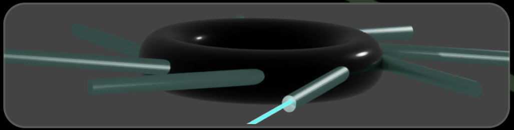
Advanced Gaussian Process Function Approximation for Uncertainty Quantification and Autonomous Experimentation

Focus: Kernel Designs and Scalability

Marcus M. Noack

Lawrence Berkeley National Lab
MarcusNoack@lbl.gov

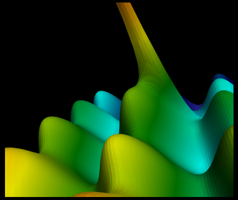
gpCAM CAMERA Applied Math NEUTRONS FOR SOCIETY MOLECULAR FOUNDRY NSLS-II NATIONAL SYNCHROTRON LIGHT SOURCE II ALS ADVANCED LIGHT SOURCE BERKELEY LAB Lawrence Berkeley National Laboratory



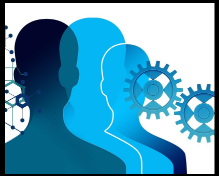
Motivation: Stochastic Modeling and Autonomous Experimentation

Preliminaries: The Basics of Gaussian-Processes and Autonomous Data Acquisition

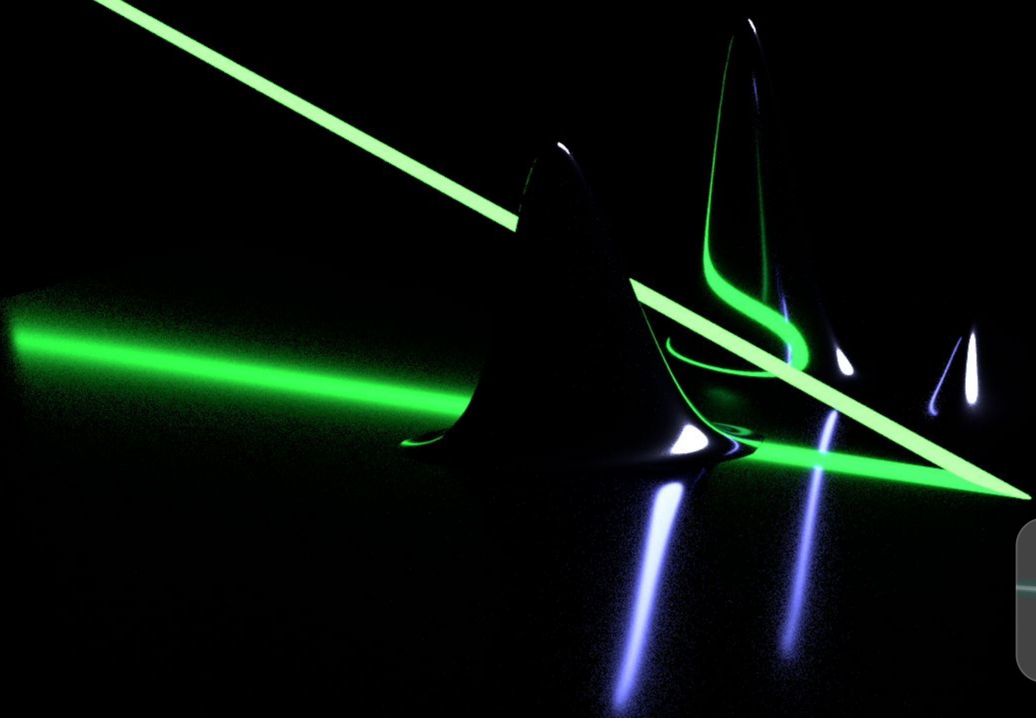
Challenges: Approximation Accuracy, UQ, Domain Awareness, and Scalability

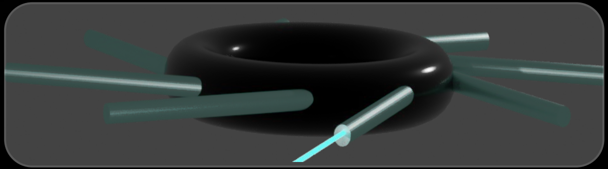


Advancements: Flexible Non-Stationary and Compactly Supported Kernel Designs



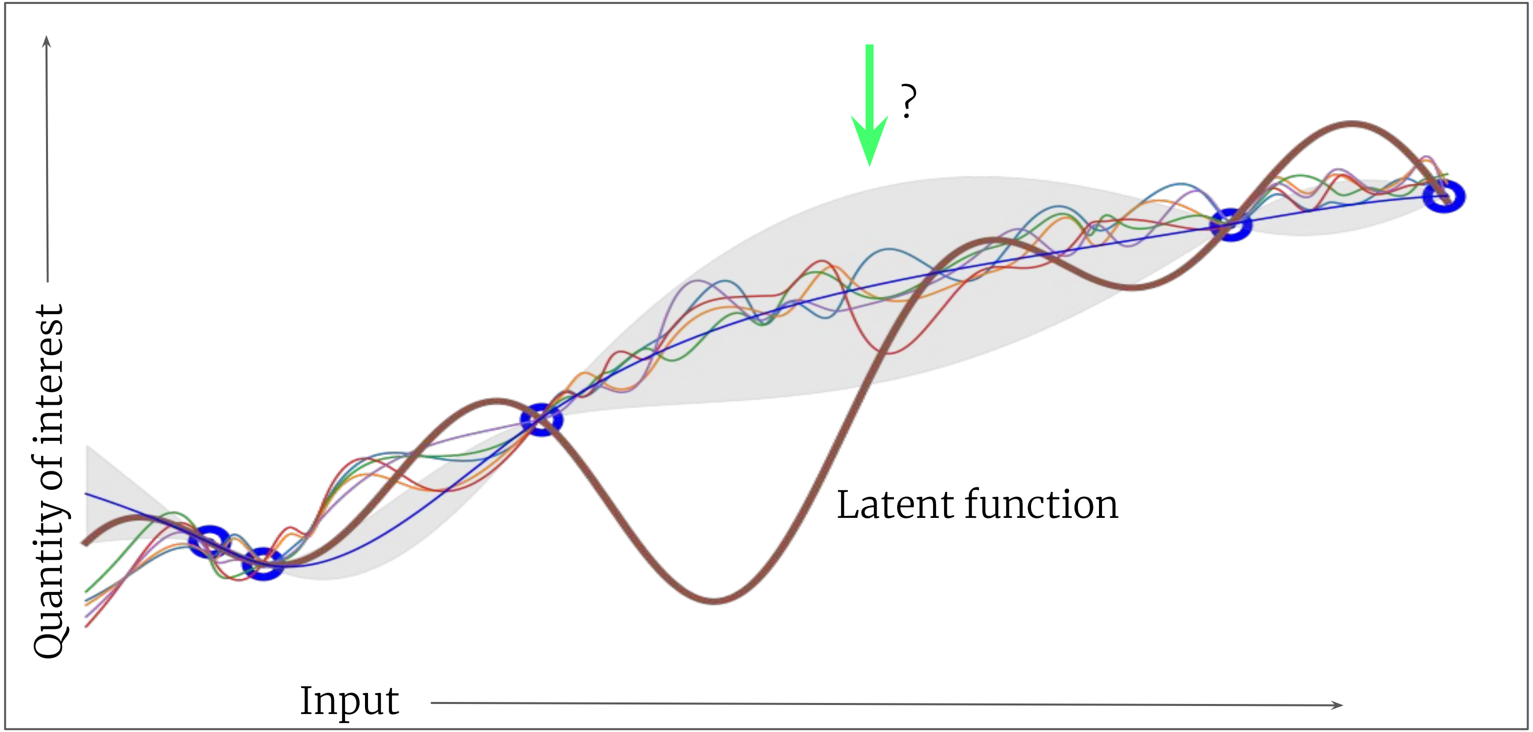
Synergy: Community and Software



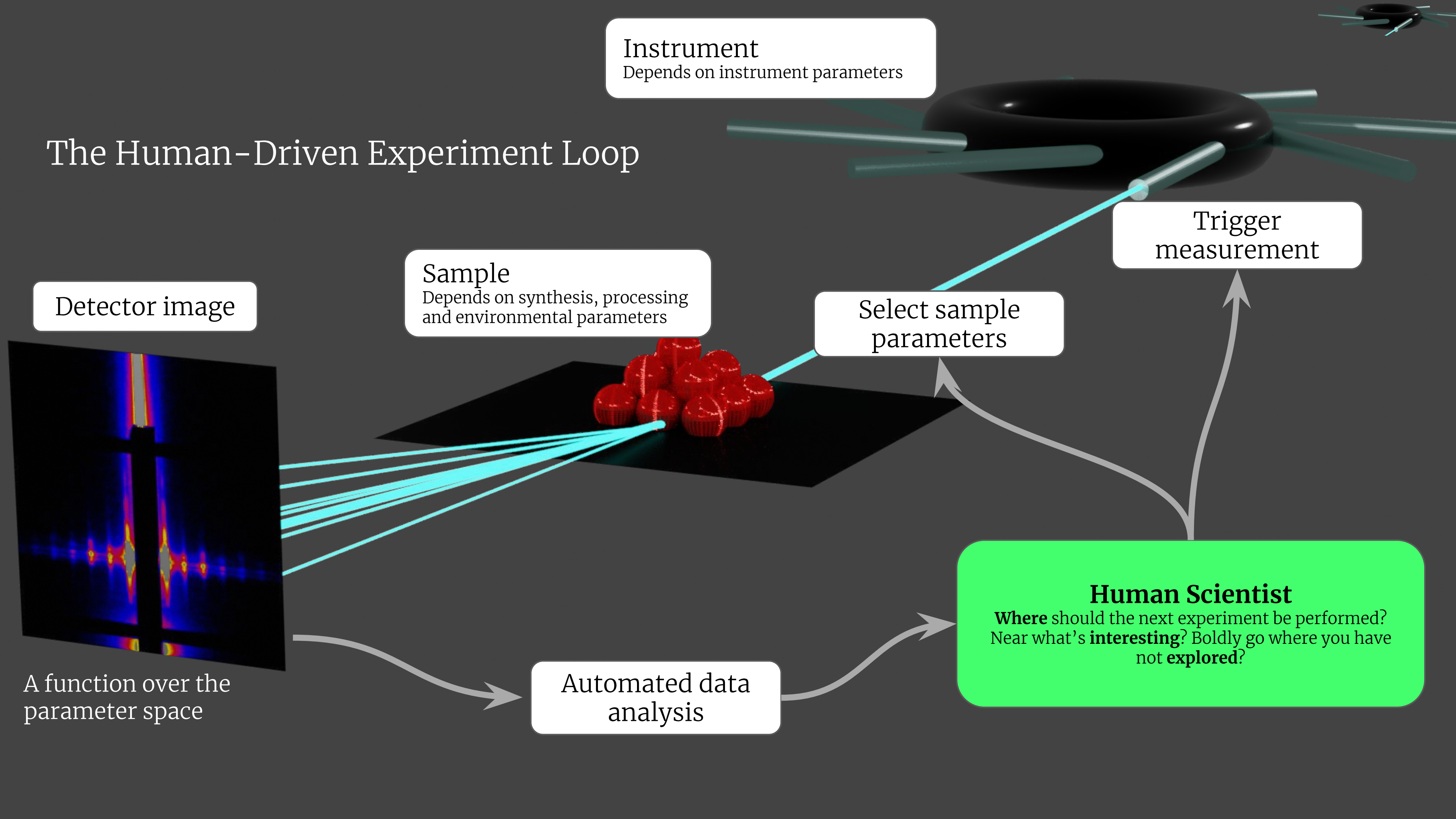


Motivation: Stochastic Modeling and Autonomous Experimentation

The **Gaussian stochastic process** is a popular framework for function approximation from noisy data.



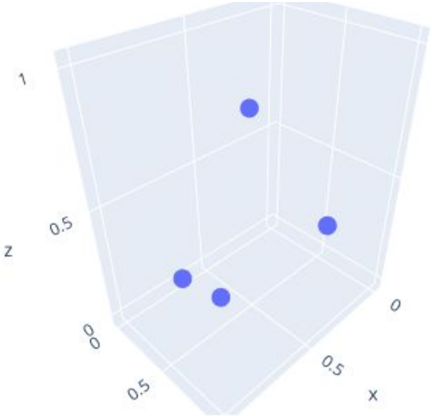
The Human-Driven Experiment Loop

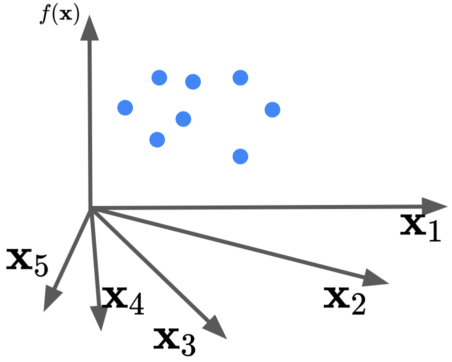


Raster scanning is collecting lots of redundant data and only works in 2d.

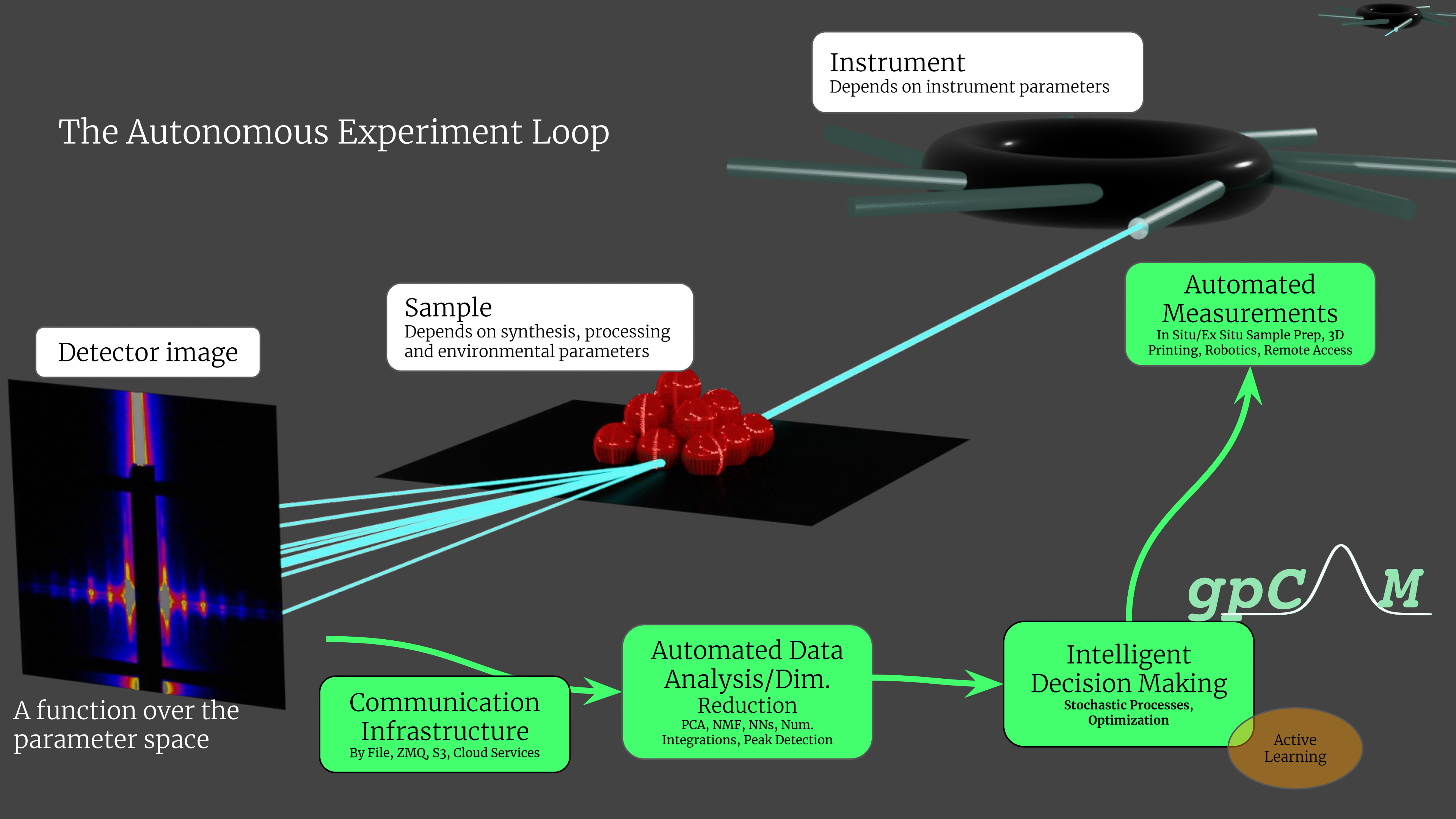
Random data collection might work in higher dims but is not optimal.

Intuitive experiment control is labor intensive and often suboptimal.



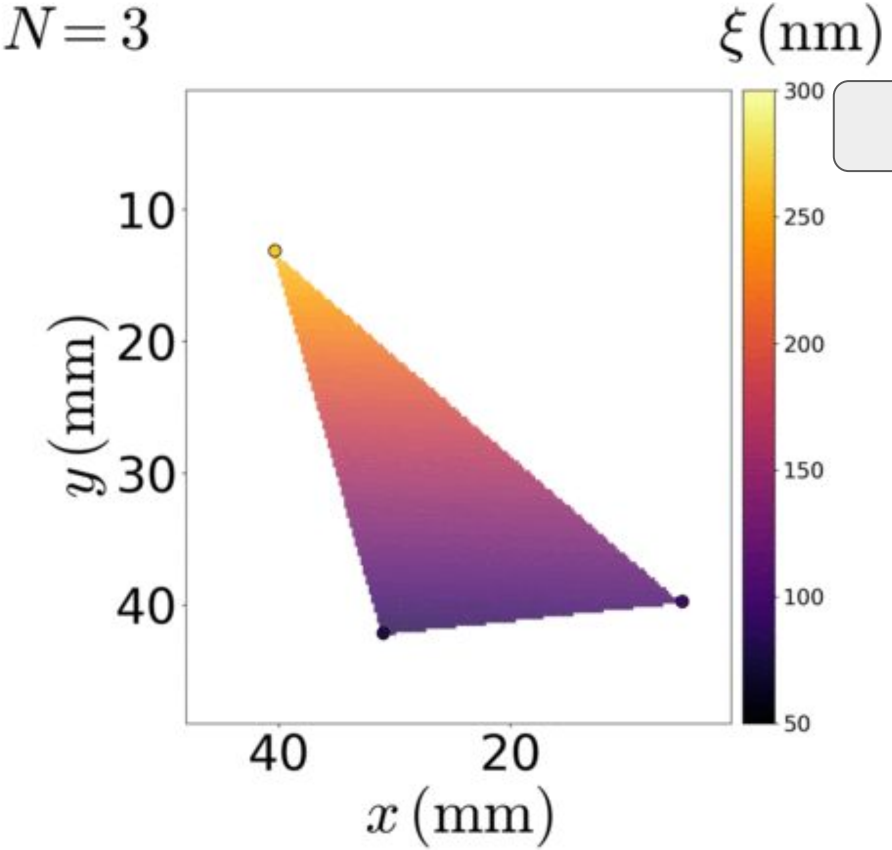


We need a way to choose optimal measurements independent of dimensionality.


The Autonomous Experiment Loop
Instrument
Depends on instrument parameters
Sample
Depends on synthesis, processing and environmental parameters
Detector image
A function over the parameter space
Automated Measurements
In Situ/Ex Situ Sample Prep, 3D Printing, Robotics, Remote Access
Communication Infrastructure
By File, ZMQ, S3, Cloud Services
Automated Data Analysis/Dim. Reduction
PCA, NMF, NNs, Num. Integrations, Peak Detection
Intelligent Decision Making
Stochastic Processes, Optimization
Active Learning
gpCAM

Autonomous SAXS Exploration of Nanoscale Ordering in a Blade-Coated Polymer-Grafted Nanorod Film

Facility: AFRL and NSLS II | **Technique:** SAXS | **Collabs:** Strait, Vaia, Fukuto, Yager, Li |
Achievement: 15% of data required, higher resolution in areas of interest



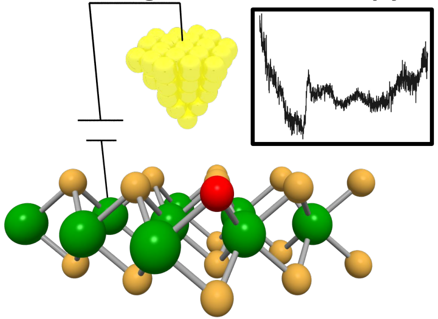
Grain Size

Center for Functional Nanomaterials
Brookhaven National Laboratory

Defect Identification through Autonomous Scanning Tunneling Spectroscopy

Facility: Molecular Foundry @ LBNL | **Technique:** STS Microscopy | **Collabs:** Thomas, Rossi |
Achievement: ~4% of data required, ~35 hrs vs ~1 mo acq. Time

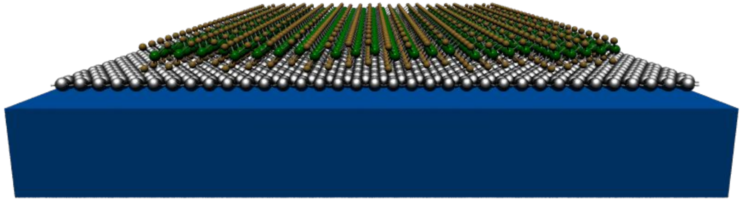
Scanning Probe Microscopy

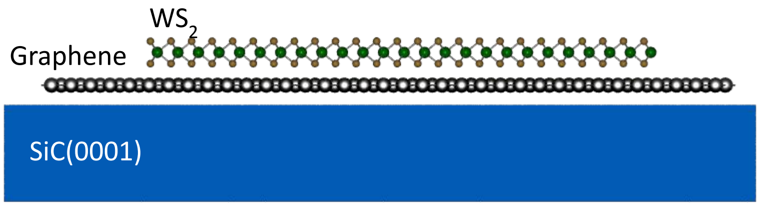


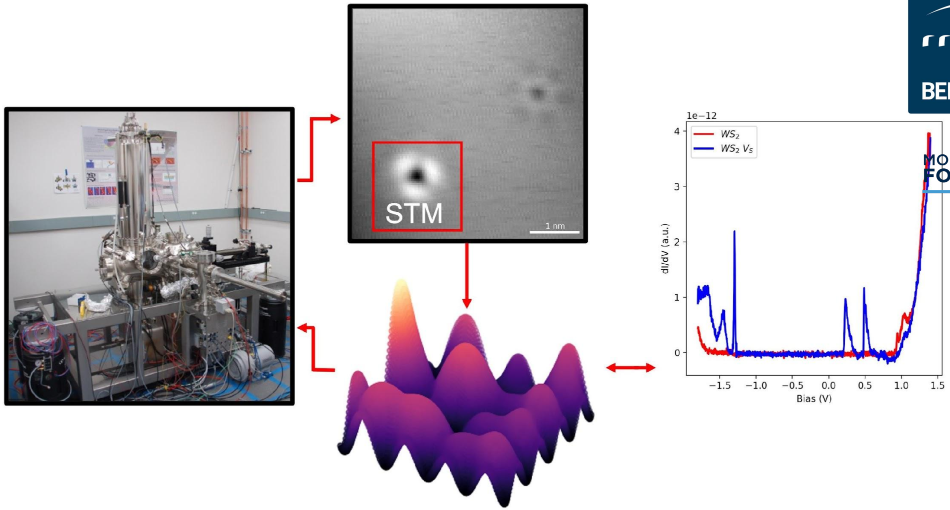
STM / STS

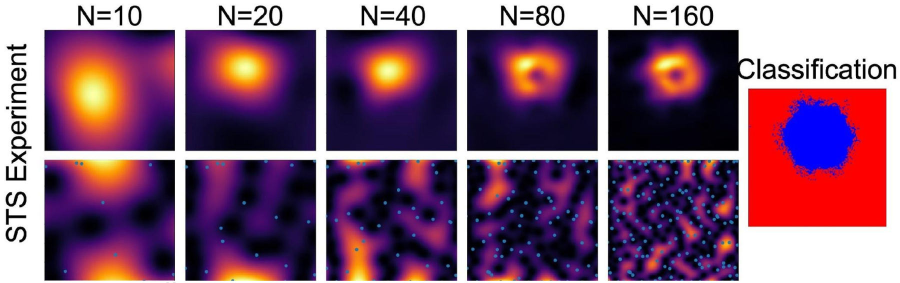
Structure and electronic properties

Investigate Next Frontiers in 2D Quantum Materials







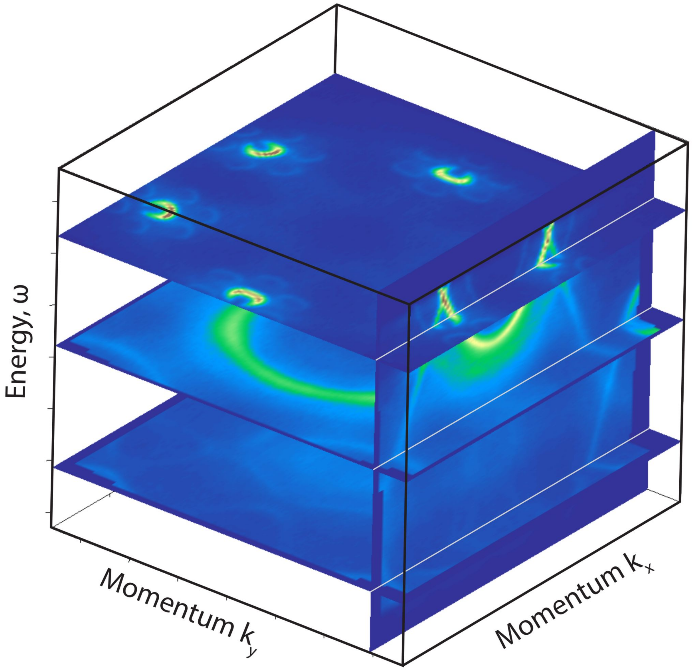


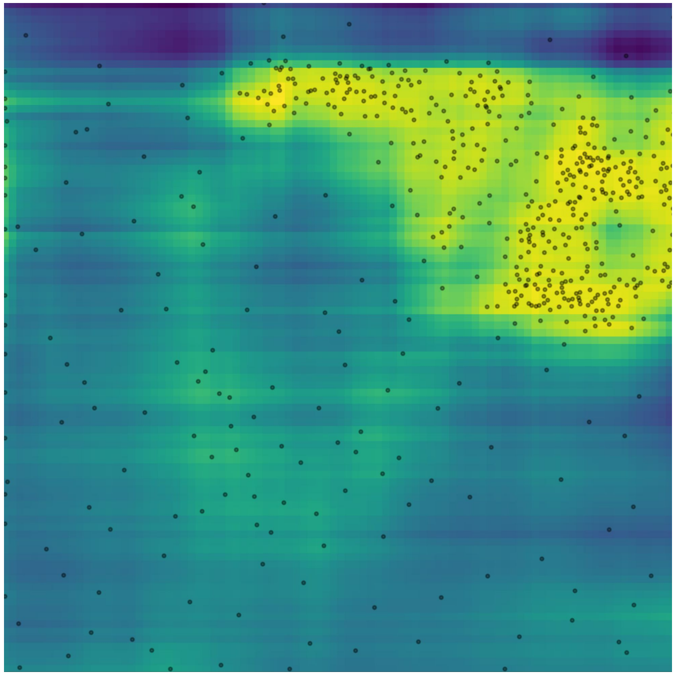
Thomas, John C., et al. "Autonomous scanning probe microscopy investigations over WS2 and Au {111}." npj Computational Materials 8.1 (2022): 1-7.

Autonomous Steering of ARPES Data Acquisition

Facilities: ALS @ LBNL | **Technique:** ARPES | **Collabs:** Melton, Rotenberg, Zwart, Hexemer |
Achievement: 12% of data required

ALS
ADVANCED LIGHT SOURCE



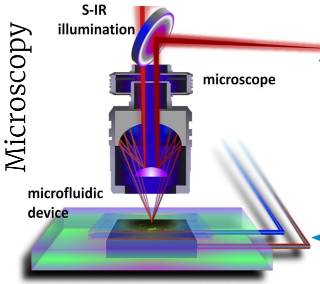


K-Means-Driven Gaussian Process Data Collection for **Angle-Resolved Photoemission Spectroscopy**,
Charles N. Melton, Marcus M. Noack, Taisuke Ohta, Thomas E. Beechem, Jeremy Robinson, Xiaotian Zhang, Aaron Bostwick, Chris Jozwiak, Roland J. Koch, Petrus H. Zwart, Alexander Hexemer, and Eli Rotenberg

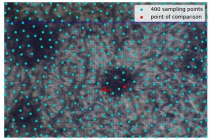
Autonomous Control of Synchrotron Infrared Spectroscopy

Facility: ALS @ LBNL | **Technique:** IR Spec. Micr. | **Collabs:** Holman, Zwart, Chen, Lee | **Achievement:** ~5% of data required, collected in ~10% of the time, materials targeted

Infrared Spectral Microscopy



GPR-based Surrogate Model

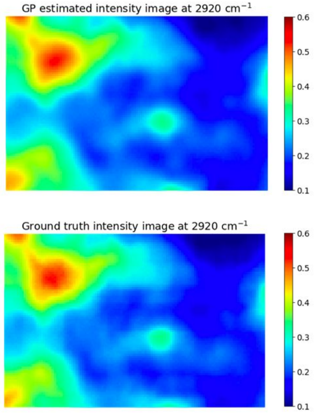


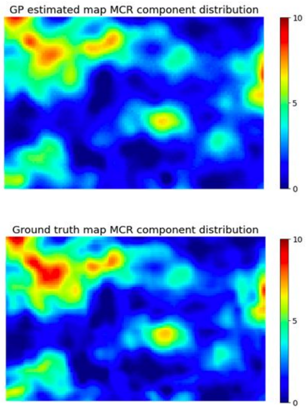
$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mu \\ \mu_*\end{bmatrix}, \begin{bmatrix} K(X,X)+\sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*)\end{bmatrix}\right)$$

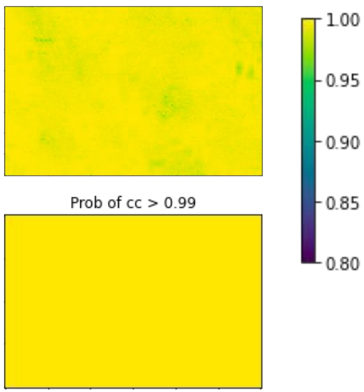
Adaptive Experimental Design

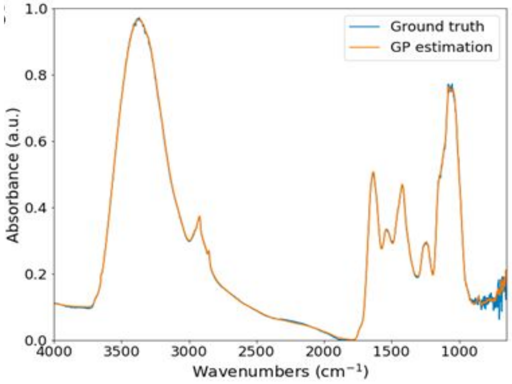
	Spectral Map	MCR Component Map	Estimated Spectral Accuracy

1 hour, 400 sample points / 9 hours, 9.8K sample points





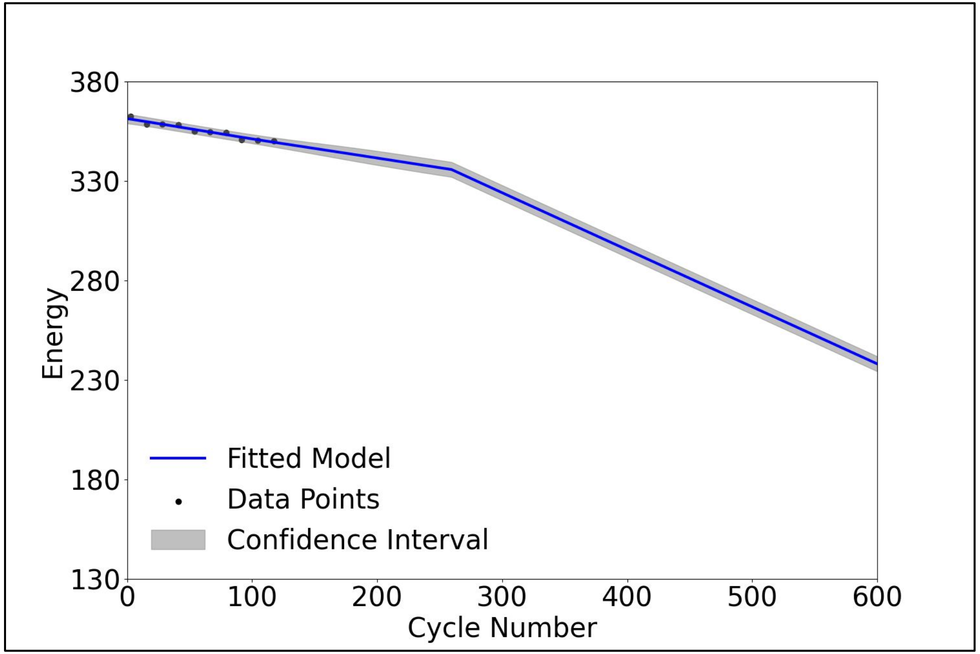


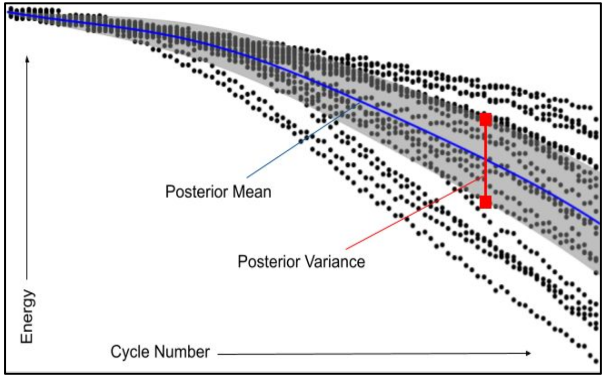


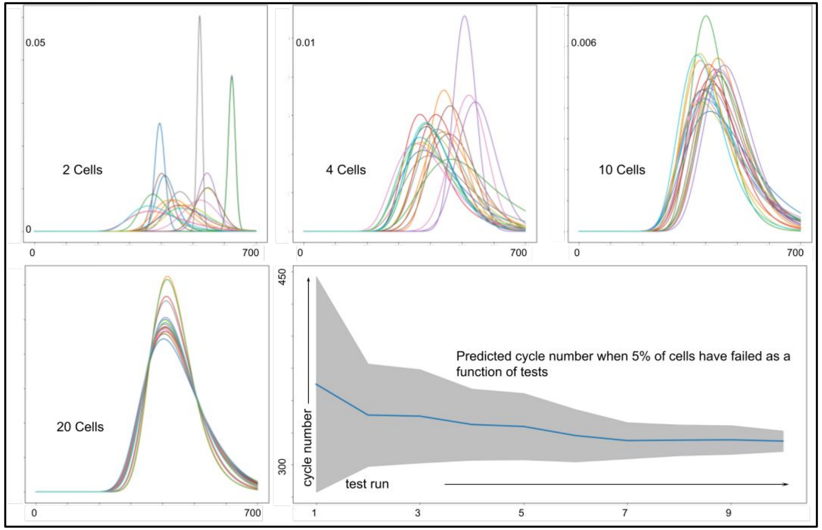
Other applied-science fields benefit from stochastic function approximations and UQ ...

Battery-lifetime prediction can be formulated as a stochastic process

Institution: ETA @LBNL | **Collabs:** Harris, Battaglia, Bakhtian | **Achievement/Objective:** Early prediction of battery lifetime

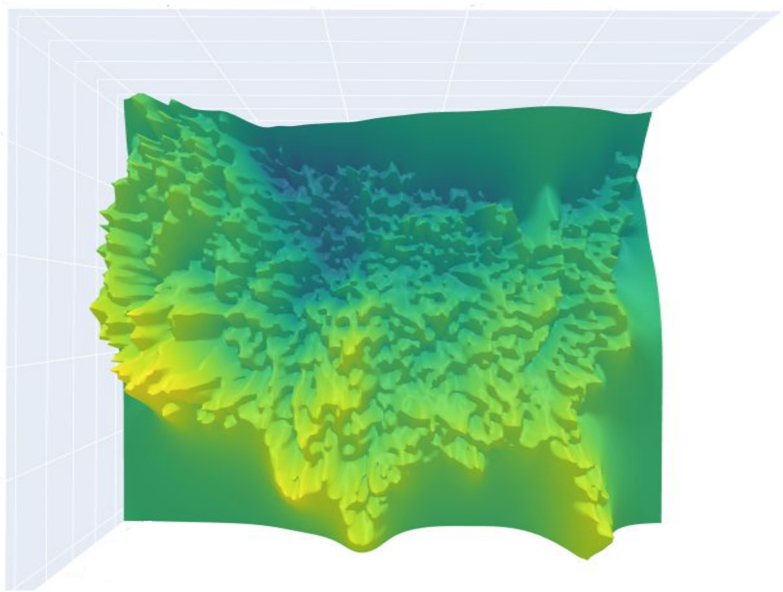


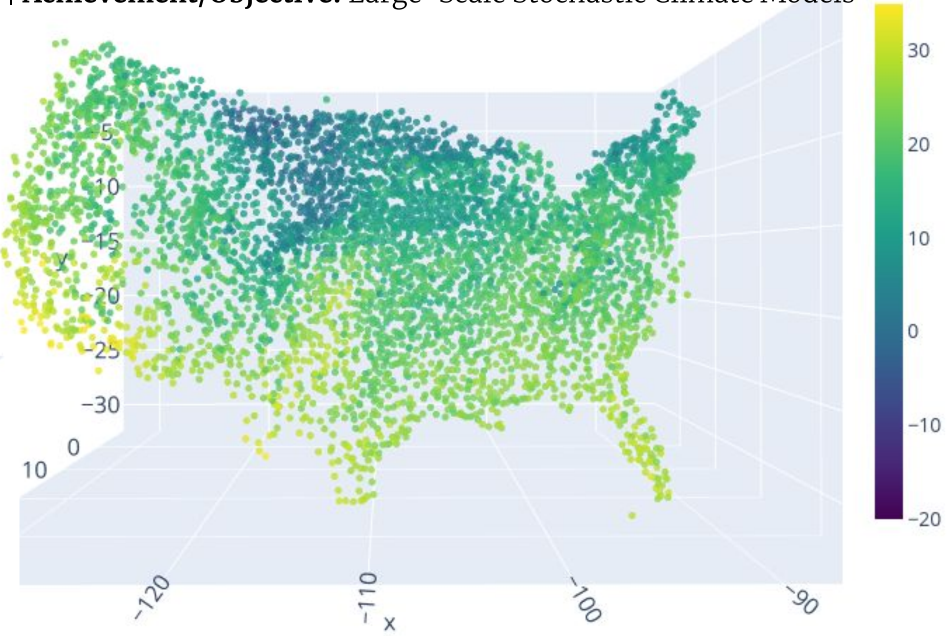


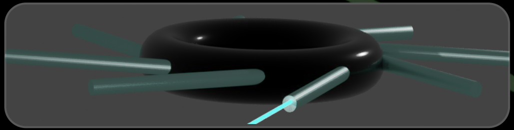


Stochastic Climate Modeling via GPs

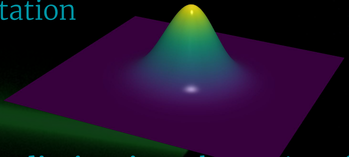
Institution: CASCADE @LBNL | **Collabs:** Mark Risser, Bill Collins | **Achievement/Objective:** Large-Scale Stochastic Climate Models



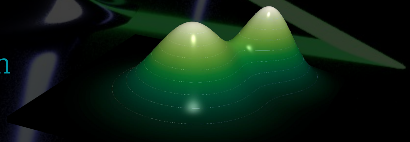




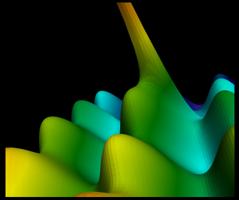
Motivation: Stochastic Modeling and Autonomous Experimentation



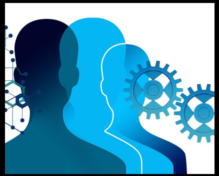
Preliminaries: The Basics of Gaussian-Processes and Autonomous Data Acquisition



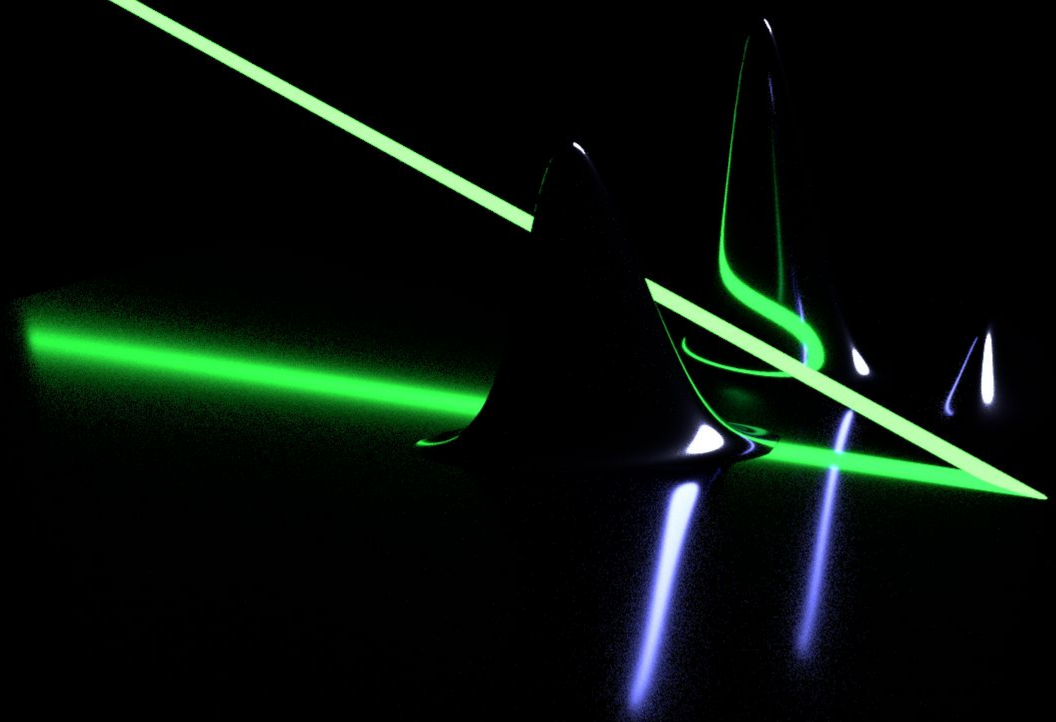
Challenges: Approximation Accuracy, UQ, Domain Awareness, and Scalability

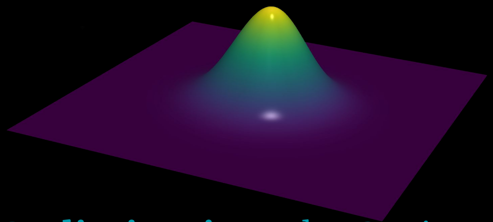


Advancements: Flexible Non-Stationary and Compactly Supported Kernel Designs



Synergy: Community and Software





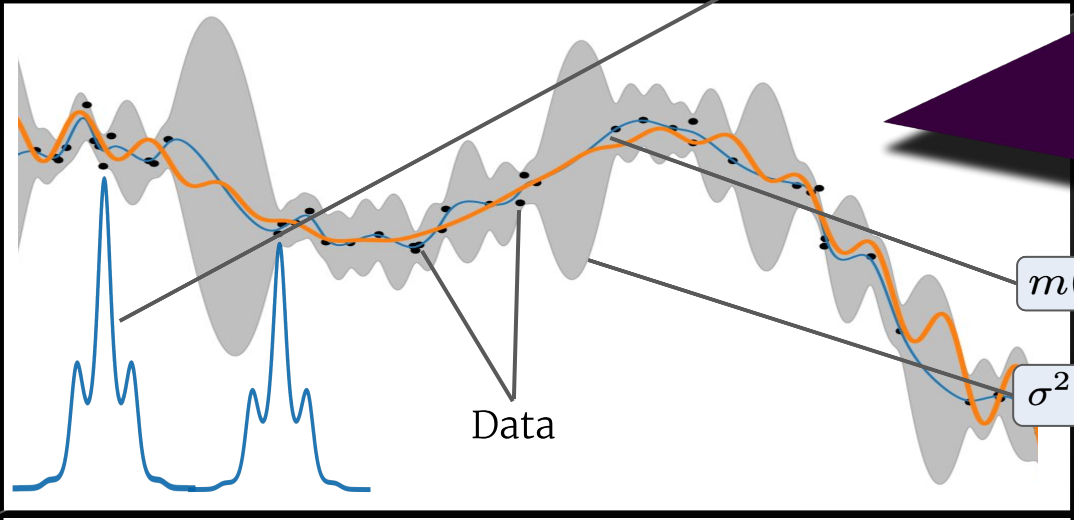
Preliminaries: The Basics of Gaussian-Processes and Autonomous Data Acquisition

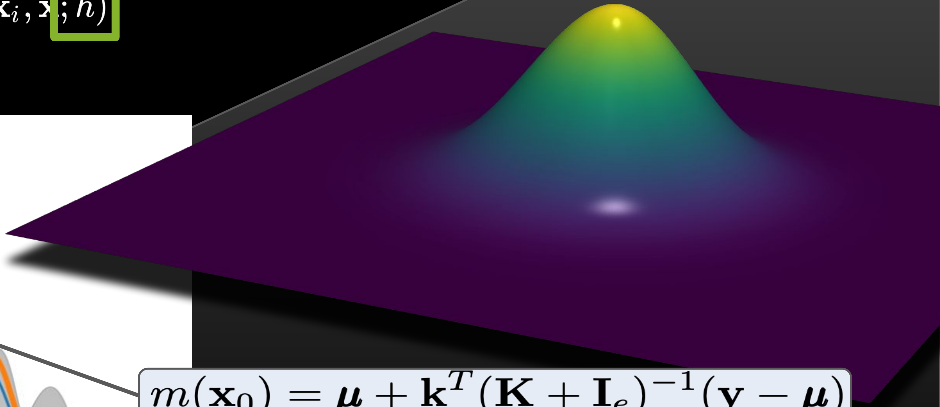
Gaussian Processes get their name from a normal distribution over functions

$$p(\mathbf{f}) = \frac{1}{\sqrt{(2\pi)^{\dim}|\mathbf{K}|}} \exp\left[-\tfrac{1}{2}(\mathbf{f}-\mathbf{m})^T \mathbf{K}^{-1} (\mathbf{f}-\mathbf{m})\right]$$

$$\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$$

$$f(\mathbf{x}) = \sum \alpha_i k(\mathbf{x}_i, \mathbf{x}; h)$$

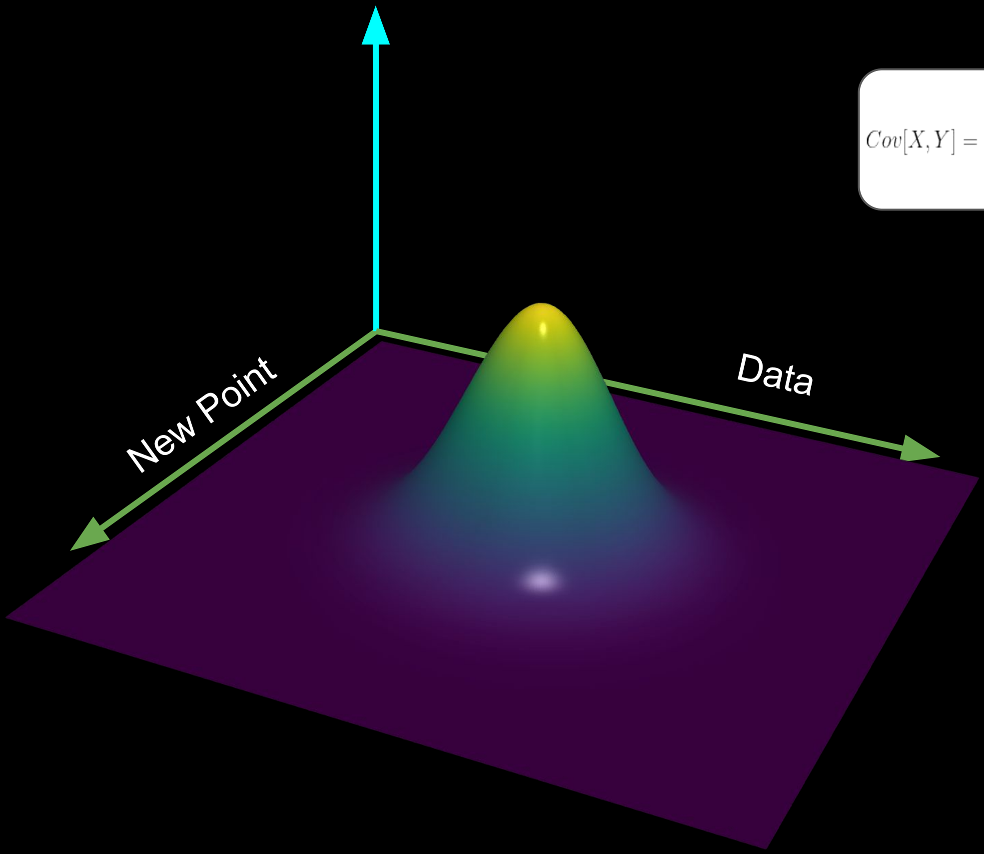




$$m(\mathbf{x}_0) = \boldsymbol{\mu} + \mathbf{k}^T (\mathbf{K} + \mathbf{I}_e)^{-1} (\mathbf{y} - \boldsymbol{\mu})$$

$$\sigma^2(\mathbf{x}_0) = k(\mathbf{x_0}, \mathbf{x_0}) - \mathbf{k}^T (\mathbf{K} + \mathbf{I}_e)^{-1} \mathbf{k}$$

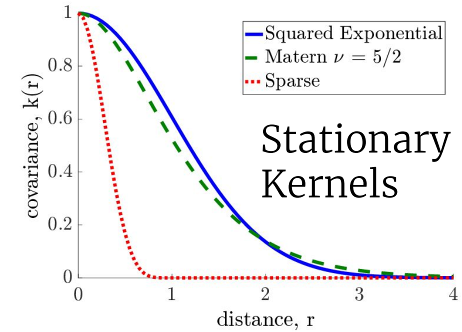
Defining the Covariance, a.k.a. the Kernel Trick



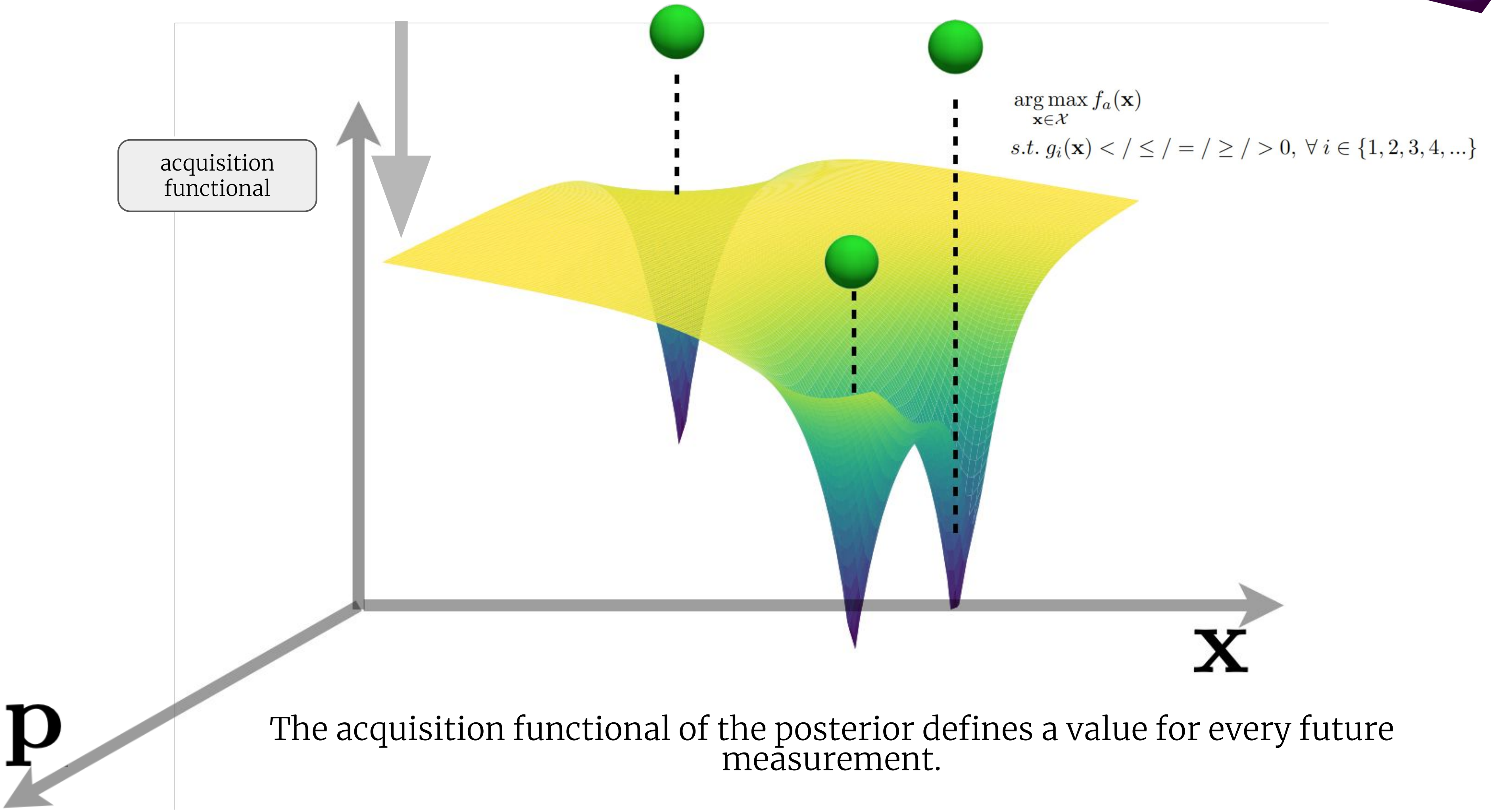
$$Cov[X,Y] = \begin{bmatrix} ((X_1 - E(X_1))(Y_1 - E(Y1))) & ((X_1 - E(X_1))(Y_2 - E(Y2))) & ((X_1 - E(X_1))(Y_3 - E(Y3))) \\ ((X_2 - E(X_2))(Y_1 - E(Y1))) & ((X_2 - E(X_2))(Y_2 - E(Y2))) & ((X_2 - E(X_2))(Y_3 - E(Y3))) \\ ((X_3 - E(X_3))(Y_1 - E(Y1))) & ((X_3 - E(X_3))(Y_2 - E(Y2))) & ((X_3 - E(X_3))(Y_3 - E(Y3))) \end{bmatrix}$$

$$Cov[X,Y] = \begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & k(\mathbf{x}_1,\mathbf{x}_2) & k(\mathbf{x}_1,\mathbf{x}_3) \\ k(\mathbf{x}_2,\mathbf{x}_1) & k(\mathbf{x}_2,\mathbf{x}_2) & k(\mathbf{x}_2,\mathbf{x}_3) \\ k(\mathbf{x}_3,\mathbf{x}_1) & k(\mathbf{x}_3,\mathbf{x}_2) & k(\mathbf{x}_3,\mathbf{x}_3) \end{bmatrix}$$

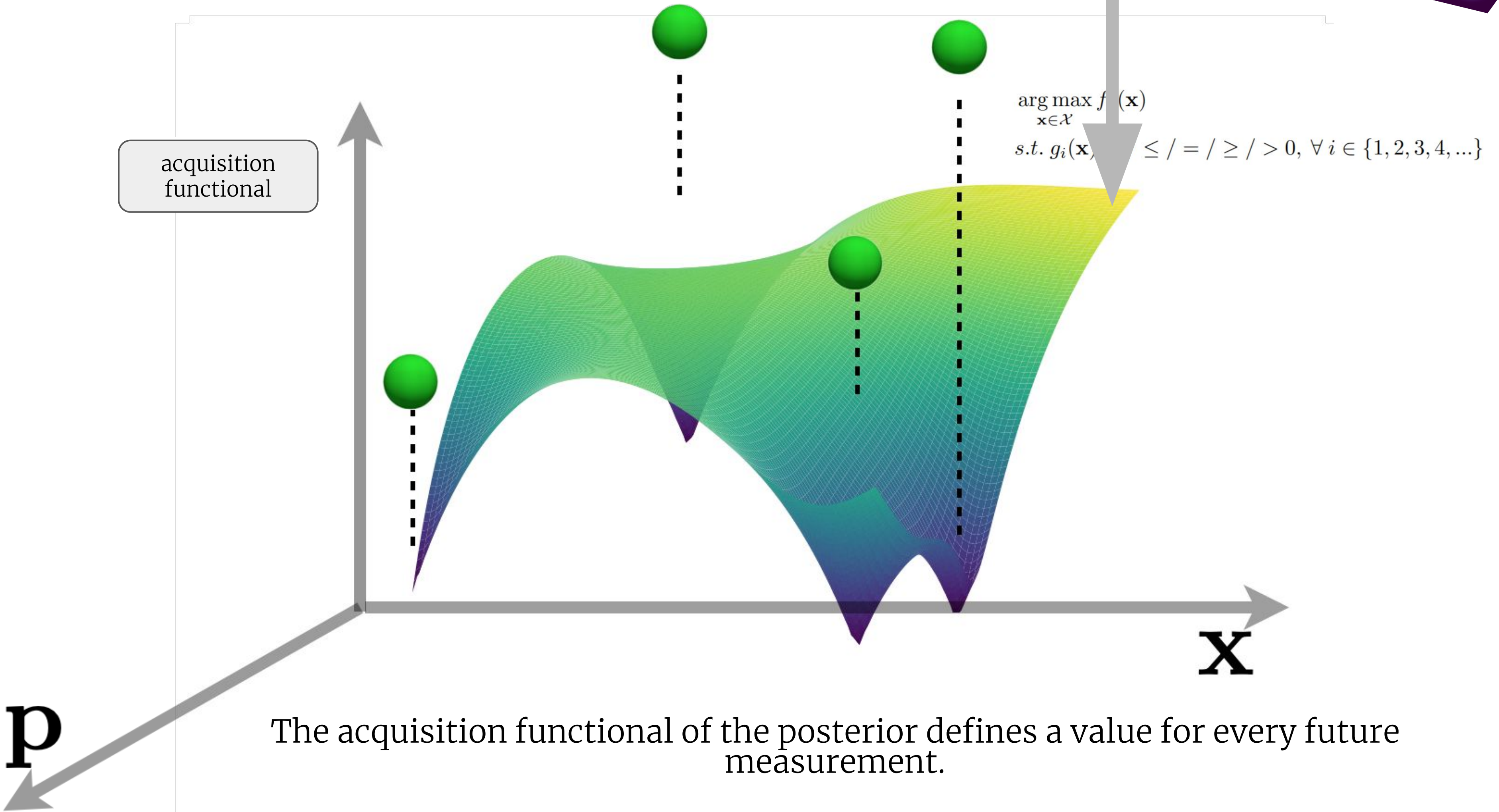
$$\sum_i^N \sum_j^N c_i\, c_j\, k(x_i, x_j) \geq 0\ \forall N,\ \mathbf{x} \in \mathcal{R}^N,\ \mathbf{c} \in \mathcal{R}^N$$



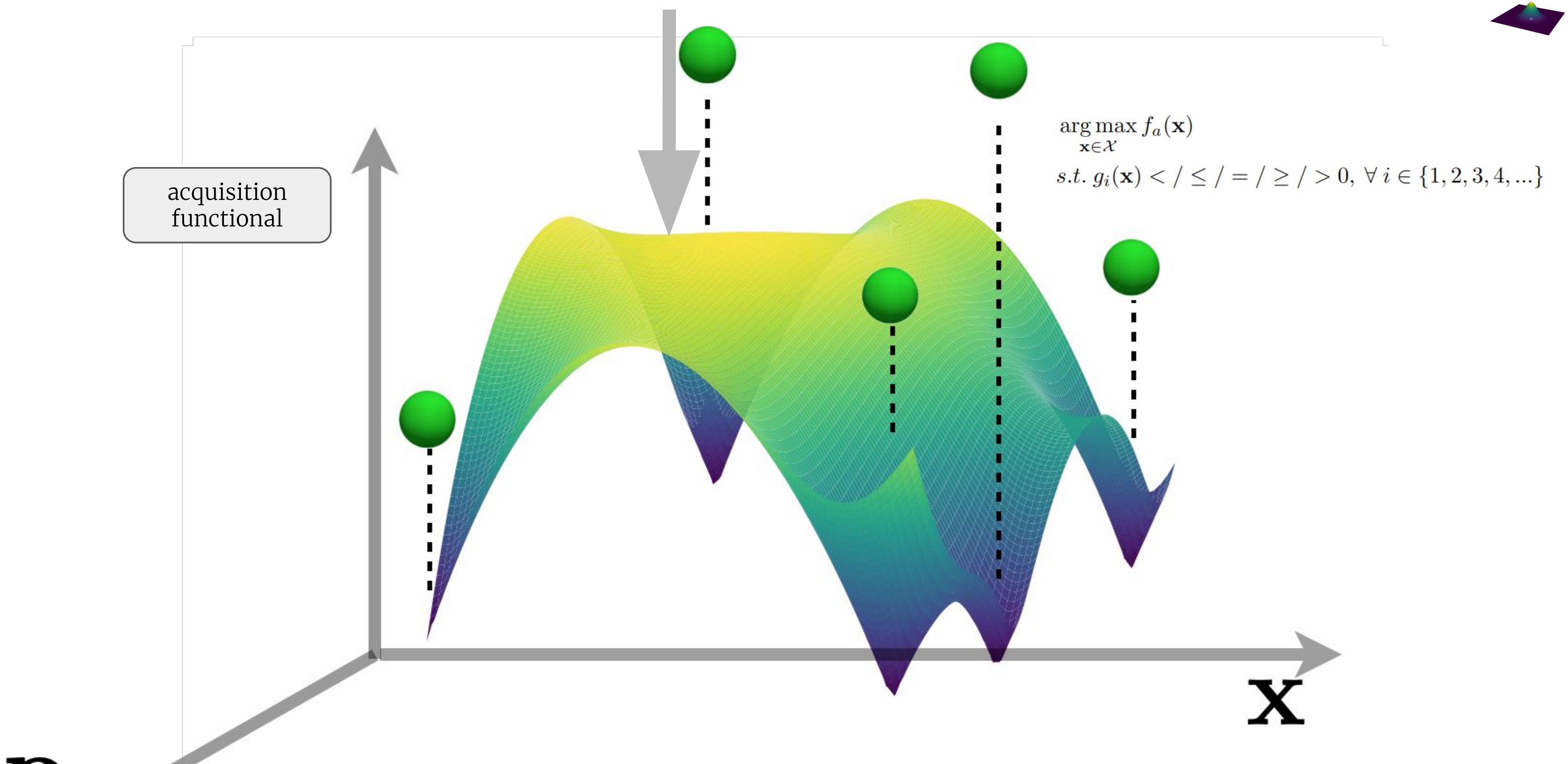
Stationary Kernels



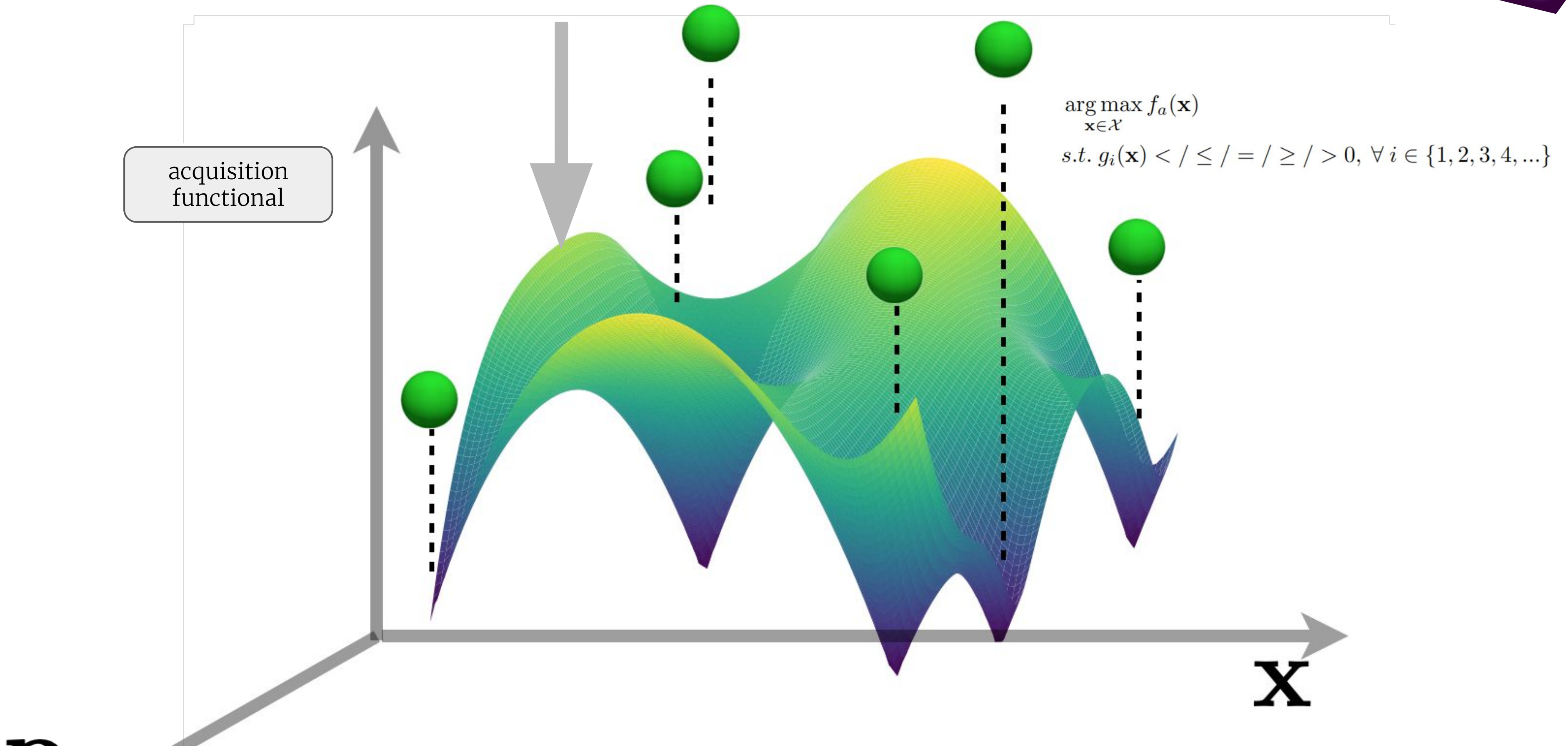
The acquisition functional of the posterior defines a value for every future measurement.



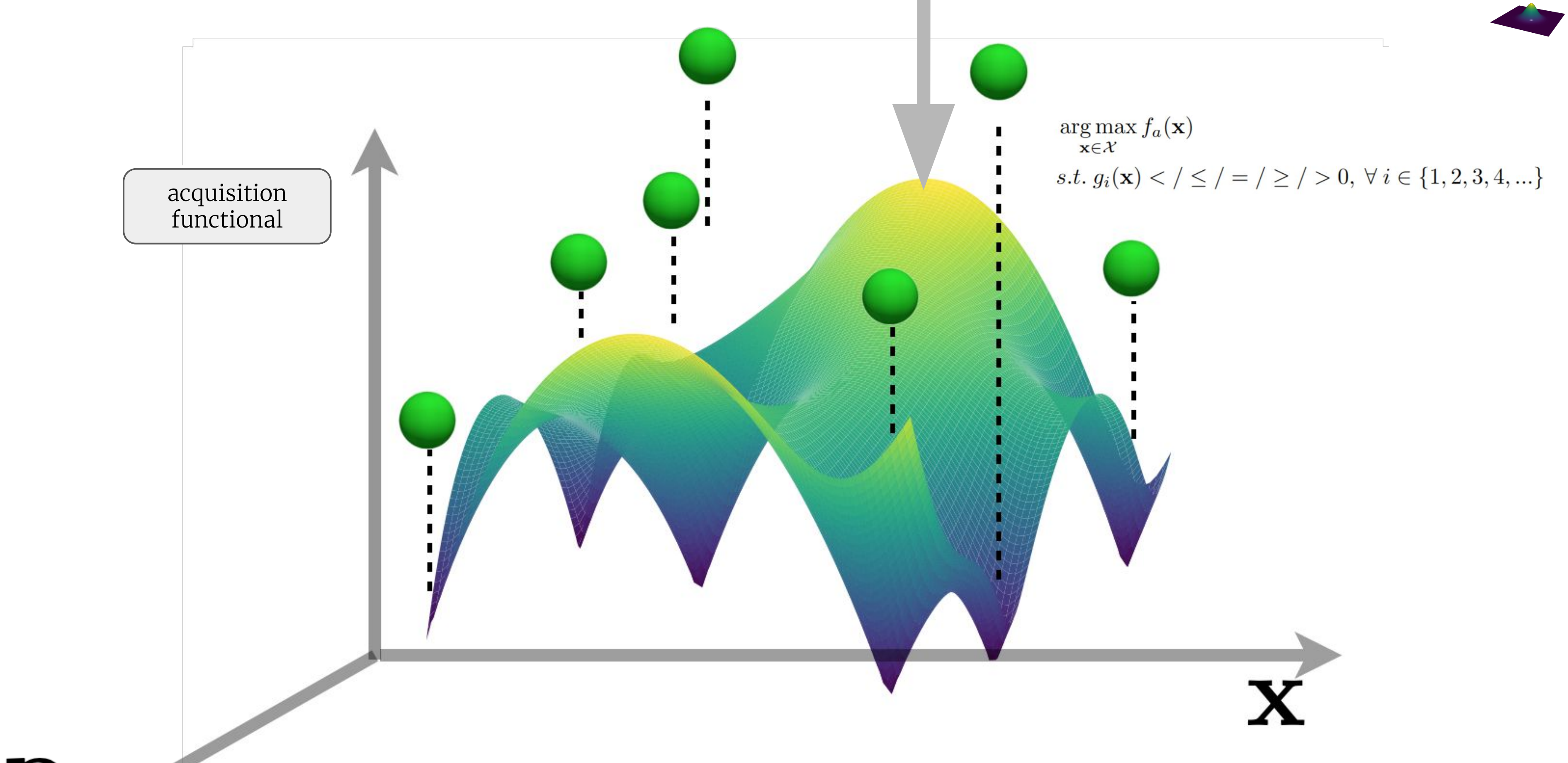
The acquisition functional of the posterior defines a value for every future measurement.



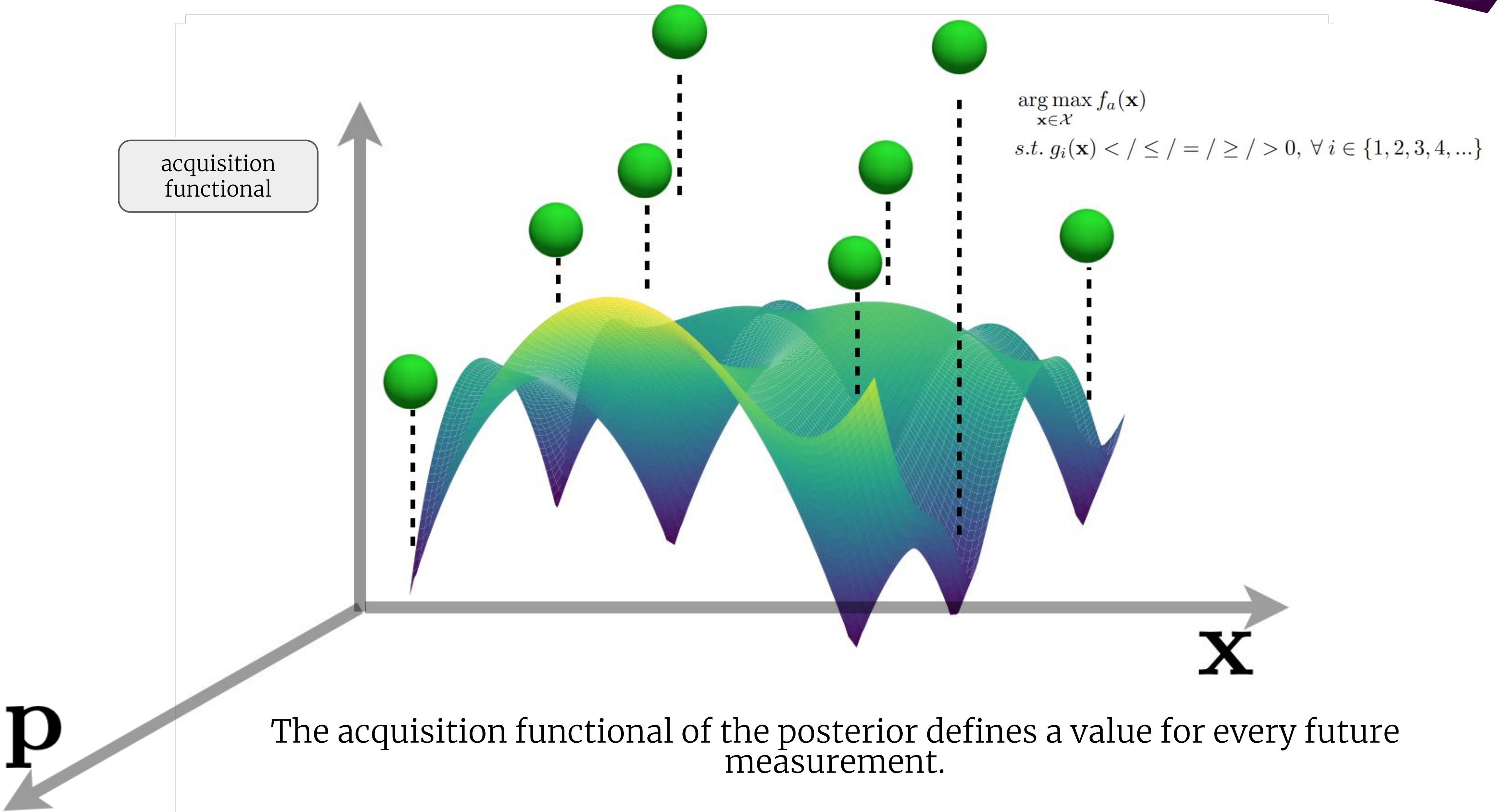
The acquisition functional of the posterior defines a value for every future measurement.



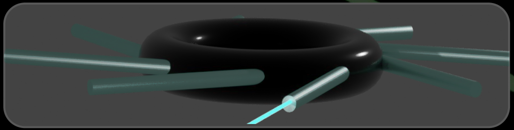
The acquisition functional of the posterior defines a value for every future measurement.



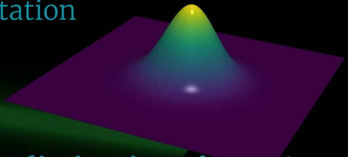
The acquisition functional of the posterior defines a value for every future measurement.



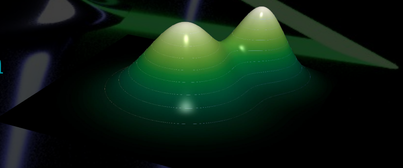
The acquisition functional of the posterior defines a value for every future measurement.



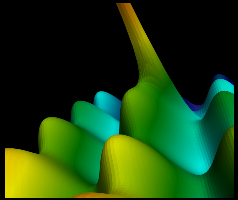
Motivation: Stochastic Modeling and Autonomous Experimentation



Preliminaries: The Basics of Gaussian-Processes and Autonomous Data Acquisition

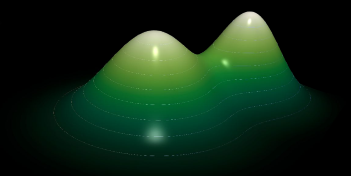


Challenges: Approximation Accuracy, UQ, Domain Awareness, and Scalability



Advancements: Flexible Non-Stationary and Compactly Supported Kernel Designs

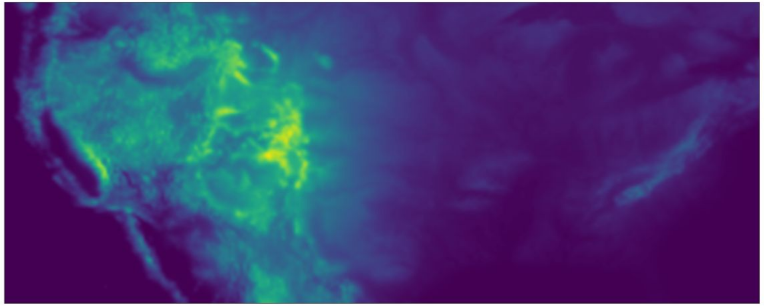
Synergy: Community and Software



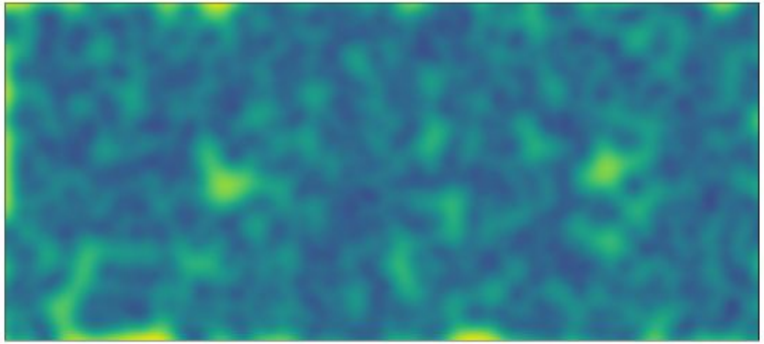
Challenges: Approximation Accuracy, UQ, Domain Awareness, and Scalability

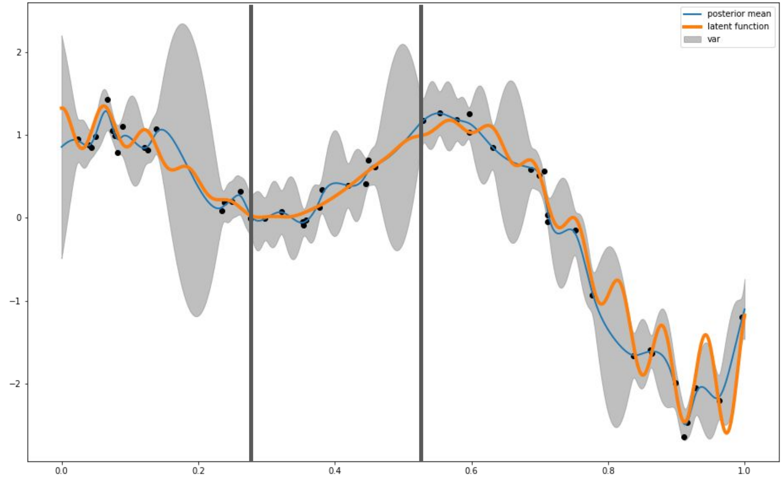
(1) Accurate uncertainty quantification is not achieved with standard GPs.

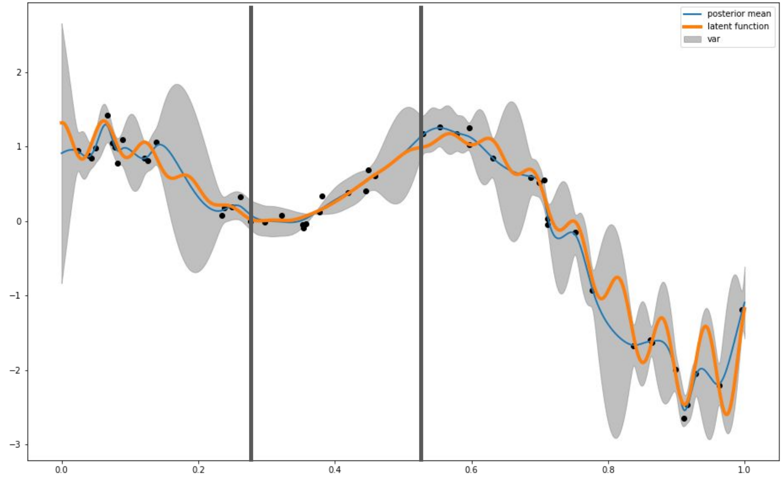
ground truth



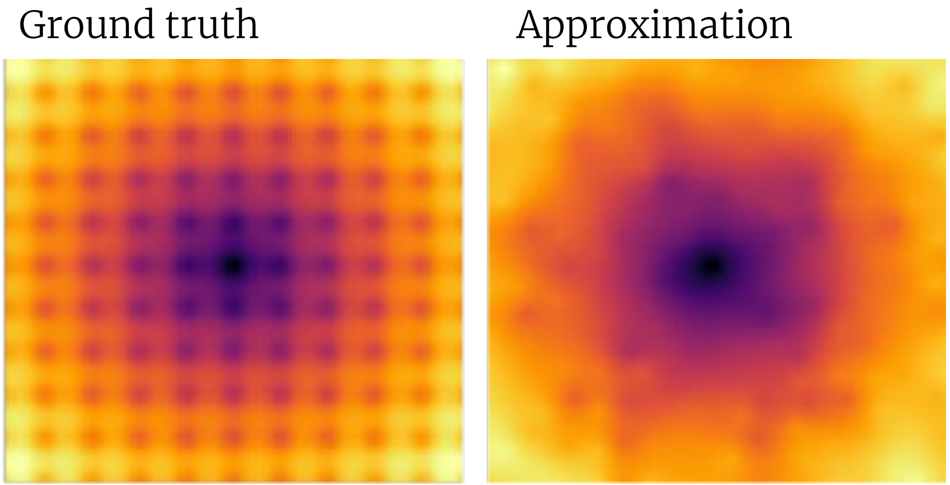
uncertainty



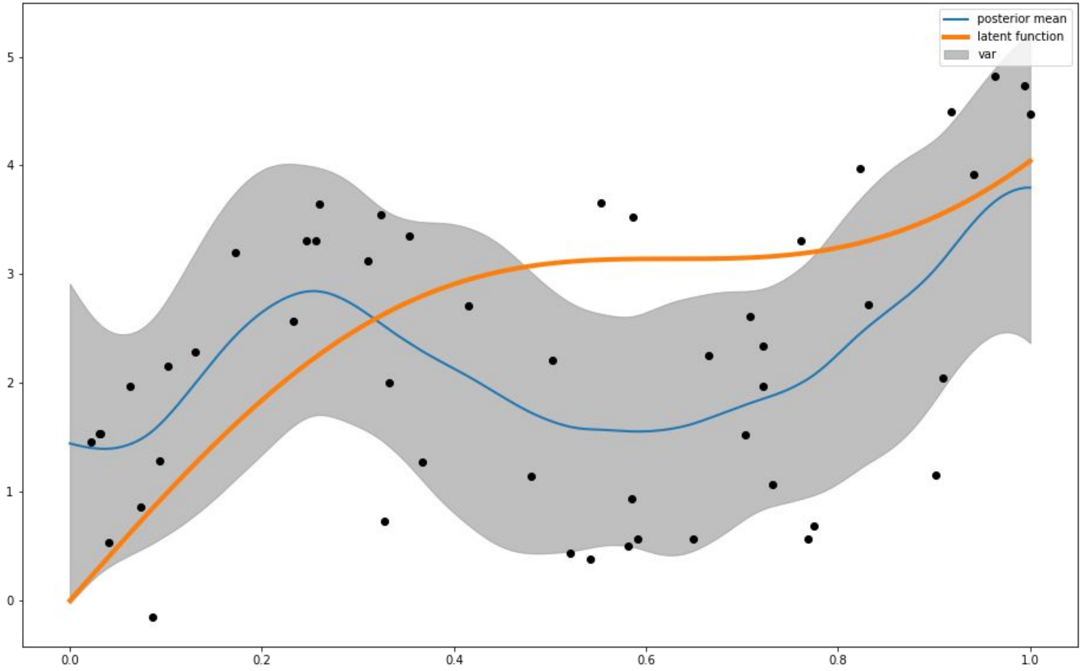


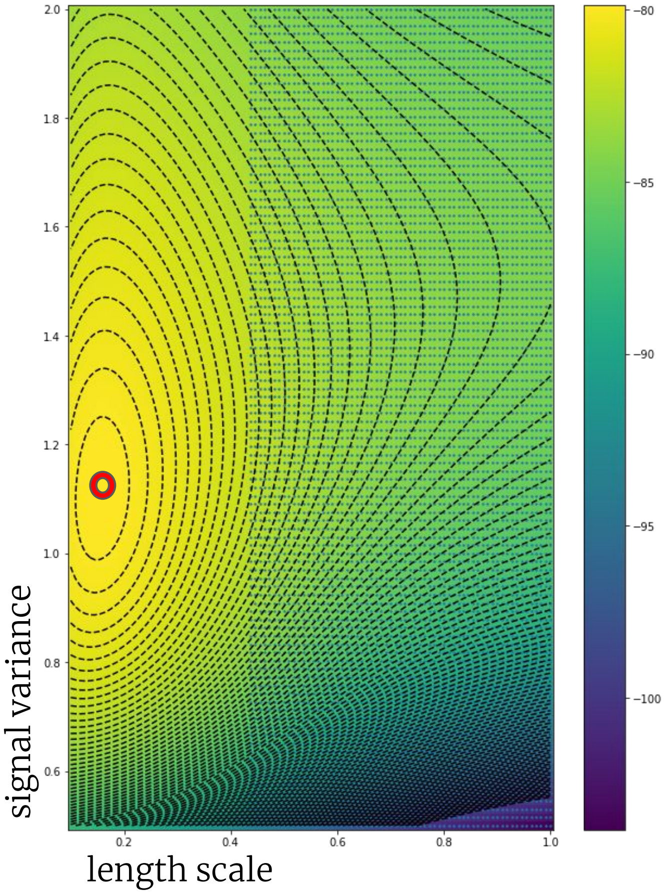


(2) Standard GPs don’t adhere to physical constraints.



(3) Standard GPs might not have (optimal) solutions that satisfy optimization constraints.





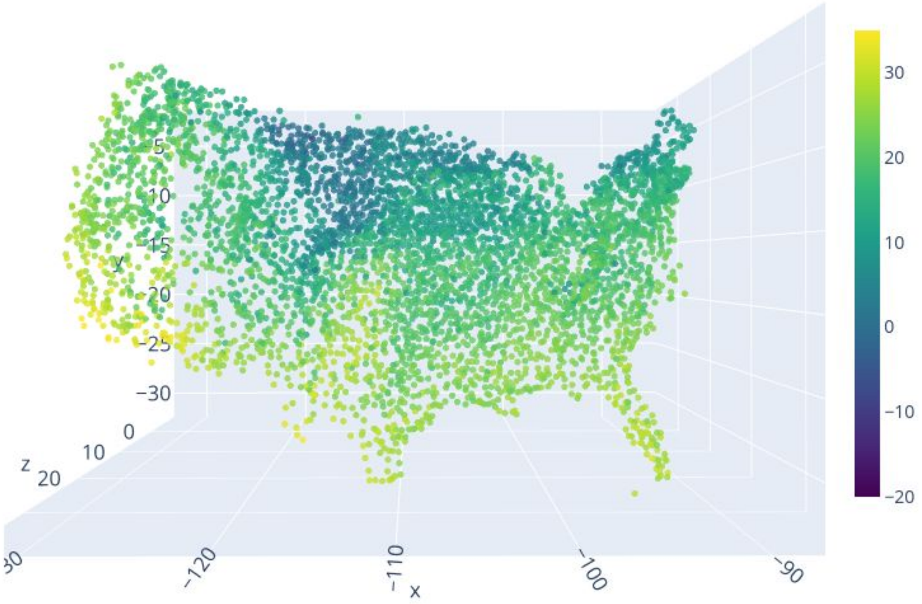
(4) GPs don't scale well.

~5000 climate stations x 10 000 days

$2.5 * 10^{15}$ floats

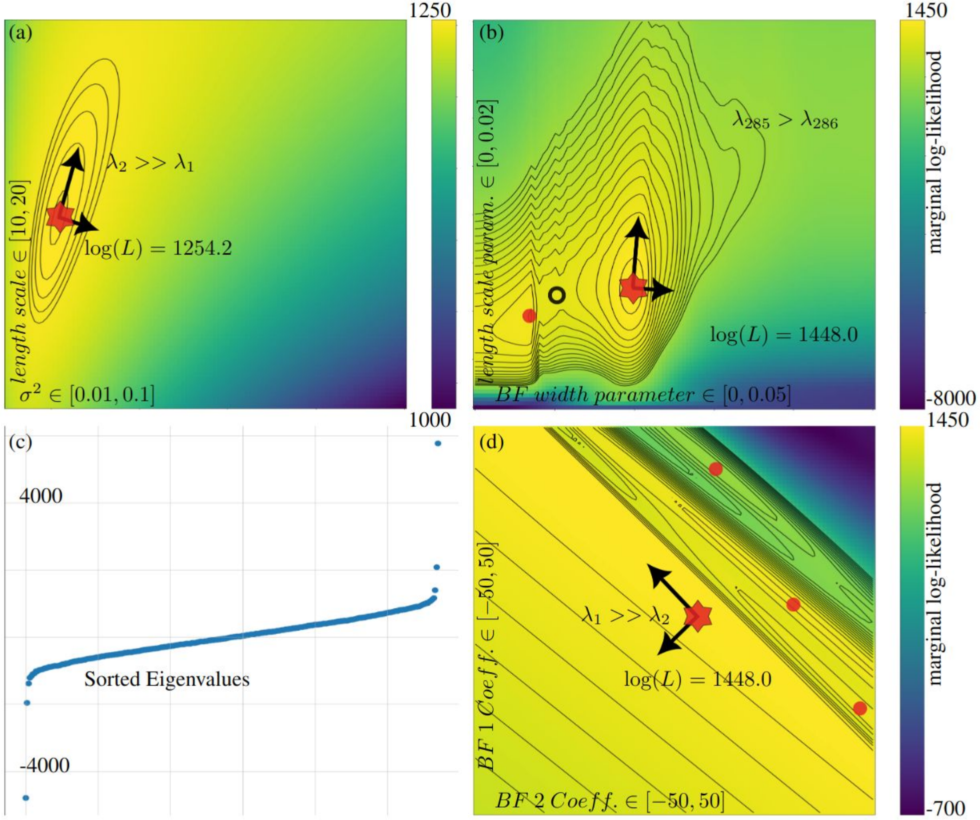
$2 * 10^{16}$ bytes = $2 * 10^{7}$ Gbytes RAM

~ 625 000 desktop computers



(5) Advanced GPs lead to tough optimization problems.

$$\arg\max_{\phi} \Big(\log(L(D,\phi)) \; = \\ -\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}(\phi))(\mathbf{K}(\phi)+\mathbf{V})^{-1}(\mathbf{y}-\boldsymbol{\mu}(\phi)) \\ -\frac{1}{2}\log(|\mathbf{K}(\phi)+\mathbf{V}|) - \frac{\dim(\mathbf{y})}{2}\log(2\pi)\Big)$$



(1) Accurate uncertainty quantification is not achieved with standard GPs.

(2) Standard GPs don’t adhere to physical constraints.

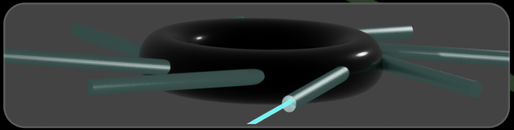
(3) Standard GPs might not have (optimal) solutions that satisfy optimization constraints.

(4) GPs don’t scale well.

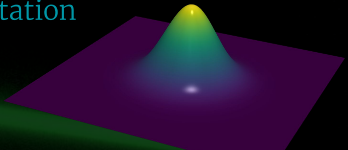
(5) Advanced GPs lead to tough optimization problems.

(6) Standard GPs don’t work on non-linear spaces.

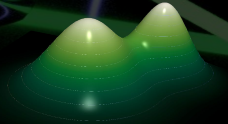
(7) Sometimes Gaussianity just doesn’t cut it.



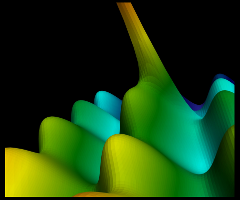
Motivation: Stochastic Modeling and Autonomous Experimentation



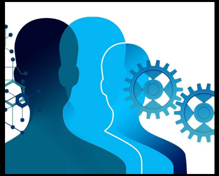
Preliminaries: The Basics of Gaussian-Processes and Autonomous Data Acquisition



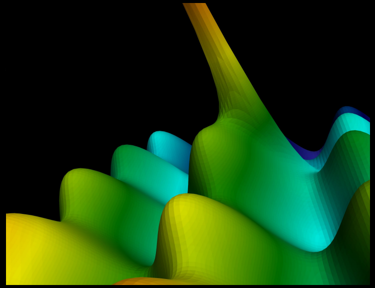
Challenges: Approximation Accuracy, UQ, Domain Awareness, and Scalability



Advancements: Flexible Non-Stationary and Compactly Supported Kernel Designs



Synergy: Community and Software



Advancements: Flexible
Non-Stationary and Compactly
Supported Kernel Designs

Kernels can be utilized to increase the flexibility of the function approximation and to inject domain-awareness into the model

Stationary Kernels:

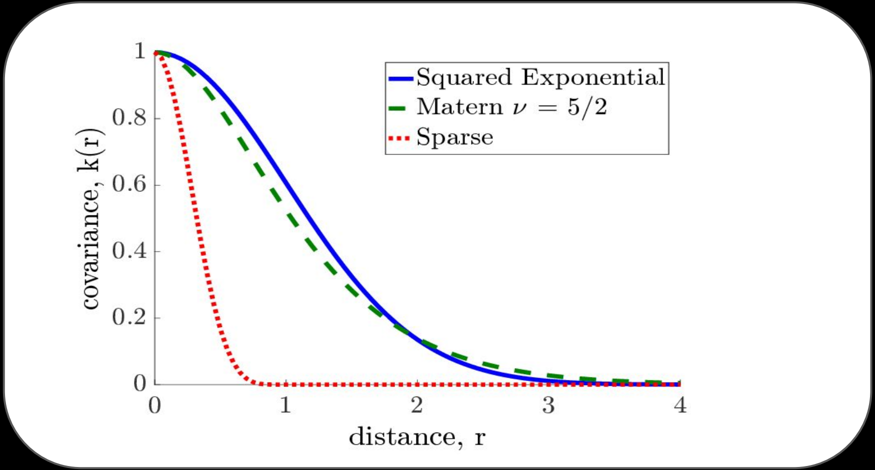
$$k(\mathbf{x}_1, \mathbf{x}_2) = \sigma_s \exp[-\frac{1}{2l}||\mathbf{x}_1 - \mathbf{x}_2||^2] \quad (1)$$

Non-Stationary Kernels:

$$k(\mathbf{x}_1, \mathbf{x}_2) = \frac{\sigma_s^2(\mathbf{x}_1)\,\sigma_s^2(\mathbf{x}_2)}{\sqrt{|\frac{\mathbf{\Sigma}(\mathbf{x}_1)+\mathbf{\Sigma}(\mathbf{x}_2)}{2}|}} \mathcal{M}(\sqrt{Q(\mathbf{x}_1, \mathbf{x}_2)}) \quad (2)$$

$$k(\mathbf{x}_1, \mathbf{x}_2) = k(g(\mathbf{x}_1), g(\mathbf{x}_2)) \quad (3)$$

$$k(\mathbf{x}_1, \mathbf{x}_2) = f(\mathbf{x}_1) f(\mathbf{x}_2) \quad (4)$$

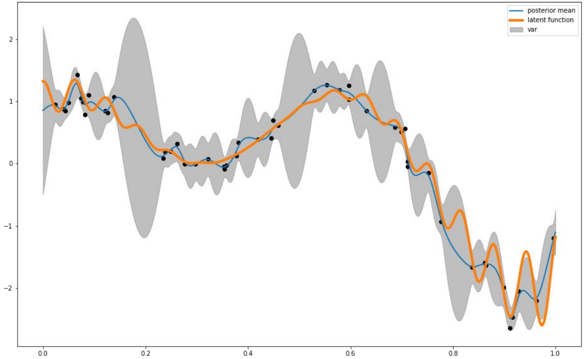


f,g can be any operator (NN, RBS, piecewise linear or constant)

Customized stationary and non-stationary kernels can give GPs superpowers ...

Non-stationary kernels can make UQ much more realistic.

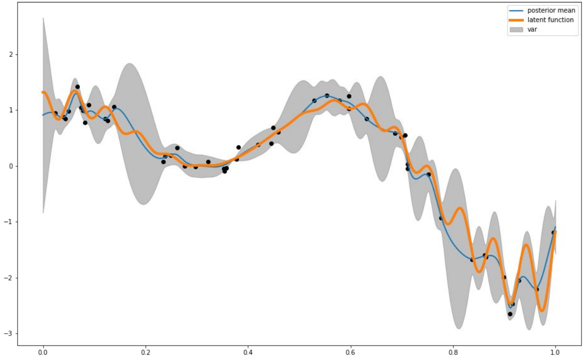
Stationary



$$k_{stat}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_s^2 \left(1 - \frac{\sqrt{3d}}{l}\right) \exp\left[-\frac{\sqrt{3d}}{l}\right]$$

d...Euclidean distance
l... length scale
σ_s^2 ... signal variance

Parametric Non-Stationary



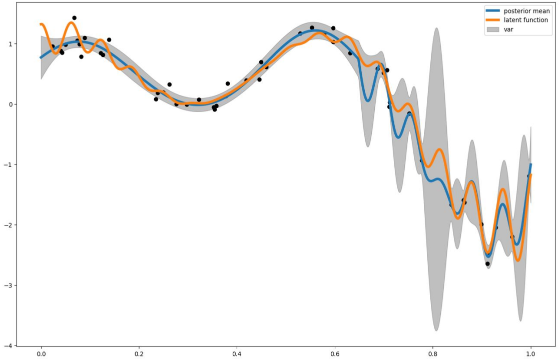
$$k_{non}(\mathbf{x}_1, \mathbf{x}_2) = f(\mathbf{x}_1) f(\mathbf{x}_2)\ k_{stat}(\mathbf{x}_1, \mathbf{x}_2)$$

$$f(x) = \sum_i^N \alpha_i \beta(\mathbf{x}_i, \mathbf{x}; w)$$

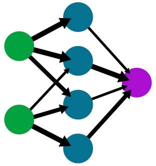
$$\beta(\mathbf{x}_i, \mathbf{x}; w) = \exp\left[-||\mathbf{x}_i - \mathbf{x}||\ w^2\right]$$

w ... width parameter

Deep Kernel



$$k_{deep} = k_{stat}(||g(\mathbf{x}_1) - g(\mathbf{x}_2)||)$$



Non-stationary kernels can make UQ much more realistic.

$$k_{stat}(\mathbf{x}_1, \mathbf{x}_2) = \sigma_s^2 \left(1 - \frac{\sqrt{3d}}{l}\right) \exp\left[-\frac{\sqrt{3d}}{l}\right]$$

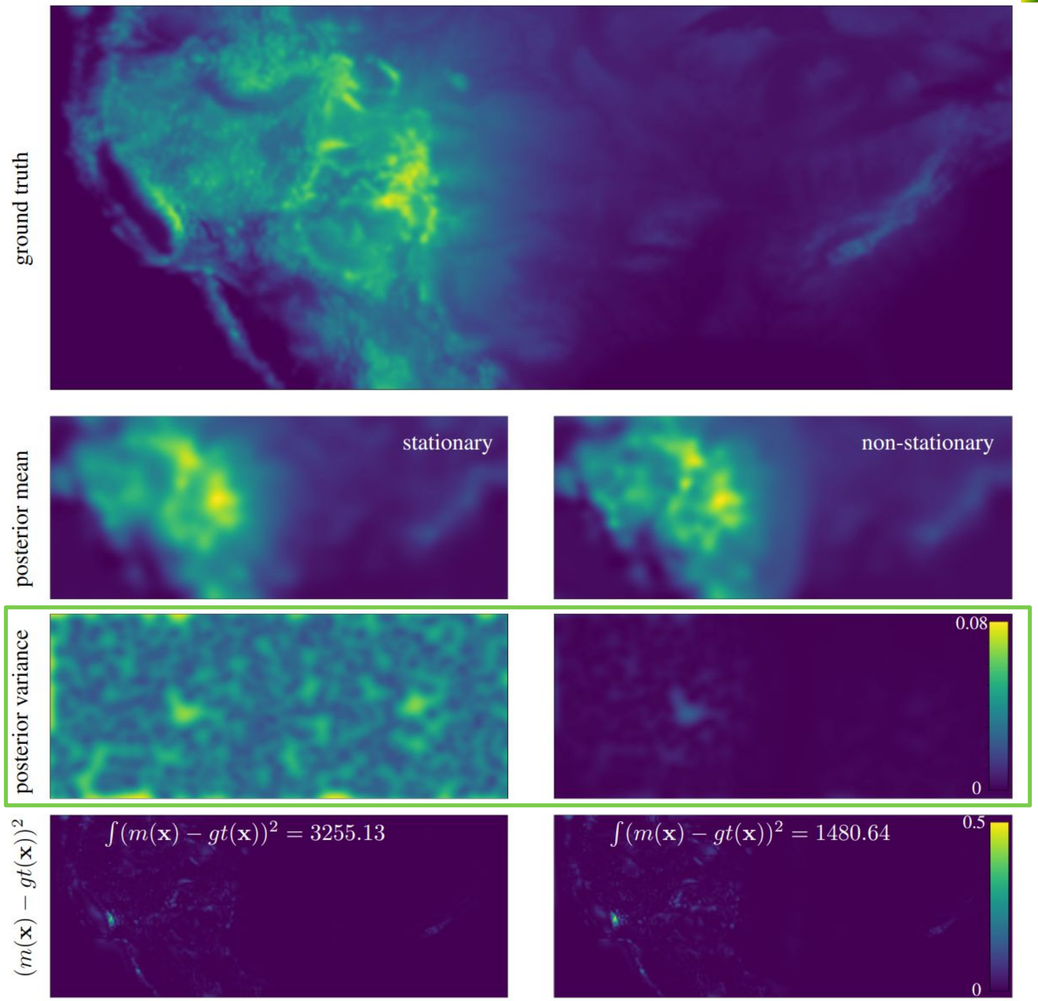
d...Euclidean distance
l... length scale
σ_s^2... signal variance

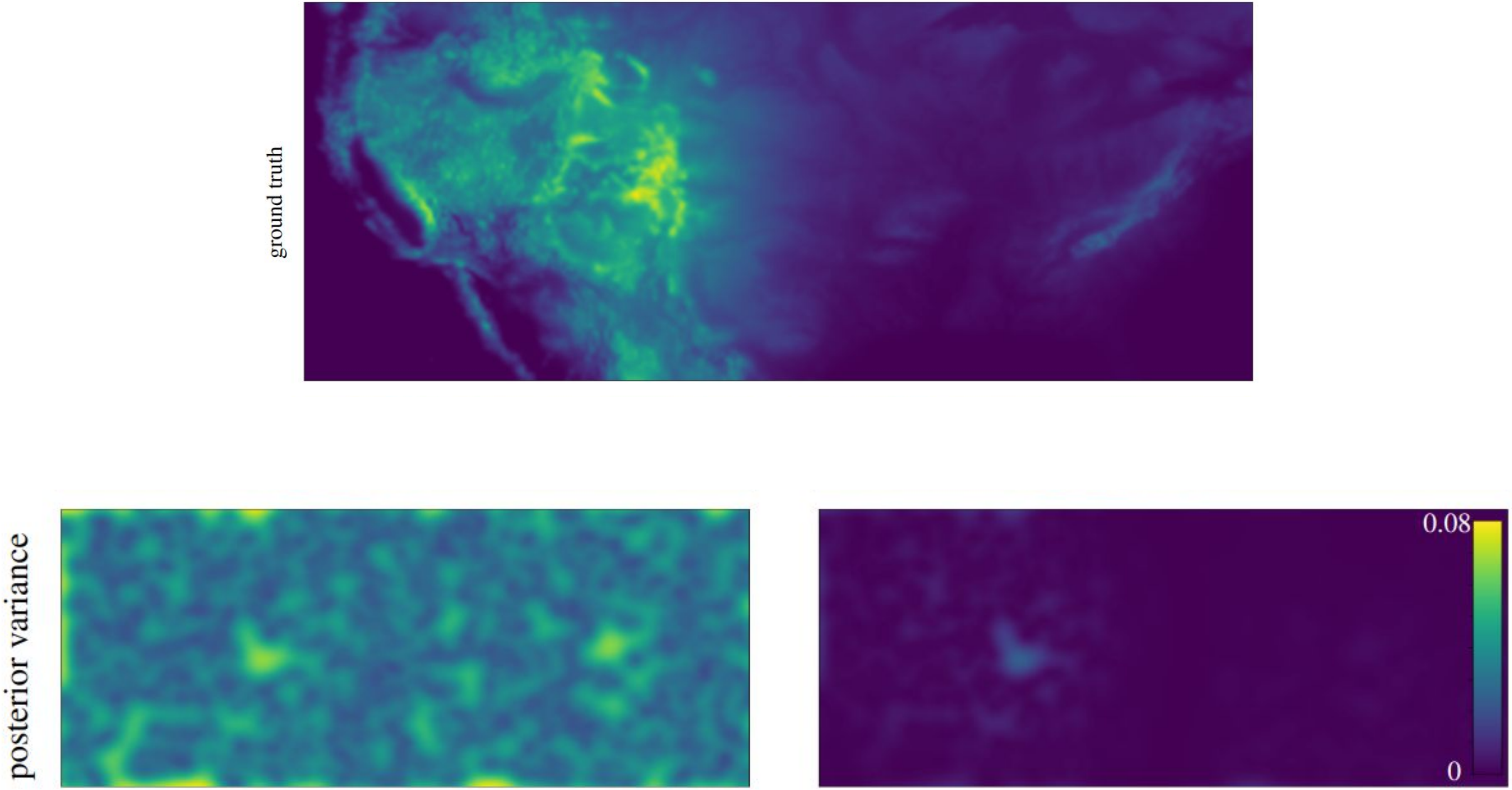
$$k_{non}(\mathbf{x}_1, \mathbf{x}_2) = f(\mathbf{x}_1) f(\mathbf{x}_2)\, k_{stat}(\mathbf{x}_1, \mathbf{x}_2)$$

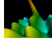
$$f(x) = \sum_i^N \alpha_i \beta(\mathbf{x}_i, \mathbf{x}; w)$$

$$\beta(\mathbf{x}_i, \mathbf{x}; w) = \exp\left[-||\mathbf{x}_i - \mathbf{x}||\, w^2\right]$$

w... width parameter

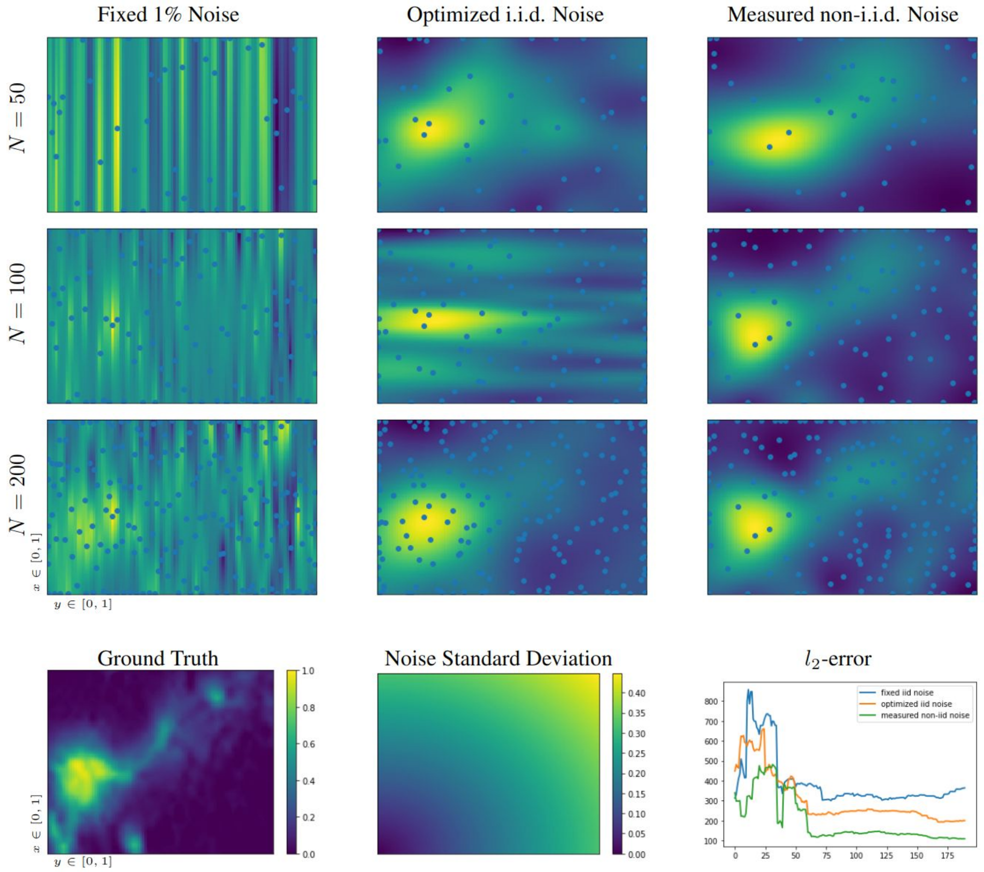



ground truth
posterior variance
0.08
0



Accurate treatment of noise is vital for a successful AE.

$$p(\mathbf{y}|\mathbf{f}) = \frac{1}{\sqrt{(2\pi)^{\dim}|\mathbf{V}|}} \exp\left[-\frac{1}{2}(\mathbf{y}-\mathbf{f})^T\boxed{\mathbf{V}}^{-1}(\mathbf{y}-\mathbf{f})\right]$$



(1) Accurate uncertainty quantification is not achieved with standard GPs.

(2) Standard GPs don’t adhere to physical constraints.

(3) Standard GPs might not have (optimal) solutions that satisfy optimization constraints.

(4) GPs don’t scale well.

(5) Advanced GPs lead to tough optimization problems.

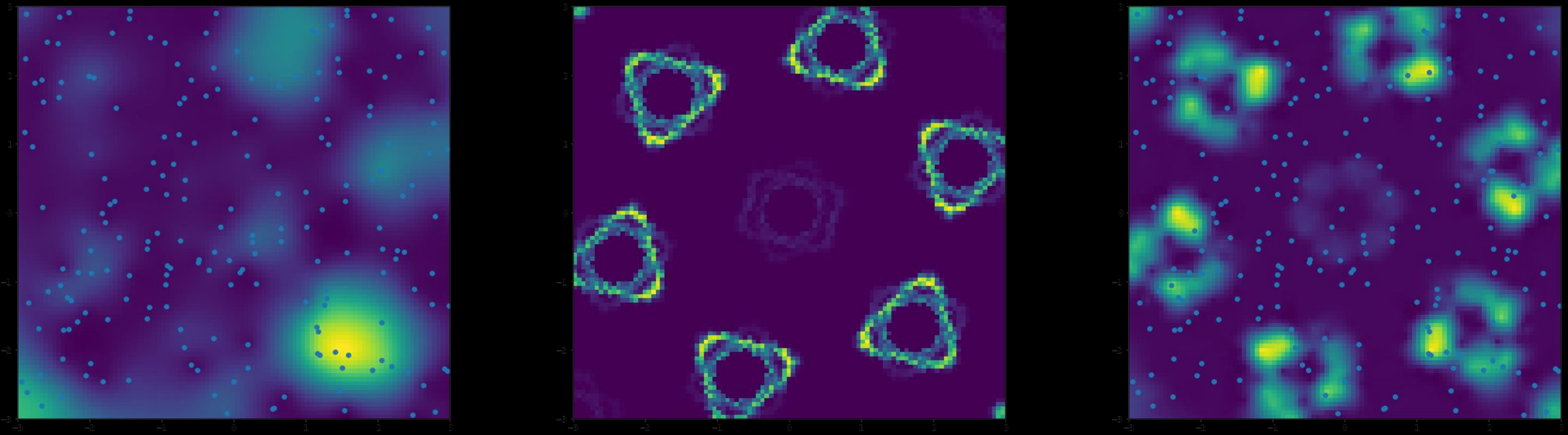
(6) Standard GPs don’t work on non-linear spaces.

(7) Sometimes Gaussianity just doesn’t cut it.

Kernels can constrain the function space (RKHS) to only allow for physically-viable solutions

Facilities: Thales @ ILL, Grenoble, France | **Technique:** Neutron Scattering | **Collabs:** Boehm, Mutti, Weber | **Achievement:** Constraining the RKHS to 6-fold symmetric functions

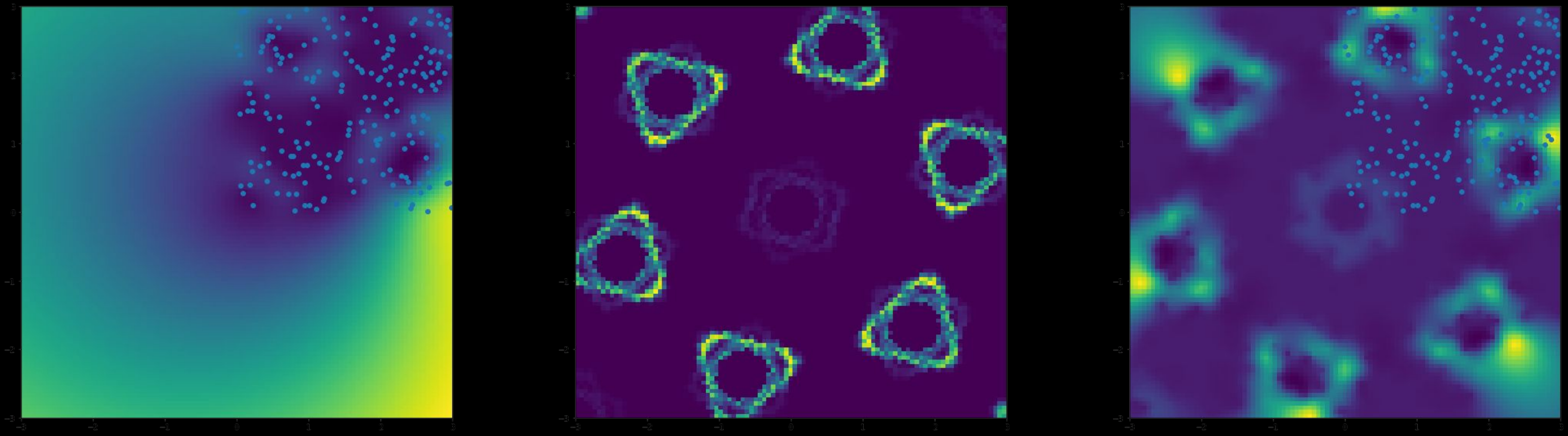
$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{36} \sum_{\phi} \sum_{\theta} \tilde{k}(\mathcal{R}_\phi \mathbf{x}_i, \mathcal{R}_\theta \mathbf{x}_j)$$



Kernels can constrain the function space (RKHS) to only allow for physically-viable solutions

Facilities: Thales @ ILL, Grenoble, France | **Technique:** Neutron Scattering | **Collabs:** Boehm, Mutti, Weber | **Achievement:** Constraining the RKHS to 6-fold symmetric functions

$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{36} \sum_{\phi} \sum_{\theta} \tilde{k}(\mathcal{R}_\phi \mathbf{x}_i, \mathcal{R}_\theta \mathbf{x}_j)$$

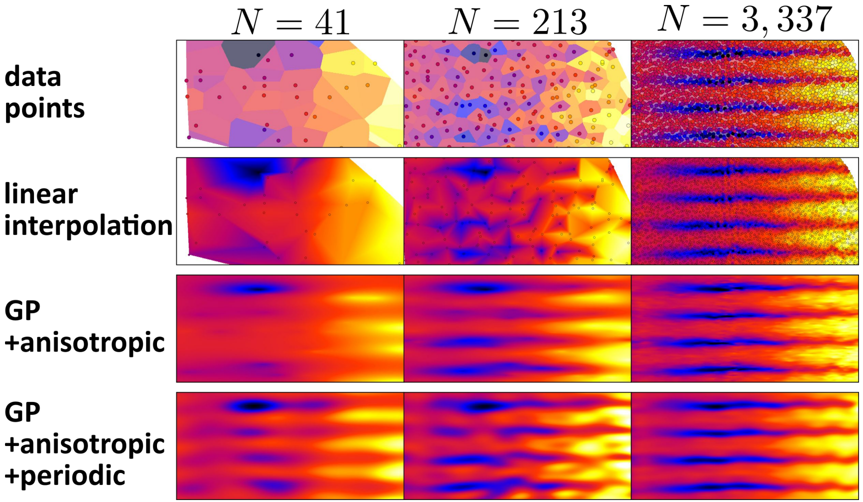


gpCAM

NEUTRONS FOR SOCIETY

Physics Knowledge in the Form of Periodicity for X-Ray Scattering

Facility: NIST, CFN, NSLS II | **Technique:** SAXS,GISAXS | **Collabs**: Yager, Fukuto, Seppala | **Achievement:** Use of non-stationary kernels to learn and exploit local characteristics



(1) Accurate uncertainty quantification is not achieved with standard GPs.

(2) Standard GPs don't adhere to physical constraints.

(3) Standard GPs might not have (optimal) solutions that satisfy optimization constraints.

(4) GPs don't scale well.

(5) Advanced GPs lead to tough optimization problems.

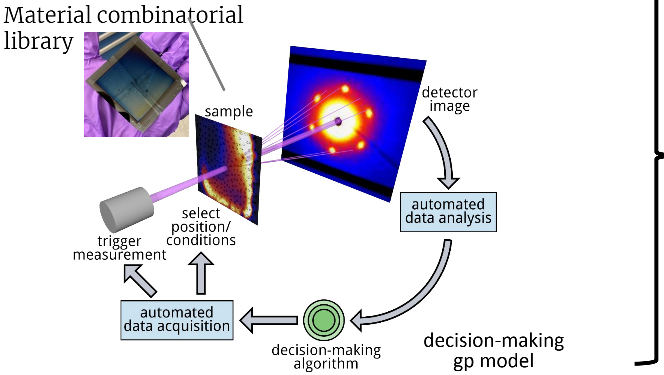
(6) Standard GPs don't work on non-linear spaces.

(7) Sometimes Gaussianity just doesn't cut it.

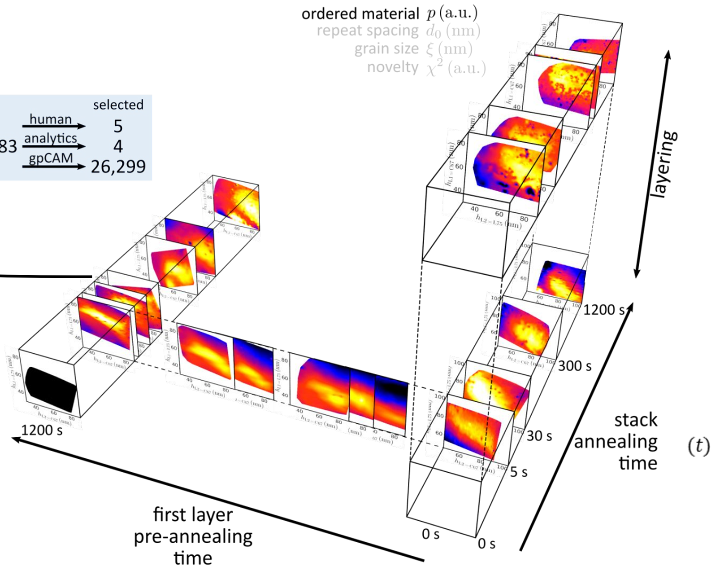
Kernels can do even more...

Autonomous exploration of multidimensional material state-spaces underlying self-assembly of copolymer mixtures

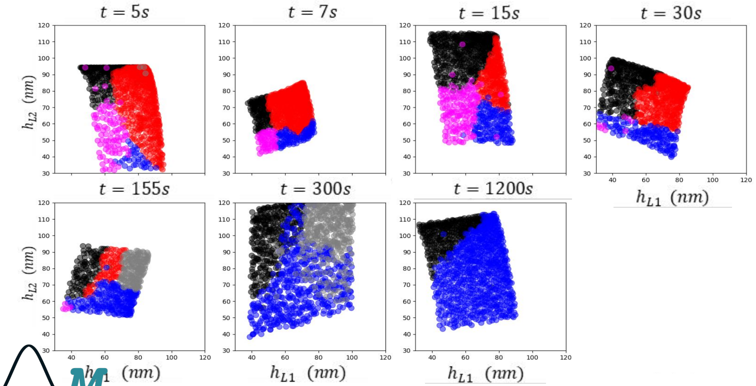
Facility: CFN & NSLS II at BNL | **Technique:** small angle x-ray scattering | **Collabs:** Russel, Fukuto, Yager | **Achievement:** Optimal mapping of a 5-dim. material state-space



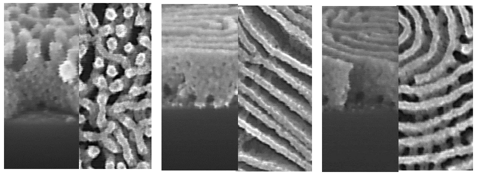
	total		selected
parameters:	≥12	human →	5
signals:	1,023,183	analytics →	4
points:	∞	gpCAM →	26,299



Structural classification using semi-supervised clustering scheme



Identification of non-native structural motifs



Center for Functional Nanomaterials
Brookhaven National Laboratory
Brookhaven National Laboratory

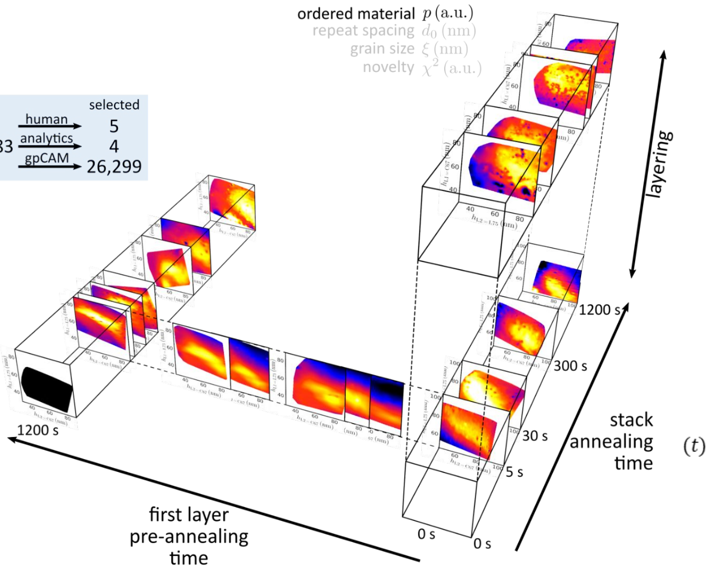
Autonomous exploration of multidimensional material state-spaces underlying self-assembly of copolymer mixtures

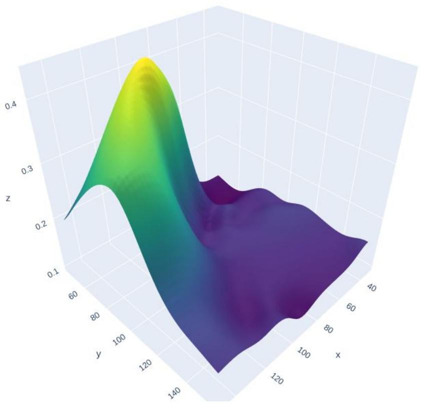
Facility: CFN & NSLS II at BNL | **Technique:** small angle x-ray scattering | **Collabs:** Russel, Fukuto, Yager | **Achievement:** Optimal mapping of a 5-dim. material state-space

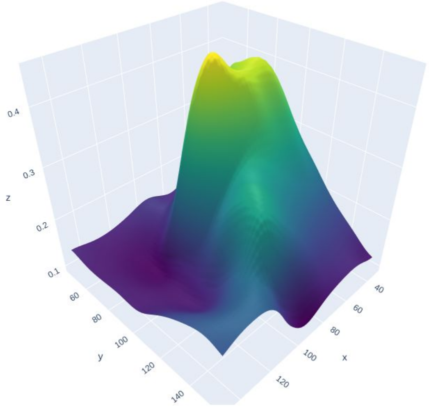
What do we know about the physics:

1. Space distances are isotropic in 2 directions and anisotropic in time direction.
2. The model function is in [0,1].
3. The sum across all 8 tasks at any position in the input space is 1.
4. 1st order differentiability in all directions

	total		selected
parameters:	≥12	human →	5
signals:	1,023,183	analytics →	4
points:	∞	gpCAM →	26,299







$$k(\mathbf{x}_1, \mathbf{x}_2) =$$
$$\sigma_s^2\, k_m(\mathbf{x}_1^{\{0,1\}}, \mathbf{x}_2^{\{0,1\}})$$
$$k_t(x_1^3, x_2^3) + f(task_1) f(task_2)$$

Center for Functional Nanomaterials
Brookhaven National Laboratory
Brookhaven National Laboratory

gpCAM

(1) Accurate uncertainty quantification is not achieved with standard GPs. ✓

(2) Standard GPs don’t adhere to physical constraints. ✓

(3) Standard GPs might not have (optimal) solutions that satisfy optimization constraints. ✓

(4) GPs don’t scale well.

(5) Advanced GPs lead to tough optimization problems.

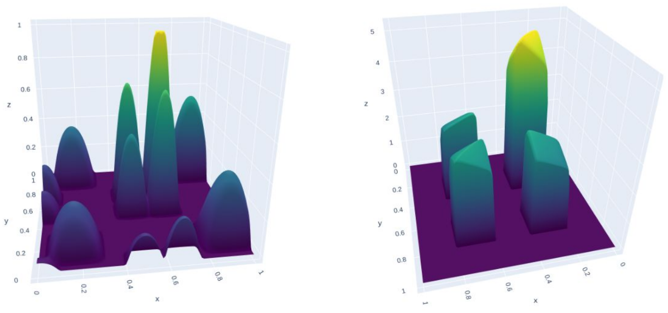
(6) Standard GPs don’t work on non-linear spaces.

(7) Sometimes Gaussianity just doesn’t cut it.

gp2Scale: Exact Gaussian Processes on Millions of Data Points

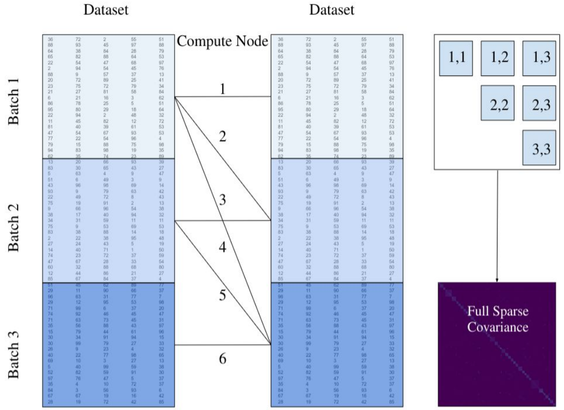
— let kernels discover, not induce, sparsity

Building Block 1: Ultra-Flexible, Compactly Supported and Non-Stationary Kernels



$$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$$

Building Block 2: HPC Distributed Covariance Computation



Building Block 3: MCMC Constrained Training

$$argmax_h \ln(L)$$

$$\text{subject to } s < \text{ sparsity requirement}$$

$$argmax_{h,s}\Big(\ln(L) + (1-s)\ln(L)\Big)$$

(1) Accurate uncertainty quantification is not achieved with standard GPs.

(2) Standard GPs don't adhere to physical constraints.

(3) Standard GPs might not have (optimal) solutions that satisfy optimization constraints.

(4) GPs don't scale well.

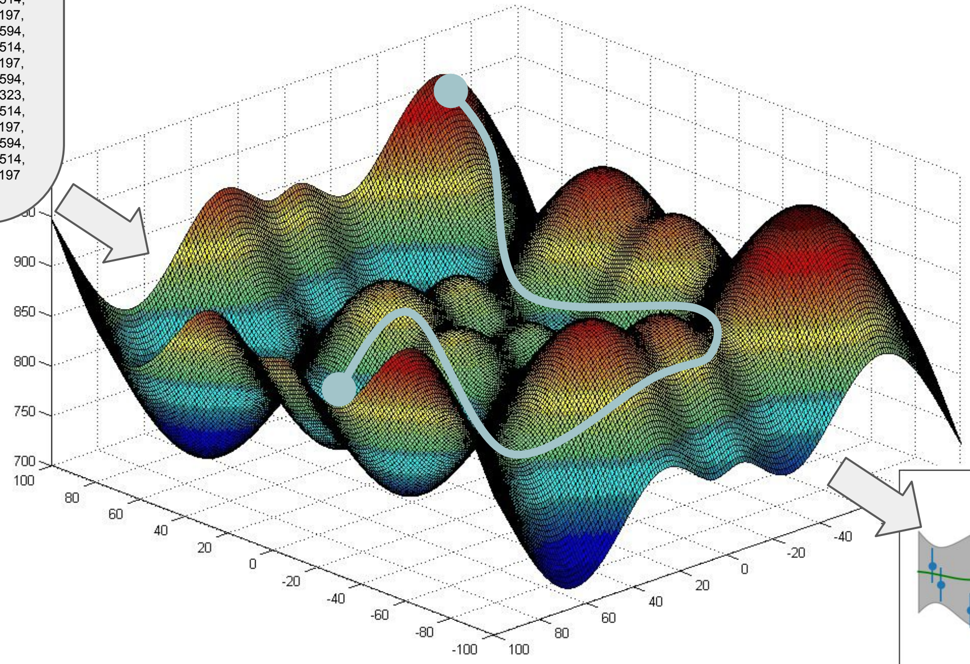
(5) Advanced GPs lead to tough optimization problems.

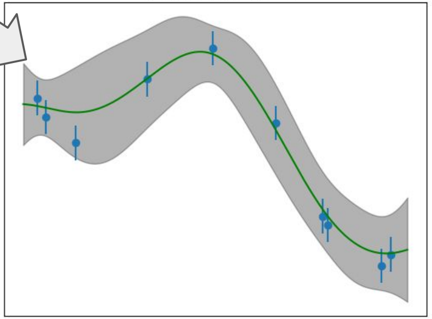
(6) Standard GPs don't work on non-linear spaces.

(7) Sometimes Gaussianity just doesn't cut it.

0.26436384, 0.21300795, 0.19834864, ..., 0.20183792, 0.82454492,0.81746336,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.09019633, 0.03045269, 0.55291218, ..., 0.66801905, 0.75265345,1.64352323,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.09019633, 0.03045269, 0.55291218, ..., 0.66801905, 0.75265345,1.64352323,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.09019633, 0.03045269, 0.55291218, ..., 0.66801905, 0.75265345,1.64352323,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197

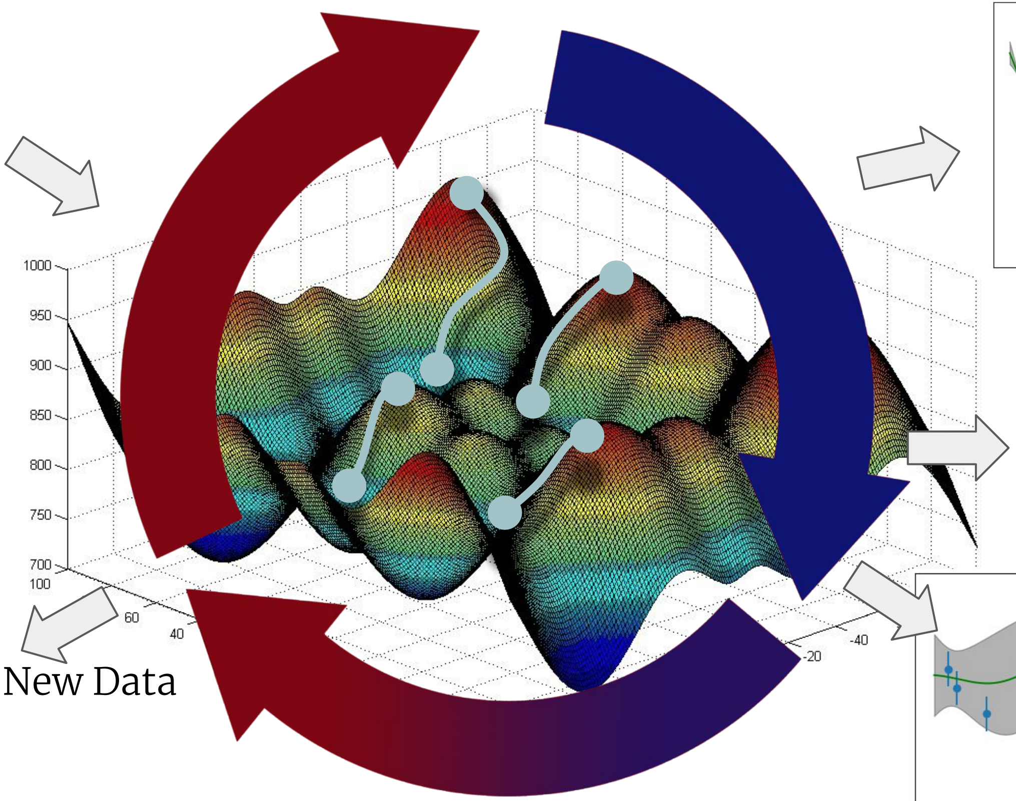
The Traditional Training/Optimization Workflow needs a Large Number of Function Evaluations and Blocks the Main Thread

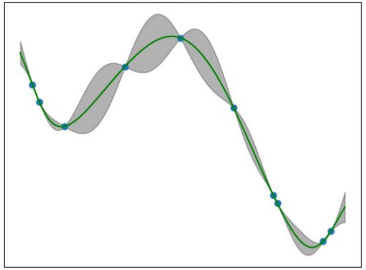


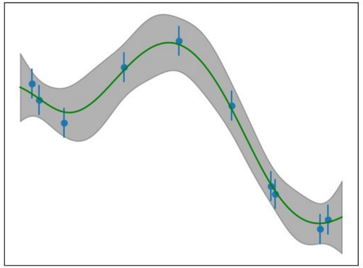


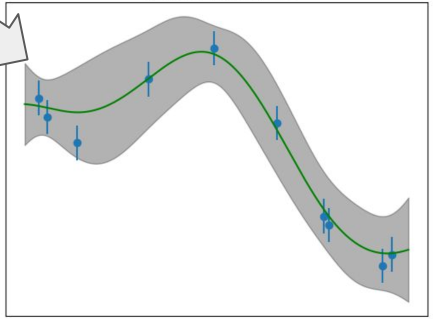
0.26436384, 0.21300795, 0.19834864, ..., 0.20183792, 0.82454492,0.81746336,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.09019633, 0.03045269, 0.55291218, ..., 0.66801905, 0.75265345,1.64352323,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.09019633, 0.03045269, 0.55291218, ..., 0.66801905, 0.75265345,1.64352323,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.09019633, 0.03045269, 0.55291218, ..., 0.66801905, 0.75265345,1.64352323,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197,
0.43558082, 0.95638452, 0.99928695, ..., 0.63067478, 0.38601846,0.52014594,
0.68064155, 0.66793227, 0.02274104, ..., 0.37098925, 0.66477699,0.71282514,
0.13523289, 0.85643443, 0.43357488, ..., 0.71829634, 0.98986933,0.60671197

Minimizing Number of Function Evaluations: Asynchronous Distributed Training









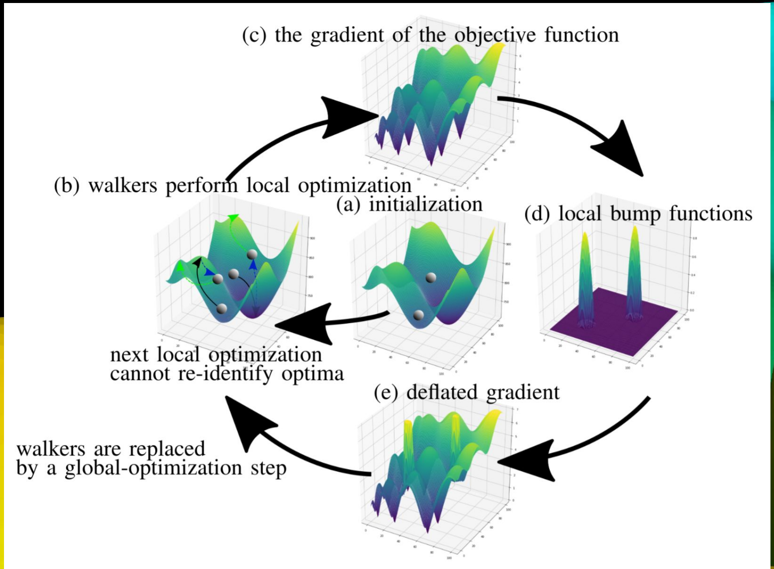
Kill/Restart/Ingest New Data

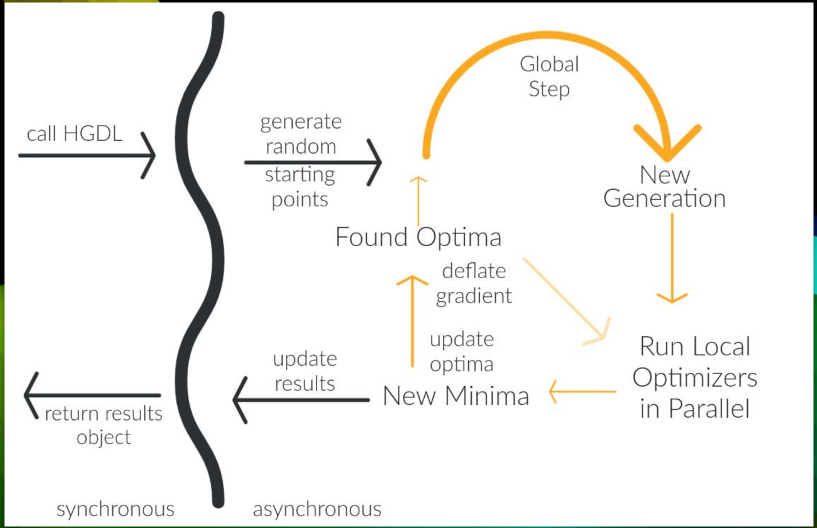
Optimization of the Log-Likelihood with HGDL

Using High Performance Asynchronous Distributed Optimization for Robust and Efficient GP Training.

$$\log(L(\mathcal{D};\phi,\mu)) = -\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu}(\mathbf{x}))^T(\mathbf{K}(\phi)+\mathbf{I}_e)^{-1}(\mathbf{y}-\boldsymbol{\mu}(\mathbf{x})) - \frac{1}{2}\log(|\mathbf{K}(\phi)+\mathbf{I}_e|)$$

Optimization Challenges: Ill-Posedness, Non-Uniqueness, Costly Func. Evals + High-Dim., Blocking Execution





HGDL yields:

1. a set of unique solutions
2. HPC readiness of training and prediction
3. asynchronous training
4. Measure for uniqueness of solutions

(1) Accurate uncertainty quantification is not achieved with standard GPs.

(2) Standard GPs don’t adhere to physical constraints.

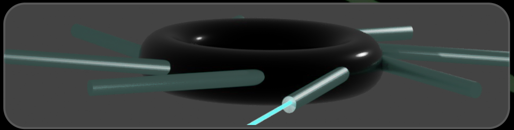
(3) Standard GPs might not have (optimal) solutions that satisfy optimization constraints.

(4) GPs don’t scale well.

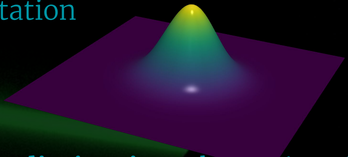
(5) Advanced GPs lead to tough optimization problems.

(6) Standard GPs don’t work on non-linear spaces.

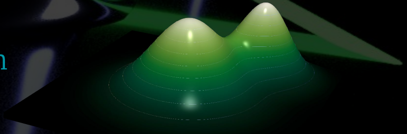
(7) Sometimes Gaussianity just doesn’t cut it.



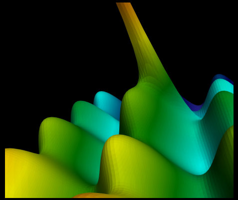
Motivation: Stochastic Modeling and Autonomous Experimentation



Preliminaries: The Basics of Gaussian-Processes and Autonomous Data Acquisition

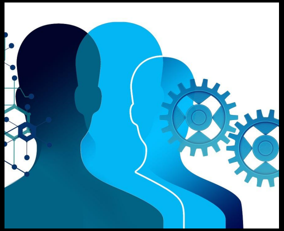


Challenges: Approximation Accuracy, UQ, Domain Awareness, and Scalability



Advancements: Flexible Non-Stationary and Compactly Supported Kernel Designs

Synergy: Community and Software

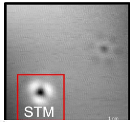


Synergy:
Community and Software

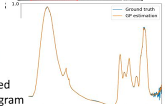
gpCAM is a CAMERA project that is the result of a broad collaboration between many institutions and facilities.

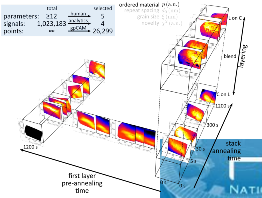
BERKELEY LAB
Bringing Science Solutions to the World

MOLECULAR FOUNDRY



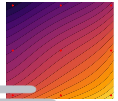
BSISB
Berkeley Synchrotron Infrared Structural Biology Imaging Program





Brookhaven National Laboratory

NSLS-II
NATIONAL SYNCHROTRON LIGHT SOURCE II

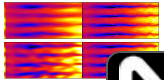


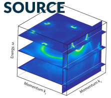
Center for Functional Nanomaterials
Brookhaven National Laboratory

ALS
ADVANCED LIGHT SOURCE

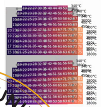
COMPUTATIONAL RESEARCH DIVISION

JGI
JOINT GENOME INSTITUTE



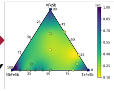


NIST
National Institute of Standards and Technology
U.S. Department of Commerce

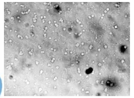


AFRL
AIR FORCE RESEARCH LABORATORY
LEAD | DISCOVER | DEVELOP | DELIVER

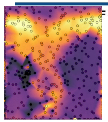
SLAC
NATIONAL ACCELERATOR LABORATORY



Agile BioFoundry

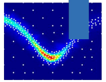


CARNEGIE MELLON UNIVERSITY
PITTSBURGH PENNSYLVANIA 1900

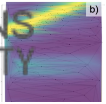


COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

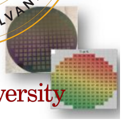
TUM



ill
NEUTRONS FOR SOCIETY



Berkeley
UNIVERSITY OF CALIFORNIA



Stony Brook University

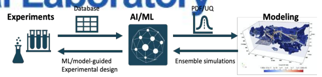
UNIVERSITY OF BIRMINGHAM

UNIVERSITÄT BAYREUTH

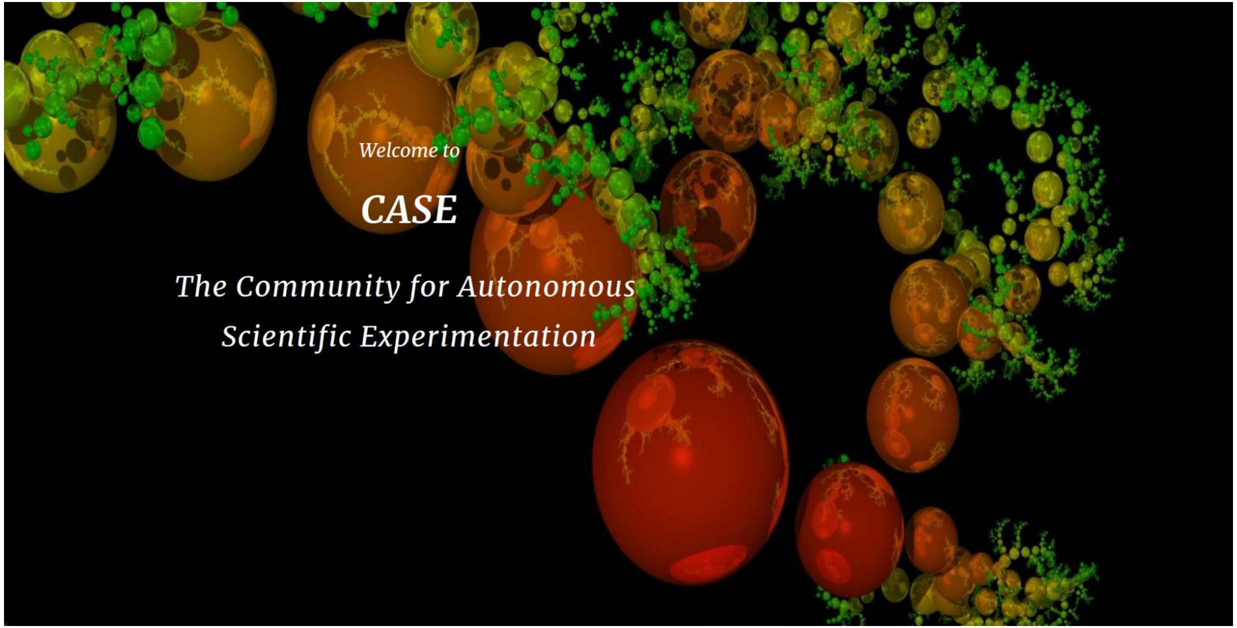
CHESS
CORNELL HIGH ENERGY SYNCHROTRON SOURCE

Argonne
NATIONAL LABORATORY

Lawrence Livermore National Laboratory



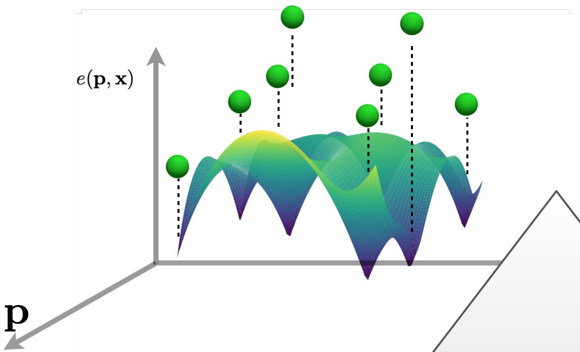
CASE: A Community for Autonomous Scientific Experimentation



Established in 2021
217+ members

https://autonomous-discovery.lbl.gov/

The product is three APIs



gpcAM
pip install gpcam

fvGP: A flexible multi-task Gaussian process tool
pip install fvgp

HGDL: Asynchronous Distributed Optimizer
pip install hgdl

```
from gpcam.autonomous_experimenter import AutonomousExperimenterGP
from instrument import instrument
import numpy as np

parameters = np.array([[3.0,45.8],
                       [4.0,47.0]])
init_hyperparameters = np.array([1,1,1])

hyperparameter_bounds =  np.array([[0.01,100],[0.01,100.0],[0.01,100]])

my_ae = AutonomousExperimenterGP(parameters, instrument, init_hyperparameters,
                                 hyperparameter_bounds,  init_dataset_size=10)

my_ae.train()

my_ae.go()
```

```
def instrument(data):
    for entry in data:
        entry["value"] = np.sin(np.linalg.norm(entry["position"]))
    return data
```

gpcam.lbl.gov

62000 downloads
21 facilities and counting
A GUI, called Tsuchinoko, is being developed by Ron Pandolfi
pip install tsuchinoko

CAMERA

Applied Math

U.S. DEPARTMENT OF ENERGY

Office of Science

Advanced Gaussian Process Function Approximation for Uncertainty Quantification and Autonomous Experimentation

Focus: Kernel Designs and Scalability

Marcus M. Noack

Lawrence Berkeley National Lab
MarcusNoack@lbl.gov