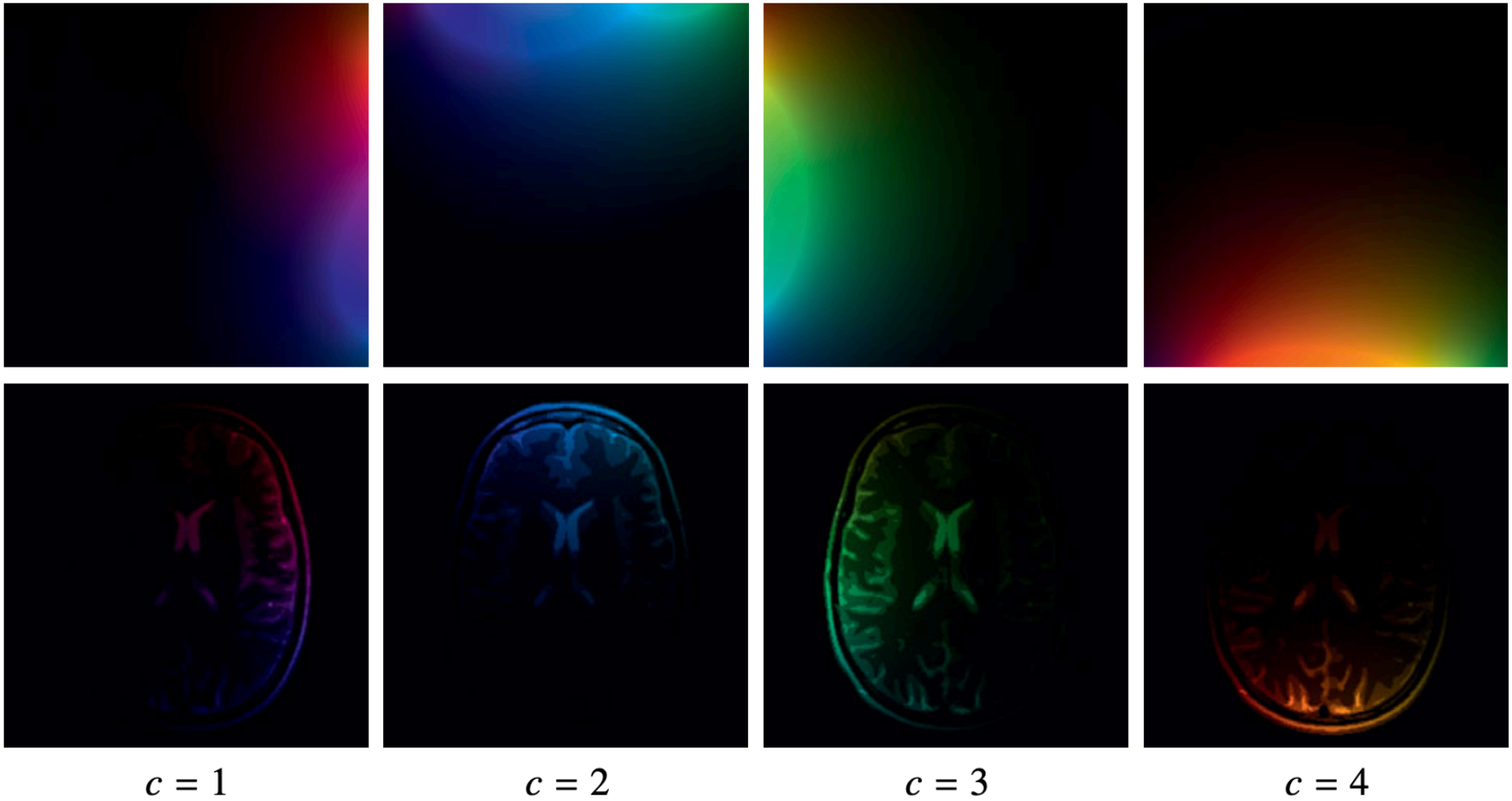


Figure 3.13 Complex sensitivity profiles $s^{(c)}$ (top), where $S^{(c)} = \text{diag}(s^{(c)})$, and coil images $x^{(c)} = S^{(c)}x$ (bottom) for the GLPU phantom. The phase is colour coded.

Adcock & Hansen, Compressive Imaging: Structure, Sampling, Learning, (2021)

Other examples

Continuous-in-time sampling: (e.g., seismic imaging) Here $f : D \times [0, T] \to \mathbb{R}$ and the training data is

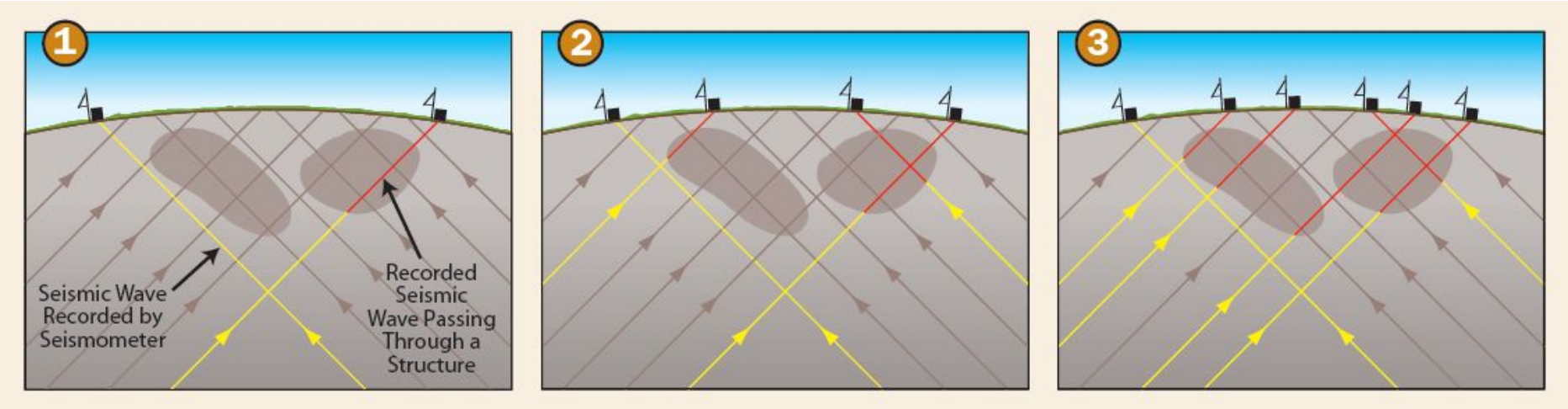
$$(x_i, f(x_i, t)), \quad i = 1, \ldots, m, \ t \in [0, T].$$

(i.e., vector-valued data)

Computed Tomography: Radon transform (i.e., transform data)

Parallel acquisition systems: Multi-view imaging, wireless sensor networks, synthetic aperture radar,... (i.e., multimodal data)

Plus many more...



Increasing the number of recording stations improves quality of the reconstruction.

`https://wiki.seg.org/wiki/Seismic_tomography`

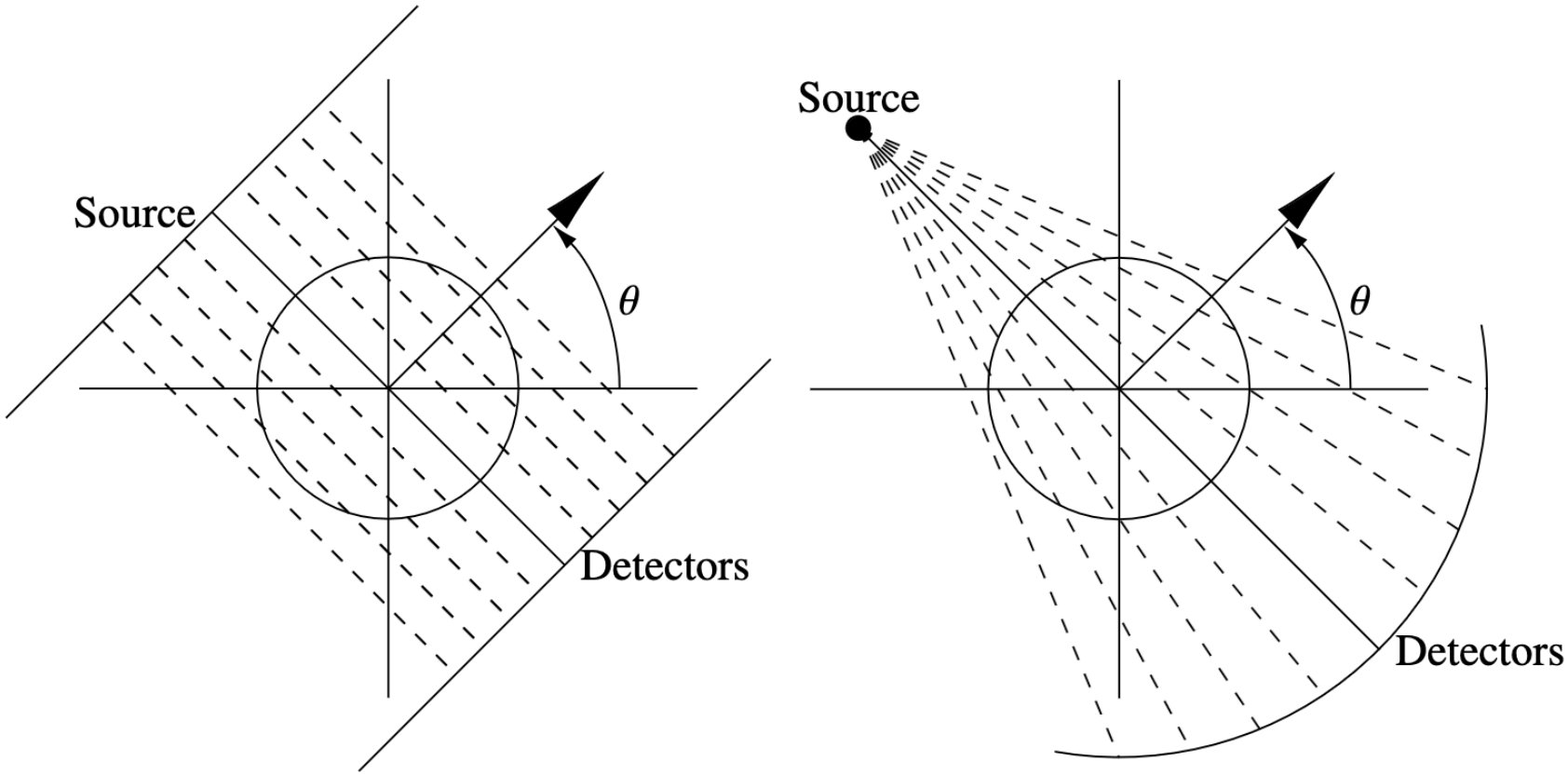
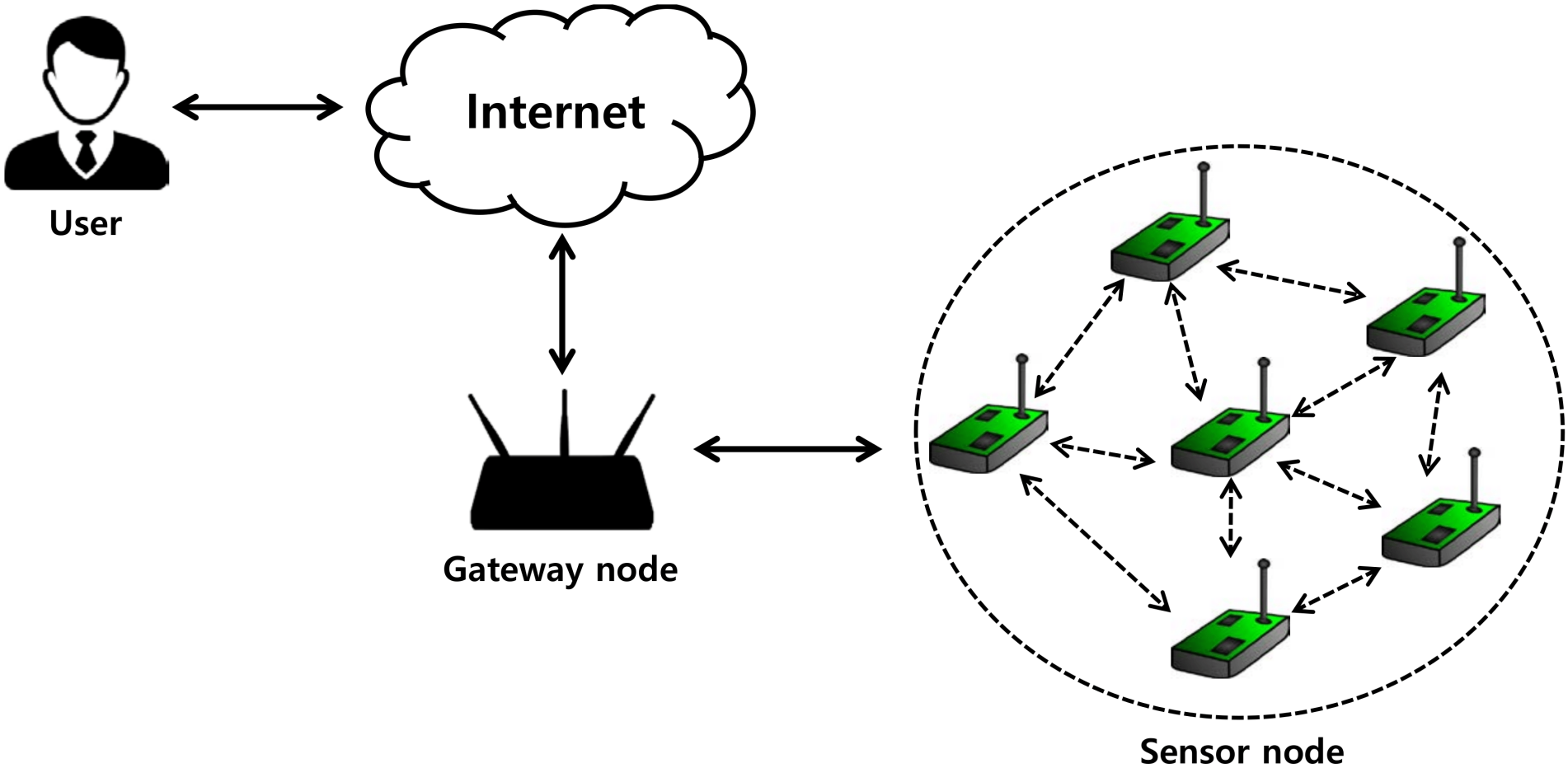


Figure 3.6 Typical geometries in X-ray CT. Left: Parallel beam. Right: Fan beam.

In contrast with MRI, there is generally far less scope to change the measurements acquired in X-ray CT. Measurements are largely dictated by the scanner geometry

Adcock & Hansen, Compressive Imaging: Structure, Sampling, Learning, (2021)



SLUA-WSN: Secure and Lightweight Three-Factor-Based User Authentication Protocol for Wireless Sensor Networks, by SungJin Yu and YoungHo Park

Banach-valued problems in UQ

Banach-valued data: In UQ, f may take values in a Banach space $\mathcal{V}$ (e.g., the solution map of a parametric PDE). (i.e., Banach-valued data)

Input $\boldsymbol{y} \in \mathbb{R}^d$ → **Blackbox model** → Output $u(\cdot, \boldsymbol{y}) \in \mathcal{V}$ → Quantity of Interest (QoI) $\mathcal{Q}(u(\cdot, \boldsymbol{y})) \in \mathbb{R}$

PDE example: Approximating solutions to a parameterized PDE of the form

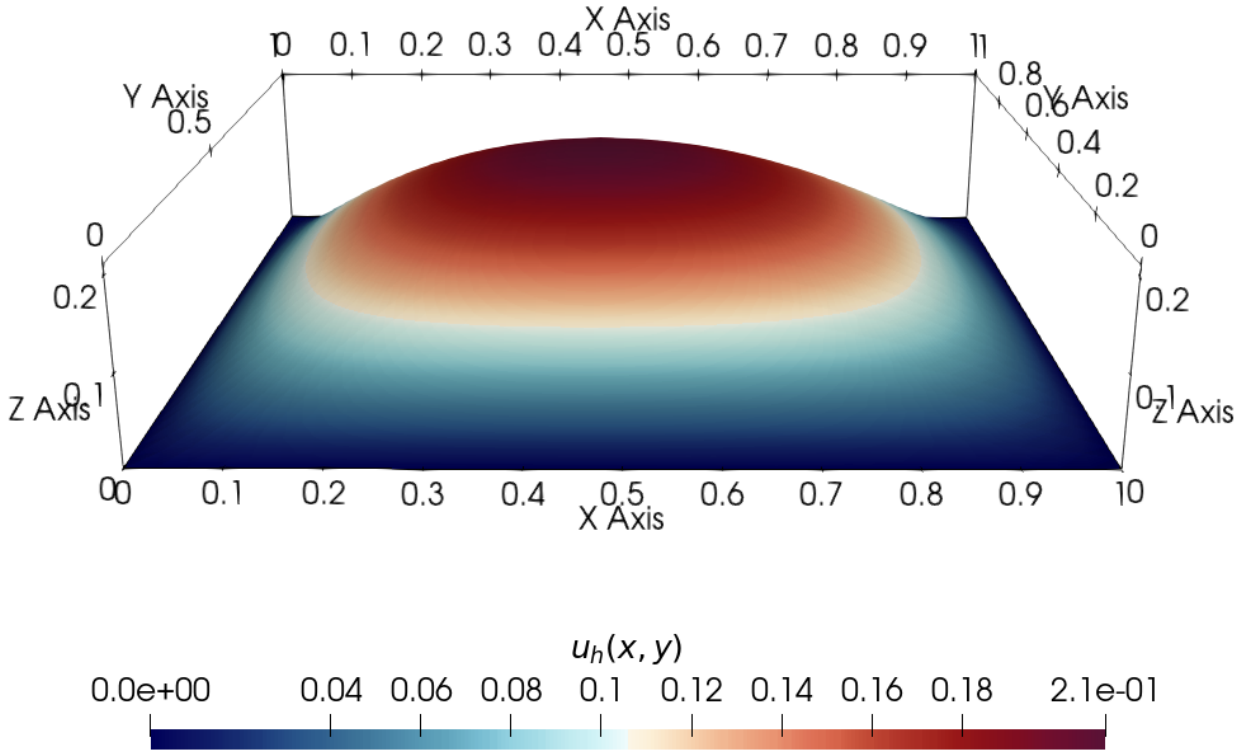
$$\mathcal{L}_{\boldsymbol{x}}\, u(\boldsymbol{x}, \boldsymbol{y}) = 0,$$

subject to suitable boundary conditions, from sample data $\{(\boldsymbol{y}_i, d_i)\}_{i=1}^m$, where

- $\boldsymbol{y} \in \mathcal{U} \subset \mathbb{R}^d$ is the parameter variable,
- $\boldsymbol{x} \in \Omega \subset \mathbb{R}^n$ is the spatial (temporal) variable,
- $\mathcal{L}_{\boldsymbol{x}}$ is a differential operator in $\boldsymbol{x}$,
- $\mathcal{V}$ is the PDE solution space, generally a Hilbert or Banach space.

Goal: Cheaply approximate a QoI $g = \mathcal{Q}(u(\boldsymbol{x}, \boldsymbol{y}))$ (a function of $\boldsymbol{x}$ or $\boldsymbol{y}$), e.g.,

$$\text{(parametric)} \qquad g(\boldsymbol{y}) = \int_\Omega u(\boldsymbol{x}, \boldsymbol{y})\, \mathrm{d}\boldsymbol{x} \quad \text{(spatial mean)}$$



Model problem:

$$-\nabla \cdot (a(\boldsymbol{x}, \boldsymbol{y}) \nabla u(\boldsymbol{x}, \boldsymbol{y})) = g(\boldsymbol{x}), \ \forall \boldsymbol{x} \in \Omega, \ \boldsymbol{y} \in \mathcal{U}, \quad u(\boldsymbol{x}, \boldsymbol{y}) = 0, \ \forall \boldsymbol{x} \in \partial\Omega, \ \boldsymbol{y} \in \mathcal{U}.$$

$$a(\boldsymbol{x}, \boldsymbol{y}) = 3 + x_1 y_1 + x_2 y_2$$

This gives a family of partial differential equation problems for which, depending on the values of y_1 and y_2 the height of the above plot will change. Here we want to approximate the mapping $\boldsymbol{y} \mapsto u(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{V}$.

This talk

Goal: Develop a framework for active learning with theoretical guarantees for general types of multimodal data and nonlinear approximation spaces.

Specifically: We extend leverage score (or coherence-motivated) sampling to general data and nonlinear approximation spaces.

Hampton & Doostan (2015), Cohen & Migliorati (2017), Woodruff (2014), Erdelyi, Musco & Musco (2020).

Let $\mathcal{U}$ be a domain with a measure λ (typically the Lebesgue measure), $\mathbb{V}$ be a class of functions $f : \mathcal{U} \to \mathbb{C}$ and ρ be a probability measure over $\mathcal{U}$ with Radon-Nikodym derivative $p = \mathrm{d}\rho/\,\mathrm{d}\lambda$. Then leverage score is defined as

$$\tau(\boldsymbol{y}) = \tau(\mathbb{V}, p)(\boldsymbol{y}) = \sup\left\{ \frac{|f(\boldsymbol{y})|^2 p(\boldsymbol{y})}{\|f\|_p^2} : f \in \mathbb{V} \setminus \{0\} \right\}$$

where $\|f\|_p^2 = \int_{\mathcal{U}} |f(\boldsymbol{y})|^2 p(\boldsymbol{y})\,\mathrm{d}\lambda(\boldsymbol{y}) = \int_{\mathcal{U}} |f(\boldsymbol{y})|^2\,\mathrm{d}\rho(\boldsymbol{y})$.

Leverage score sampling is a type of importance sampling where one draws samples randomly according to the probability density proportional to τ (or, often in practice, some easy-to- compute upper bound $\tilde{\tau}$).

Setup:

- separable Hilbert space $\mathbb{X}$, & normed vector subspace $\mathbb{X}_0$, termed the object space,
- an unknown object $f \in \mathbb{X}_0$
- a known approximation space $\mathbb{U}$,
- $C \geq 1$ measurement processes and, for each $c = 1, \ldots, C$,
- a measure space $(\mathcal{U}_c, \mathcal{D}_c, \rho_c)$, termed the measurement domain,
- a separable Hilbert space $\mathbb{Y}_c$, termed the measurement space,
- a mapping $L_c : \mathcal{U}_c \to \mathcal{B}(\mathbb{X}_0, \mathbb{Y}_c)$, termed the sampling operator.

Example (classical regression problem): $\mathbb{X} = L^2_\rho(\mathcal{U})$, $\mathbb{X}_0 = C(\overline{\mathcal{U}})$, $C = 1$, $\mathcal{U}_1 = \mathcal{U}$, $\rho_1 = \rho$, $\mathbb{Y}_1 = \mathbb{R}$ and

$$L(\boldsymbol{y}) : f \in C(\overline{\mathcal{U}}) \mapsto f(\boldsymbol{y}) \in \mathbb{R}, \qquad \forall \boldsymbol{y} \in \mathcal{U}.$$

Assumptions

Nondegeneracy. For each c, $\boldsymbol{y}_c \in \mathcal{U}_c \mapsto L_c(\boldsymbol{y}_c)(x) \in \mathbb{Y}_c$ is measurable for every $x \in \mathbb{X}_0$ and $\boldsymbol{y}_c \in \mathcal{U}_c \mapsto \|L_c(\boldsymbol{y}_c)(x)\|_{\mathbb{Y}_c}^2 \in \mathbb{R}$ is integrable. Moreover, the sampling operators preserve the $\mathbb{X}$-norm up to constants:

$$\alpha\|x\|_{\mathbb{X}}^2 \leq \sum_{c=1}^{C} \int_{\mathcal{U}_c} \|L_c(\boldsymbol{y}_c)(x)\|_{\mathbb{Y}_c}^2 \, \mathrm{d}\rho_c(\boldsymbol{y}_c) \leq \beta\|x\|_{\mathbb{X}}^2, \quad \forall x \in \mathbb{X}_0$$

Active learning. For each c, we can evaluate $L(\boldsymbol{y}_c)(f)$ for any $\boldsymbol{y}_c \in \mathcal{U}_c$.

Sampling measures: For each c, we consider a probability measure μ_c on $(\mathcal{U}_c, \mathcal{D}_c)$ with Radon–Nikodym derivative $\nu_c = \frac{d\mu_c}{d\rho_c}$ that is > 0 a.e.

Training data: For each c, draw $\boldsymbol{y}_{ic}$, $i = 1, \dots, m_c$, i.i.d. from μ_c.
Consider the training data:

$$(\boldsymbol{y}_{ic}, d_{ic} := L_c(\boldsymbol{y}_{ic})(f) + n_{ic}) \in \mathcal{U}_c \times \mathbb{Y}_c, \quad i = 1, \dots, m_c, \ c = 1, \dots, C.$$

Learning: We consider the empirical least-squares fit

$$\hat{f} \in \arg\min_{u \in \mathbb{U}} \sum_{c=1}^{C} \frac{1}{m_c} \sum_{i=1}^{m_c} \frac{1}{\nu_c(\boldsymbol{y}_{ic})} \|d_{ic} - L_c(\boldsymbol{y}_{ic})(u)\|_{\mathbb{Y}_c}^2.$$

Aim: Choose μ_c (equivalently ν_c) to ensure good generalization using as few measurements m_c as possible.

Empirical nondegeneracy implies good generalization

Empirical NonDegeneracy (EmpND) holds over a set $\mathbb{V} \subseteq \mathbb{X}_0$ for a collection $\{\boldsymbol{y}_{ic}\}$ if, for constants $\alpha' \approx \alpha$ and $\beta' \approx \beta$, we have

$$\alpha' \|v\|_{\mathbb{X}}^2 \leq \sum_{c=1}^{C} \frac{1}{m_c} \sum_{i=1}^{m_c} \|L_c(\boldsymbol{y}_{ic})(v)\|_{\mathbb{Y}_c}^2 \leq \beta' \|v\|_{\mathbb{X}}^2, \quad \forall v \in \mathbb{V}.$$

Informal statement

Let E be the event that (EmpND) holds over $\mathbb{U} - \mathbb{U}$ for a draw $\{\boldsymbol{y}_{ic}\}$. Suppose that $\|f\|_{\mathbb{X}} \leq 1$ and let $\check{f} = \min\{1, 1/\|f\|_{\mathbb{X}}\}\hat{f}$. Then

$$\mathbb{E}\|f - \check{f}\|_{\mathbb{X}}^2 \lesssim (\beta'/\alpha') \cdot \inf_{u \in \mathbb{U}} \|f - u\|_{\mathbb{X}}^2 + \mathbb{P}(E^c) + \text{noise}.$$

Goal: Find conditions on ν_c for (EmpND) to hold with high probability.

The generalized Christoffel function

Definition

Let $(\mathcal{U}, \mathcal{D}, \rho)$ be a measure space, $\mathbb{Y}$ be a Hilbert space and the sampling operator $L : \mathcal{U} \to \mathcal{B}(\mathbb{X}_0, \mathbb{Y})$ be such that $\boldsymbol{y} \in \mathcal{U} \mapsto L(\boldsymbol{y})(x) \in \mathbb{Y}$ is measurable for every $x \in \mathbb{X}_0$ and $\mathbb{V} \subseteq \mathbb{X}_0$, $\mathbb{V} \neq \{0\}$. The generalized Christoffel function of $\mathbb{V}$ with respect to L is

$$K(\mathbb{V})(\boldsymbol{y}) = \sup \left\{ \|L(\boldsymbol{y})(v)\|_{\mathbb{Y}}^2 / \|v\|_{\mathbb{X}}^2 : v \in \mathbb{V},\ v \neq 0 \right\}, \boldsymbol{y} \in \mathcal{U}.$$

Example (classical regression): Recall that $L(\boldsymbol{y})(f) = f(\boldsymbol{y})$. Hence

$$K(\mathbb{V})(\boldsymbol{y}) = \sup \left\{ |v(\boldsymbol{y})|^2 / \|v\|_{L^2_\rho(\mathcal{U})}^2 : v \in \mathbb{V},\ v \neq 0 \right\}, \quad \boldsymbol{y} \in \mathcal{U}.$$

Known as the (reciprocal) Christoffel function or leverage score of $\mathbb{V}$.

Main result I: linear spaces

Theorem

Let $0 < \epsilon < 1$ and $\mathbb{U} \subseteq \mathbb{X}_0$ be a subspace with $\dim(\mathbb{U}) = n$. Suppose that, for $c = 1, \ldots, C$,

$$m_c \gtrsim \alpha^{-1} \cdot \operatorname{ess\,sup} \{K_c(\mathbb{U})(\boldsymbol{y}_c)/\nu_c(\boldsymbol{y}_c)\} \cdot \log(2n/\epsilon).$$

Then (EmpND) holds over $\mathbb{U}$ with probability at least $1 - \epsilon$.

Active learning strategy: Choose $\nu_c \propto K_c(\mathbb{U})$. Yields the near-optimal, log-linear sample complexity bound

$$m_c \gtrsim \alpha^{-1} \cdot n \cdot \log(2n/\epsilon), \quad c = 1, \ldots, C.$$

Note: This extends so-called leverage score (or coherence-motivated) sampling for regression to arbitrary linear sampling operators.

Hampton & Doostan (2015), Cohen & Migliorati (2017), Woodruff (2014), Erdelyi, Musco & Musco (2020).

Main result II: nonlinear spaces

Assumption: $\mathbb{U}' := \mathbb{U} - \mathbb{U}$ is a cone and $\mathbb{U}' \subseteq \mathbb{V}_1 \cup \cdots \cup \mathbb{V}_d$, where each $\mathbb{V}_i$ is a subspace of $\mathbb{X}_0$ with $\dim(\mathbb{V}_i) \leq n$.

Theorem (Slightly weakened, Adcock, Cardenas & ND (2023))

Suppose that, for $c = 1, \ldots, C$,

$$m_c \gtrsim \alpha^{-1} \cdot \operatorname{ess\,sup} \left\{ K_c(\mathbb{U}')(\boldsymbol{y}_c)/\nu_c(\boldsymbol{y}_c) \right\} \cdot (\log(2d/\epsilon) + n),$$

then (EmpND) holds over $\mathbb{U}$ with probability at least $1 - \epsilon$.

Active learning strategy: Choose $\nu_c \propto K_c(\mathbb{U}')$. Yields the sample complexity bound

$$m_c \gtrsim \alpha^{-1} \cdot \kappa_c(\mathbb{U}') \cdot (\log(2d/\epsilon) + n)$$

where $\kappa_c(\mathbb{U}') = \int_{\mathcal{U}_c} K_c(\mathbb{U}')(\boldsymbol{y}_c) \, \mathrm{d}\rho_c(\boldsymbol{y}_c)$.

Remarks

Key point: In all cases, the strategy is to sample according to $K_c(\mathbb{U}')$.

Linear spaces: The strategy is near optimal.

Nonlinear spaces: The results are involve generalizations and improvements of various compressed sensing results for specific problems.

Optimality? The bound depends on $\kappa_c(\mathbb{U}') = \int_{\mathcal{U}_c} K_c(\mathbb{U}')(\boldsymbol{y}_c)\,\mathrm{d}\rho_c(\boldsymbol{y}_c)$, which is problem dependent. Can be estimated numerically/theoretically.

Practical challenge: Find an estimator $\widetilde{K}_c(\mathbb{U}') \approx K_c(\mathbb{U}')$ from which it is computationally feasible to sample (problem dependent).

Example I: gradient-augmented regression in UQ

Setup: Consider the solution map $\boldsymbol{y} \in \mathbb{R}^d \mapsto u(\cdot, \boldsymbol{y}) \in H_0^1(\Omega)$ of a parametric elliptic diffusion equation with lognormal diffusion.

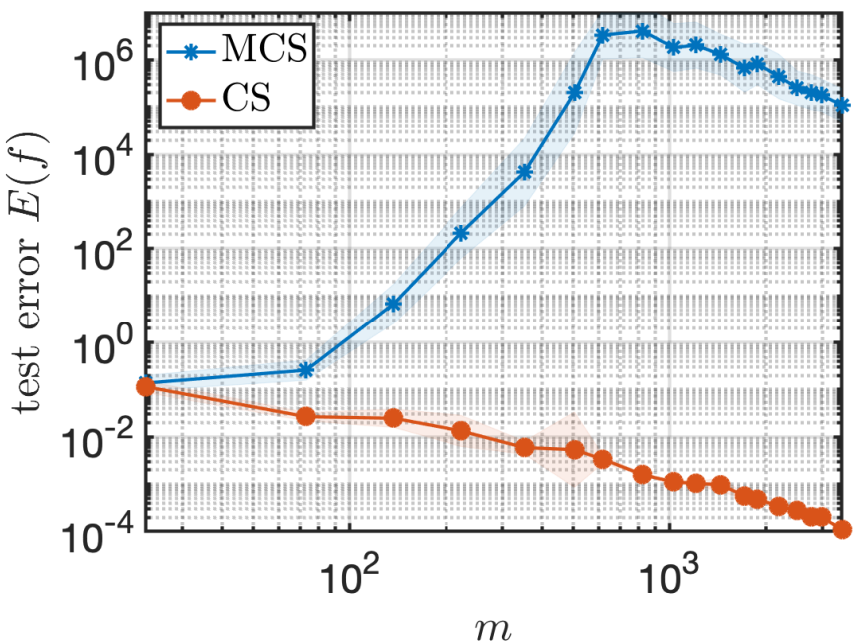
Training data: We consider gradient-augmented data

$$(\boldsymbol{y}_i, u(\cdot, \boldsymbol{y}_i), \nabla_{\boldsymbol{y}} u(\cdot, \boldsymbol{y}_i)) \in \mathbb{R}^d \times (H_0^1(\Omega))^{d+1}.$$

Approximation space (AS): A nested sequence of polynomial subspaces.

Comparison: We compare against Monte Carlo Sampling (MCS), where $\boldsymbol{y}_i \sim_{\text{i.i.d.}} \rho$ and ρ is the Gaussian measure on $\mathbb{R}^d$.

Theory: If n is the dimension of the subspace, then $m \gtrsim n \log(n)$ samples suffice.



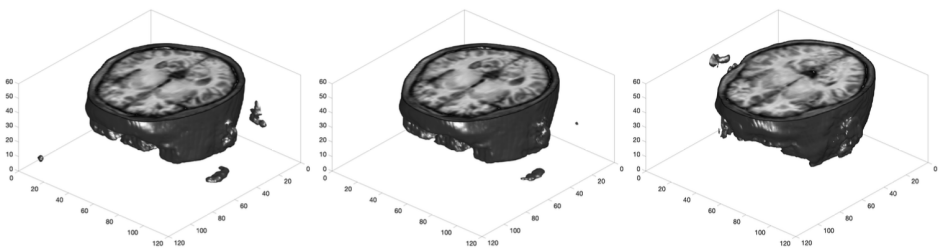
Example II: MRI reconstruction with generative models

Setup: 3D MRI acquisition using phase encoding (each sample is a horizontal line in k-space or frequency-space).

Approximation Space:
$\mathbb{U} = \{G(x) : x \in \mathbb{R}^k\} \equiv \mathrm{Ran}(G)$, where $G : \mathbb{R}^k \to \mathbb{R}^{n \times n \times n}$ is a generative model trained on 3D brain images.

Bora, Jalal, Price & Dimakis (2017), Jalal, Arvinte, Dara, Price, Dimakis & Tamir (2021)

Comparison: We compare against MCS (i.e., random sampling in 2D).

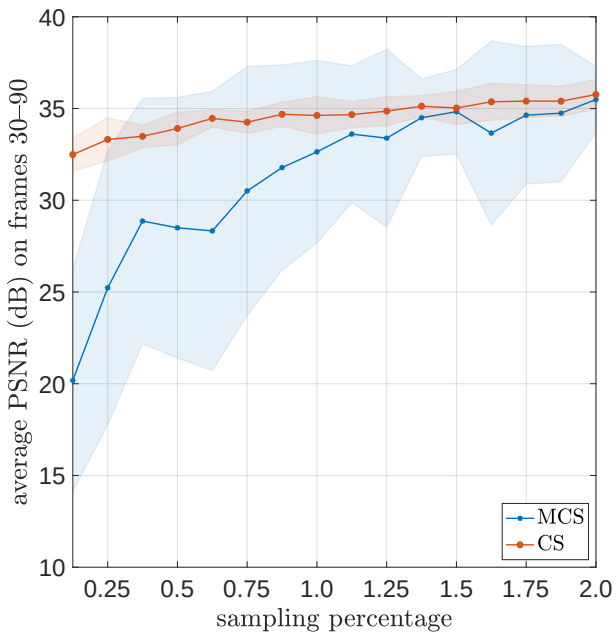


Comparing a trial of CS vs MCS at 0.125 sampling percentage. PSNR is 36.79 for CS and 27.87 for MCS.

Theory: For ReLU DNNs,

$$m \gtrsim \kappa \cdot k \cdot \log \text{ terms}$$

Empirically, $\kappa \approx 3.96$ for CS and $\kappa \approx 4616$ for MCS.



PSNR vs m

Algorithm 1 Computing $\widetilde{K}$

input: Generative model G, number of iterations t, input distribution ρ on the domain of G, DFT matrix F.
initialize: $\widetilde{K} = 0$
for $l = 1$ to t **do**
  Generate two i.i.d. realizations f_1, f_2 via the generative model, i.e., $f_i = G(z_i)$ for $z_i \sim_{\text{i.i.d.}} \rho$
  Compute $a_i = M \sum_{j \in \Delta_i} |(Fg)_j|^2 / \|v\|^2$ for $g = f_1 - f_2$ and $i = 1, \ldots, M$
  for $i = 1$ to M **do**
    Replace $\tilde{K}(i) = \max\{\tilde{K}(i), a_i\}$
  end for
end for
output: Vector $\widetilde{K}$ with updated entries.

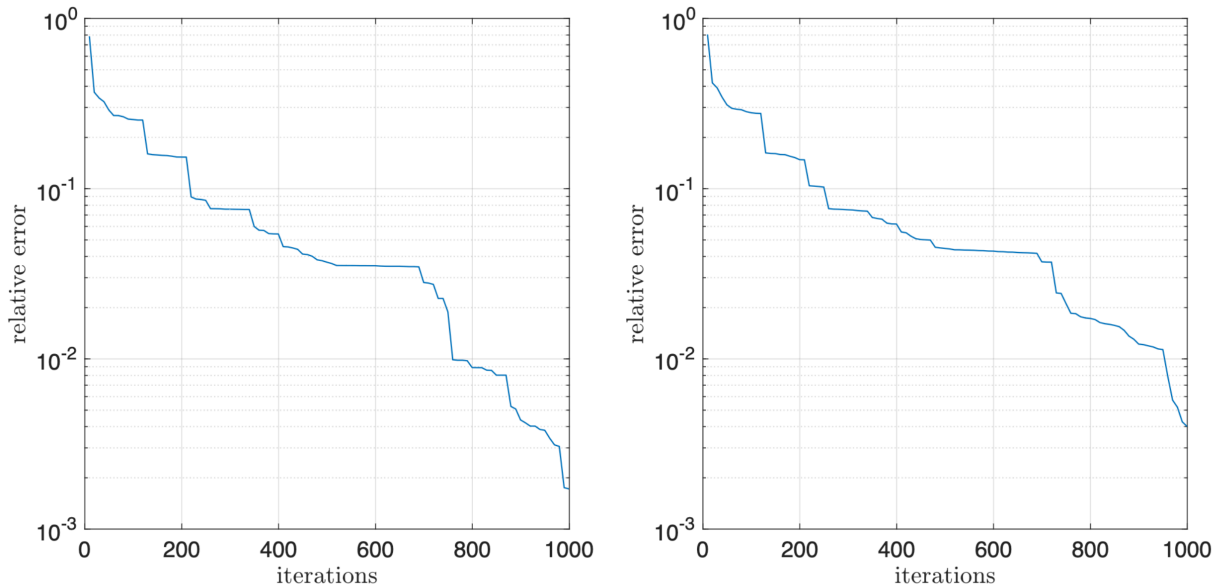
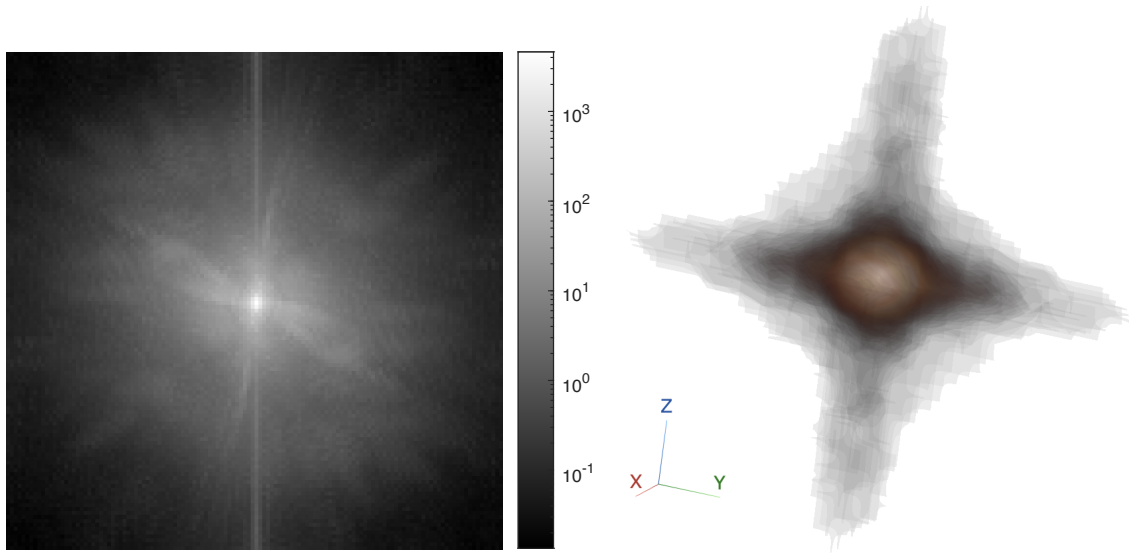


Figure 7: Convergence of the relative ℓ_2 error of iterations of Algorithm 1 in computing $\widetilde{K}$ for (left) Example C.2 and (right) Example C.1.

Cardenas, Adcock, ND, CS4ML: A general framework for active learning with arbitrary data based on Christoffel functions, (2023)

Algorithm 1 Computing $\widetilde{K}$

input: Generative model G, number of iterations t, input distribution ρ on the domain of G, DFT matrix F.
initialize: $\widetilde{K} = 0$
for $l = 1$ to t **do**
  Generate two i.i.d. realizations f_1, f_2 via the generative model, i.e., $f_i = G(z_i)$ for $z_i \sim_{\text{i.i.d.}} \rho$
  Compute $a_i = M \sum_{j \in \Delta_i} |(Fg)_j|^2 / \|v\|^2$ for $g = f_1 - f_2$ and $i = 1, \ldots, M$
  for $i = 1$ to M **do**
    Replace $\tilde{K}(i) = \max\{\tilde{K}(i), a_i\}$
  end for
end for
output: Vector $\widetilde{K}$ with updated entries.



Generated $\tilde{K}$ for (left) phase encoding and (right) 3d random sampling.

Cardenas, Adcock, ND, CS4ML: A general framework for active learning with arbitrary data based on Christoffel functions, (2023)

Example III: adaptive sampling for deep learning

Finally, suppose that $\mathbb{U}$ is a class of DNNs $u : \mathbb{R}^d \to \mathbb{R}$.

Question: How can we apply the ideas seen previously?

Assumption: The final layer is not subject to a nonlinear activation and has zero bias term (as is standard in regression problems).

Adaptive basis viewpoint: Let n be the width of the penultimate layer. Then

$$u = c_1 u_1 + \cdots + c_n u_n, \quad c_i \in \mathbb{R}$$

Here $u_i : \mathbb{R}^d \to \mathbb{R}$ is the DNN associated with the ith neuron. Therefore

$$u \in \mathbb{U}_u := \operatorname{span}\{u_1, \ldots, u_n\}. \qquad (\star)$$

Cyr, Gulian, Patel, Perego & Trask (2020)

Example III: adaptive sampling for deep learning

Finally, suppose that $\mathbb{U}$ is a class of DNNs $u : \mathbb{R}^d \to \mathbb{R}$.

Question: How can we apply the ideas seen previously?

Assumption: The final layer is not subject to a nonlinear activation and has zero bias term (as is standard in regression problems).

Adaptive basis viewpoint: Let n be the width of the penultimate layer. Then

$$u = c_1 u_1 + \cdots + c_n u_n, \quad c_i \in \mathbb{R}$$

Here $u_i : \mathbb{R}^d \to \mathbb{R}$ is the DNN associated with the ith neuron. Therefore

$$u \in \mathbb{U}_u := \operatorname{span}\{u_1, \ldots, u_n\}. \qquad (\star)$$

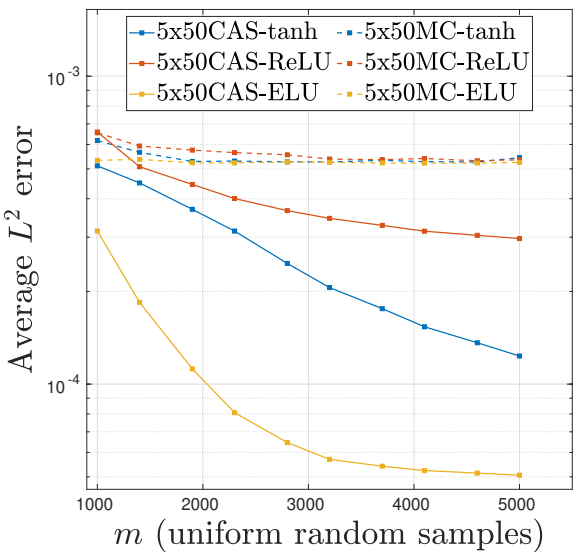
Cyr, Gulian, Patel, Perego & Trask (2020)

Example: adaptive sampling for deep learning

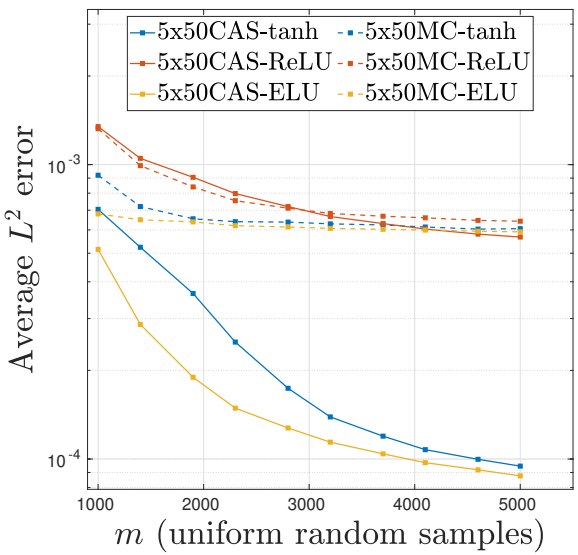
Adaptive sampling: Initialize a DNN $\hat{u}_0$ and set $m_0 = 0$.
For $l = 0, 1, \ldots$

- Construct the subspace $\mathbb{U}_l := \mathbb{U}_{\hat{u}_l}$ via $\star$.
- Generate $m_{l+1} - m_l$ samples via $K(\mathbb{U}_l)$.
- Train a new DNN $\hat{u}_{l+1}$ using the m_l samples with initialization $\hat{u}_l$.

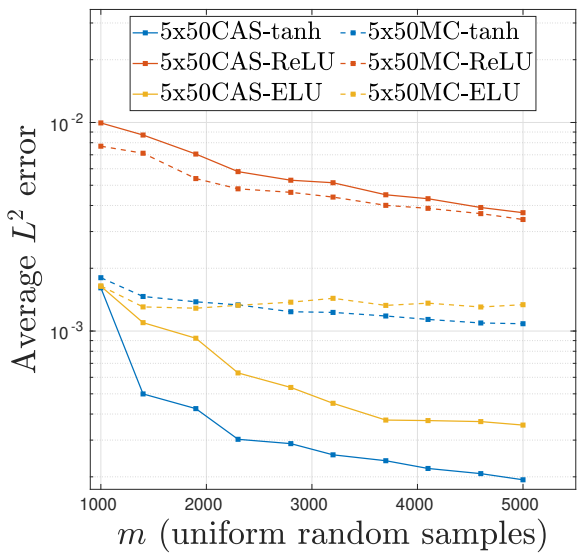
Example: Approximation of $f(y) = \exp(-(y_1 + \cdots + y_d)/d)$ using different DNN architectures.



$d = 1$

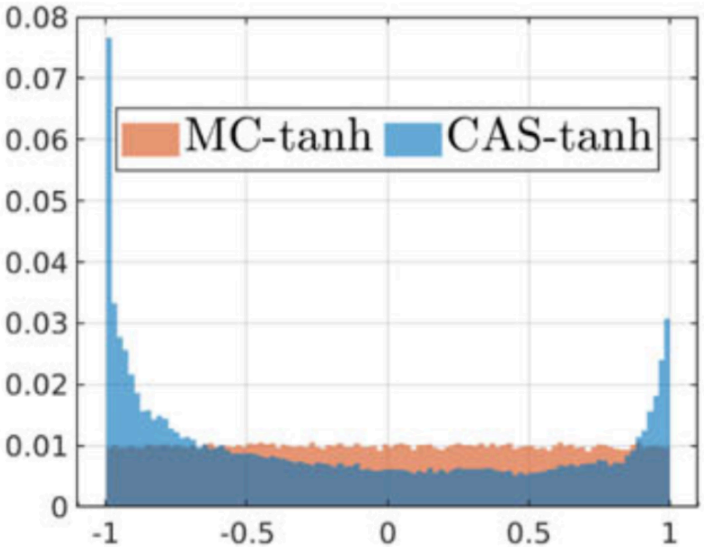


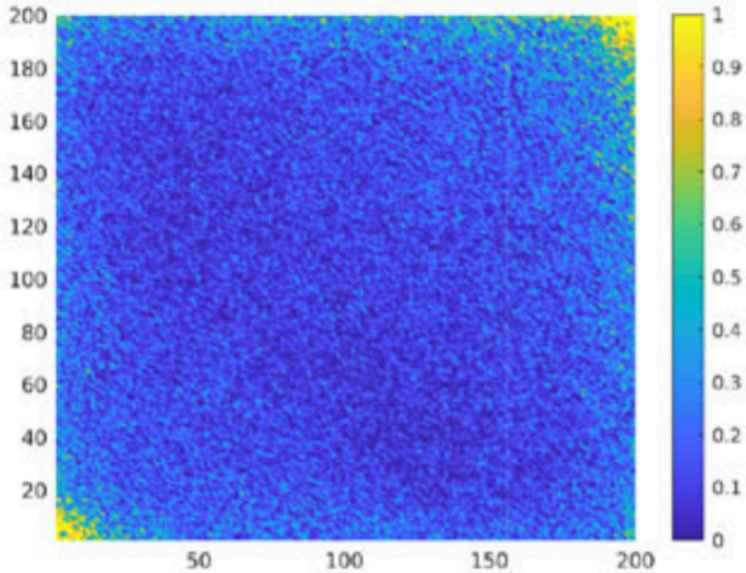
$d = 2$



$d = 8$

Adcock, Cardenas & ND (2023)





$(f, \rho) = (f_1, \tanh)$ $(f, \rho) = (f_1, \tanh)$

Learned sampling densities in (left) $d = 1$ and (right) $d = 2$ dimensions for the function $f(\boldsymbol{y}) = f_1(\boldsymbol{y}) = \exp(-(y_1 + \cdots + y_d)/d)$ and with $\tanh$ 5×50 DNNs.

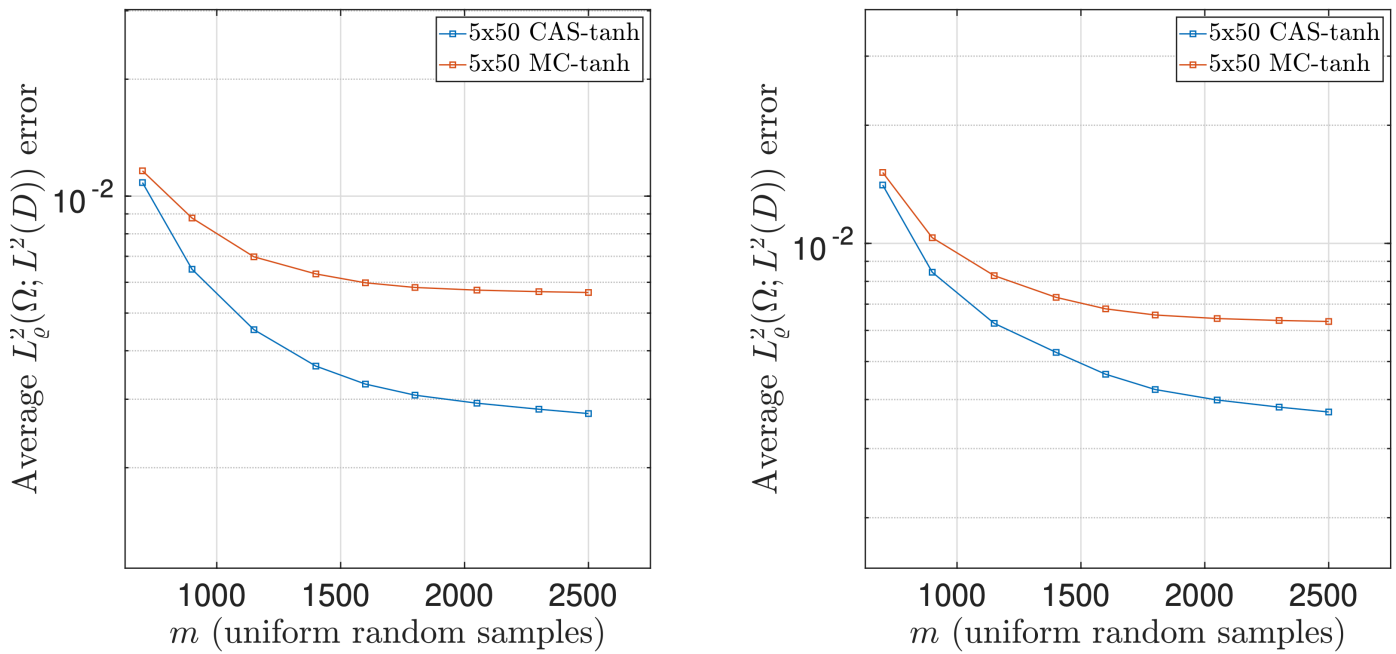
$$-\nabla \cdot (a(\boldsymbol{x},\boldsymbol{y})\nabla u(\boldsymbol{x},\boldsymbol{y})) = g(\boldsymbol{x}),\ \forall \boldsymbol{x} \in \Omega,\ \boldsymbol{y} \in \mathcal{U}, \quad u(\boldsymbol{x},\boldsymbol{y}) = 0,\ \forall \boldsymbol{x} \in \partial\Omega,\ \boldsymbol{y} \in \mathcal{U}.$$

$$a(\boldsymbol{x},\boldsymbol{y}) = \exp\left(1 + y_1 \left(\frac{\sqrt{\pi}\beta}{2}\right)^{1/2} + \sum_{i=2}^{d} \zeta_i\ \vartheta_i(\boldsymbol{x})\ y_i\right)$$

$$\zeta_i := (\sqrt{\pi}\beta)^{1/2} \exp\left(\frac{-\left(\lfloor \frac{i}{2} \rfloor\, \pi\beta\right)^2}{8}\right), \quad \vartheta_i(\boldsymbol{x}) := \begin{cases} \sin\left(\lfloor \frac{i}{2} \rfloor\, \pi x_1/\beta_p\right) & i \text{ even} \\ \cos\left(\lfloor \frac{i}{2} \rfloor\, \pi x_1/\beta_p\right) & i \text{ odd} \end{cases}$$

$$\beta_c = 1/8,\ \beta_p = \max\{1, 2\beta_c\},\ \beta = \beta_c/\beta_p.$$

Results: parametric elliptic PDE example, **(left)** $d = 8$ and **(right)** $d = 16$.



B. Adcock, J. Cardenas, ND, *CAS4DL: Christoffel Adaptive Sampling for function approximation via Deep Learning* Sampling Theory, Signal Processing, and Data Analysis (2022).

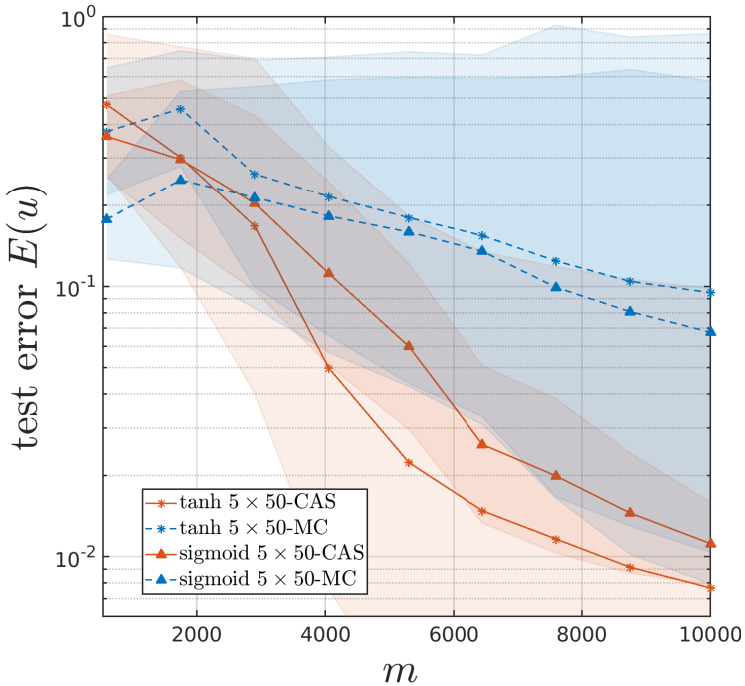
Example: adaptive sampling for deep learning

Example: Solution of the Burger's equation

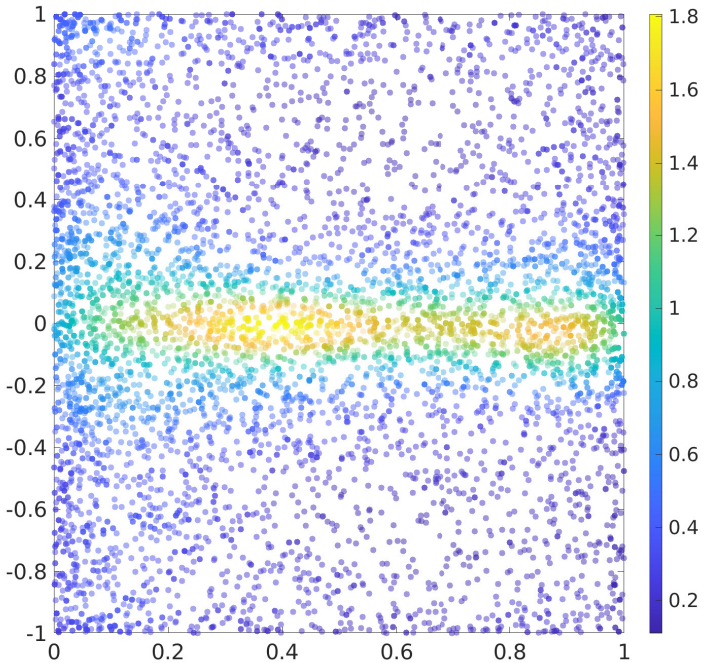
$$u_t + uu_x - (0.01/\pi)u_{xx} = 0, \quad -1 < x < 1,\ 0 < t \leq 1$$

$$u(x,0) = -\sin(\pi x),\ -1 < x < 1, \qquad u(\pm 1, t) = 0,\ 0 \leq t \leq 1.$$

using Physics-Informed Neural Networks (PINNs).



Error vs m



Clustering of the sample points

Adcock, Cardenas & ND (2023)

Conclusions

This talk: A general framework, CS4ML, for active learning in regression with arbitrary linear sampling operators.

Main idea: Sample according to the (generalized) Christoffel function.

Main results: Near-optimal for linear approximation spaces. Optimality for nonlinear spaces depends on the problem specific term $\kappa_c(\mathbb{U}')$.

Key challenges: i) how to sample from $K_c(\mathbb{U}')$ in practice and ii), in the nonlinear case, showing near-optimality in different problems of interest.

Caveats: This is not a panacea. Depending on the problem or target function, there may be little or no benefit over MCS.

References

- B. Adcock, J.M. Cardenas, ND, S. Moraga, *Towards optimal sampling for learning sparse approximations in high dimensions*. High Dimensional Optimization and Probability, Springer (2021).

- B. Adcock, J. Cardenas, ND, *CAS4DL: Christoffel Adaptive Sampling for function approximation via Deep Learning*, accepted, Sampling Theory, Signal Processing, and Data Analysis (2022).

- J. M. Cardenas, B. Adcock, ND. *CS4ML: A general framework for active learning with arbitrary data based on Christoffel functions*, NeurIPS (spotlight paper) (2023).

- B. Adcock, J. M. Cardenas, ND. *A unified framework for learning with nonlinear model classes from arbitrary linear samples*, submitted (2023).