中山大學
SUN YAT-SEN UNIVERSITY

# Machine Learning in Nuclear Theory: Bayesian Inference and Surrogate Emulators

**张振 (Zhen Zhang)**
中法核工程与技术学院 (IFCEN)，中山大学 (SYSU)

第四届全国核物理及核数据中的机器学习应用研讨会，
衡阳，10.31-11.2， 2025

# Outline

- Introduction
- Uncertainty quantification and Bayesian inference
  - Bayesian parameter estimation
  - Bayesian model averaging and selection
- Surrogate emulators
  - Gaussian process
  - Reduced basis method & Eigenvector continuation
  - Parametric matrix model
- Summary

# Machine learning in nuclear theory

1. **Inverse modeling and UQ**

   Infer model parameters or physical information from data
   - Bayesian parameter estimation
   - Bayesian model averaging/selection

2. **Computational acceleration**

   Replace or approximate expensive calculations
   - Gaussian process
   - Eigenvector continuation

3. **Nuclear property prediction**

   Predict unmeasured observables
   - Mass
   - Radius

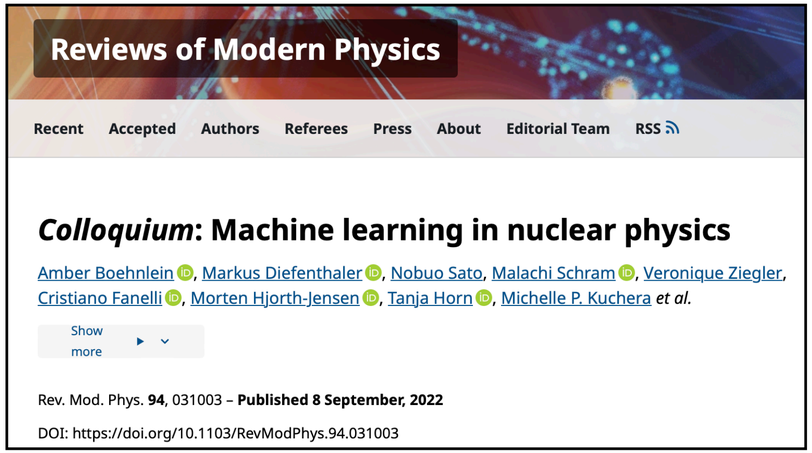
4. **Structure/mechanism discovery**

   Identify latent physical patterns
   - Phase-transition
   - Clustering

5. **Model construction**

   Learn or refine models/functionals
   - Neural-network quantum state
   - Symbolic machine learning

✦In this talk, I will focus on **emulator+Bayesian inference**


Reviews of Modern Physics

Recent Accepted Authors Referees Press About Editorial Team RSS

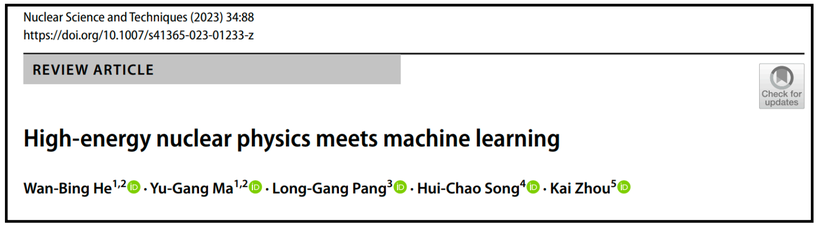
***Colloquium*: Machine learning in nuclear physics**

Amber Boehnlein, Markus Diefenthaler, Nobuo Sato, Malachi Schram, Veronique Ziegler, Cristiano Fanelli, Morten Hjorth-Jensen, Tanja Horn, Michelle P. Kuchera *et al.*

Show more

Rev. Mod. Phys. **94**, 031003 – **Published 8 September, 2022**

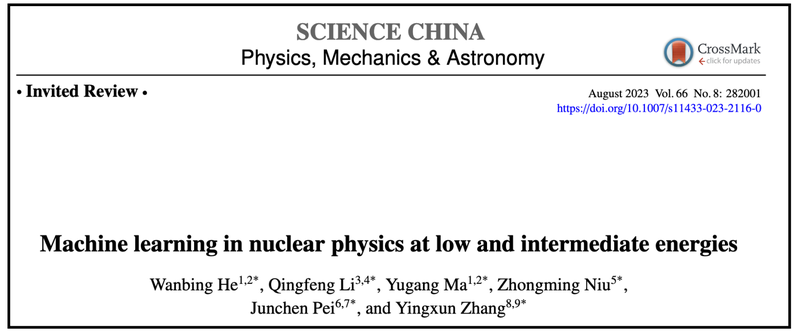
DOI: https://doi.org/10.1103/RevModPhys.94.031003


Nuclear Science and Techniques (2023) 34:88
https://doi.org/10.1007/s41365-023-01233-z

REVIEW ARTICLE

**High-energy nuclear physics meets machine learning**

Wan-Bing He[1,2] · Yu-Gang Ma[1,2] · Long-Gang Pang[3] · Hui-Chao Song[4] · Kai Zhou[5]


SCIENCE CHINA
Physics, Mechanics & Astronomy

• Invited Review •

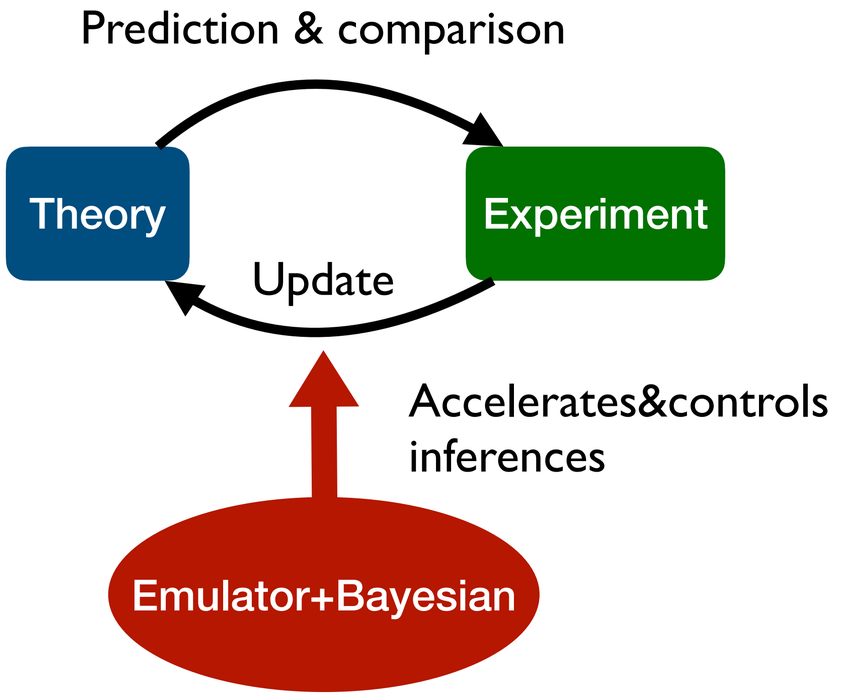
August 2023 Vol. 66 No. 8: 282001
https://doi.org/10.1007/s11433-023-2116-0

**Machine learning in nuclear physics at low and intermediate energies**

Wanbing He[1,2*], Qingfeng Li[3,4*], Yugang Ma[1,2*], Zhongming Niu[5*], Junchen Pei[6,7*], and Yingxun Zhang[8,9*]

# Emulator + Bayesian: an external accelerator for nuclear theory

- Traditional loop:
  Build theory → compute observables → compare with data → refine model
  - computationally costly, slow feedback.
- Emulator + Bayesian provides an external, plug-in-style acceleration:
  - Emulators replace computational expensive model evaluations.
  - Bayesian inference quantifies uncertainty and extracts physics from data.
  - Together they accelerate theory-data iteration **without changing the underlying physical model.**



# Uncertainty quantification and Bayesian inference

- Statistical and systematic uncertainties.
- UQ with Bayesian approach
- Tools: (Python package)
  Surmise (PCA+GP+MCMC)
  PyMultiNest (Nest sampling)
  Pymc
  ……

# Inverse problems in nuclear physics

Lattice QCD

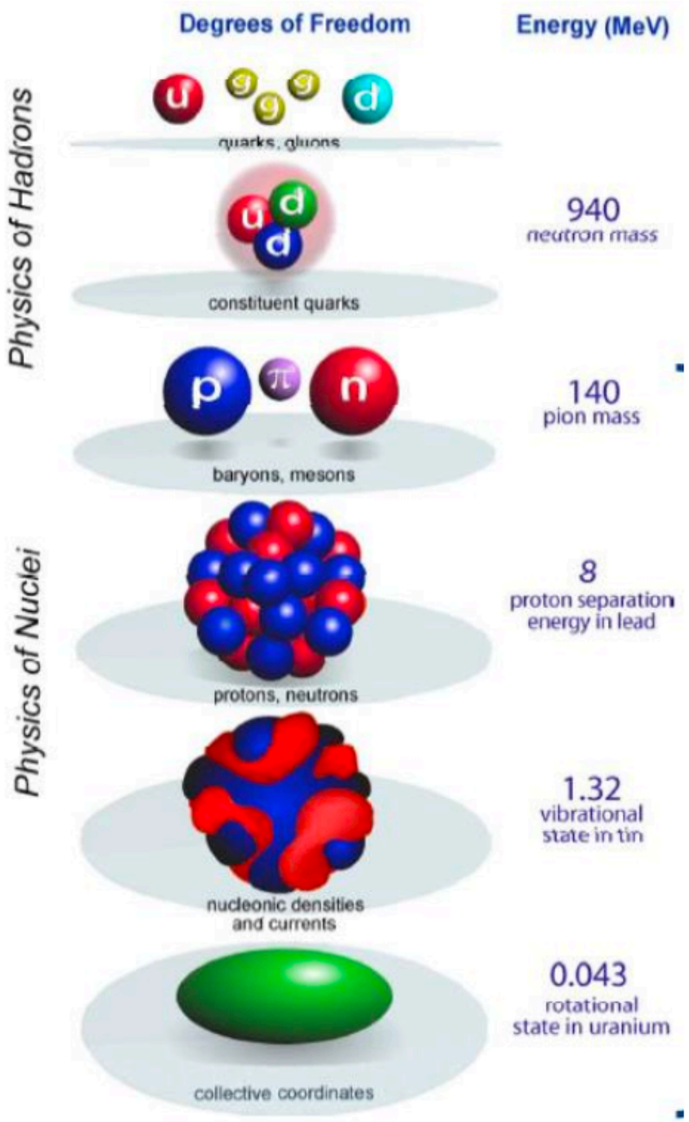
Quark model

Ab initio

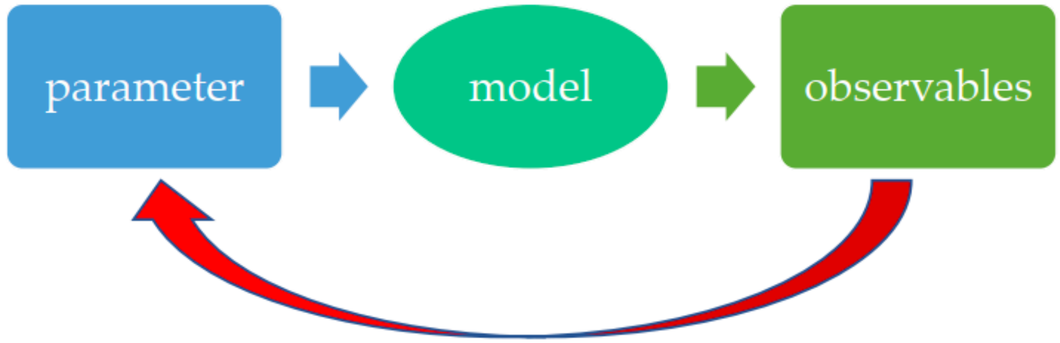
Configuration interaction

Energy density functional theory

collective models

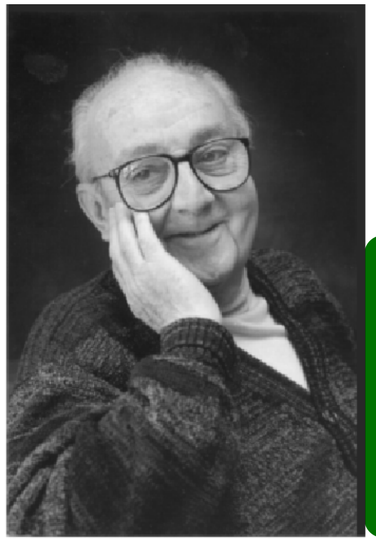



- Owing to the lack of a well-settled ab initio starting point, a considerable number of effective theories or models have been developed with parameters determined by fitting empirical knowledge or experimental data .




Inverse problem

# Uncertainty quantification



**George E.P. Box:** Box and Draper, 1987 *Empirical Model Building and Response Surfaces*

- All models are wrong, but some are useful.
- The practical question is how wrong do they have to be to not be useful.

**PRA editorial:** *Phys. Rev. A 83, 040001 (2011)*

There is a broad class of papers where estimates of theoretical uncertainties can and should be made. Papers presenting the results of theoretical calculations are expected to **include uncertainty estimates for the calculations whenever practicable,…**

# Sources of uncertainty

- **Statistical error**
  - Uncertainty propagation:


  - Numerical uncertainty ( e.g., Monte Carlo)

✦ **Systematic error** from imperfect modeling.

  - Inter-model uncertainties and model dependence.
  - Could be estimated by compare different models.

Dobaczewski, Nazarewicz and Reinhard, JPG41 (2014) 074001

Large statistical error:

Large systematic error:

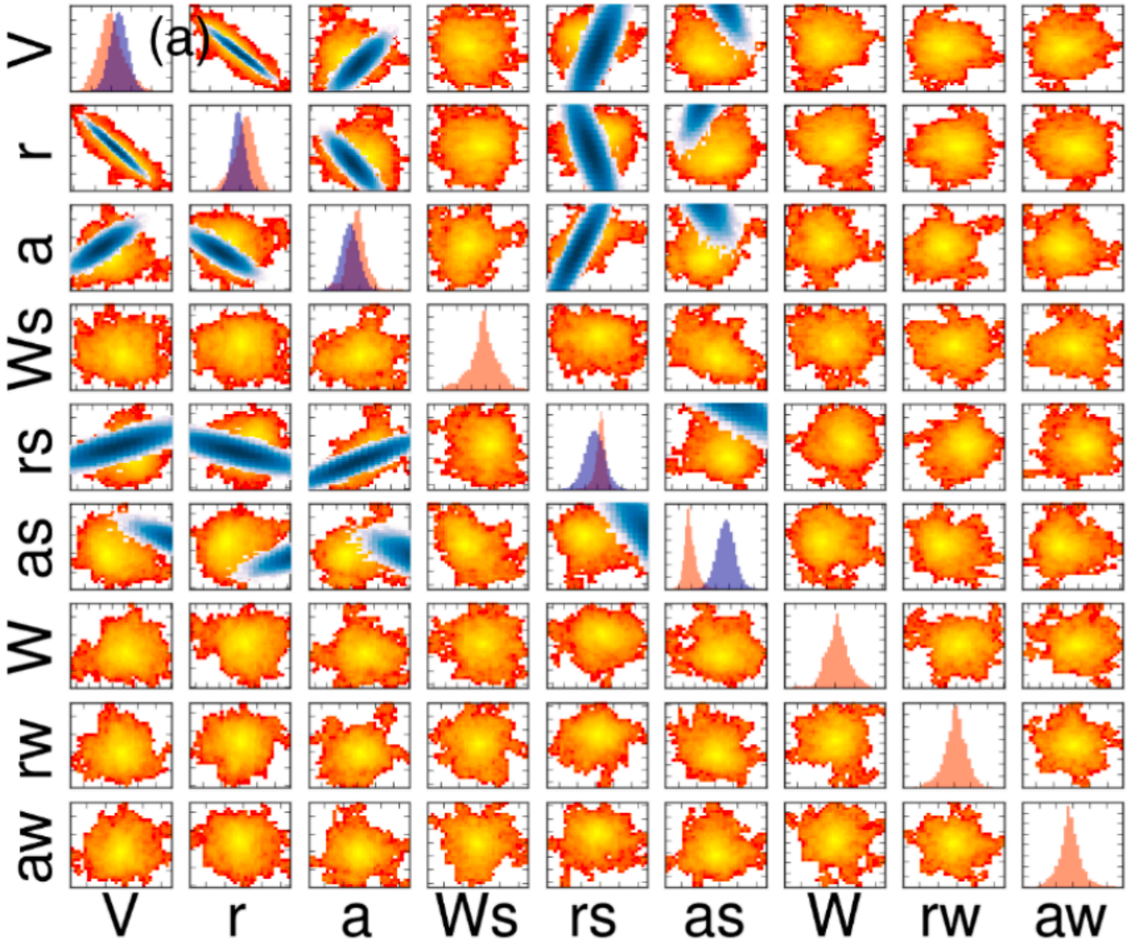
# Why Bayesian?

Frequentist vs Bayesian

Frequentist
- Local minimum or global minimum?
- Errors and correlations are determined by $\partial^2\chi^2/\partial p_i \partial p_j$ at minimum point.
  (reliable for a well-constrained model)

Bayesian
- More flexible in exploring parameter space.
- More reliable uncertainties and correlations.
- Prior knowledge can be easily taken into account.

King et al., PRL 122, 232502 (2019).




Blue: Frequentist approach
Orange: Bayesian based on MCMC

# Bayes' theorem

- Conditional probability

$P(A \mid B)$ : probability of event A occurring, given event B

- Product rule:

$$P(A, B) = P(A \mid B)P(B) = P(B \mid A)P(A)$$

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}, \qquad P(B) = P(B \mid A)P(A) + P\left(B \mid \overline{A}\right) P\left(\overline{A}\right)$$

- Bayes' rule (discrete case):

$$P(A_i \mid B) = \frac{P(B \mid A_i)P(A_i)}{\sum_i P(B \mid A_i)P(A_i)}$$

- Bayes' rule (continuous case):

$$p(A \mid B) = \frac{p(B \mid A)p(A)}{\int p(B \mid A)p(A)dA}$$

Thomas Bayes
(1702-1761)

# Bayesian parameter estimation

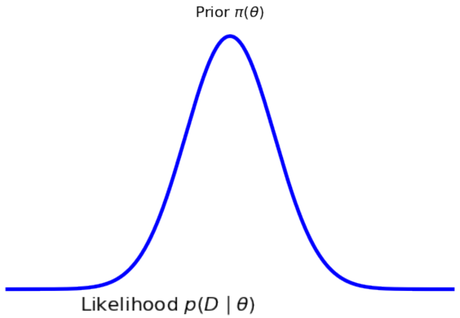
1. **Prior** $\pi(\boldsymbol{\theta})$ : knowledge or assumptions about the parameters before seeing data.
   - Usually Uniform or Gaussian with large width
2. **Likelihood function** $L(\boldsymbol{y}\,|\,\boldsymbol{\theta})$ : probability of observed data $\boldsymbol{y}$ given parameters
   - Usually Gaussian: $\propto \exp(-\chi^2/2)$, with $\chi^2 = \sum_{\mathrm{i}} (\mathrm{y_i} - \mathrm{y_i^{th}})^2/\sigma_{\mathrm{i}}^2$
3. Apply Bayes' rule:

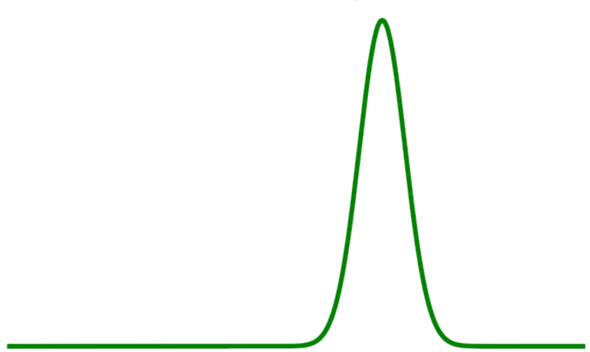
$$p(\boldsymbol{\theta}\,|\,\boldsymbol{y}) = \frac{L(\boldsymbol{y}\,|\,\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{p(\boldsymbol{y})} \propto L(\boldsymbol{y}\,|\,\boldsymbol{\theta})\pi(\boldsymbol{\theta})$$

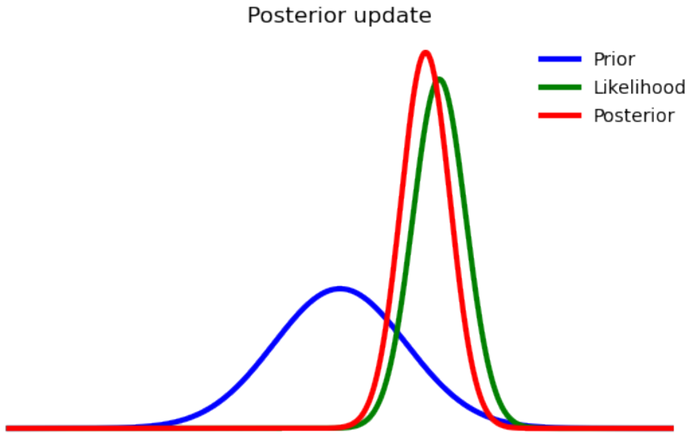
Posterior $\propto$ Prior $\times$ Likelihood

   - **Posterior** $p(\boldsymbol{\theta}\,|\,\boldsymbol{y})$: probability distribution of parameters given observed data
   - **Evidence** $p(\boldsymbol{y}) = \int L(\boldsymbol{y}\,|\,\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$:
4. Make prediction for observable $O$:

$$p(O\,|\,\boldsymbol{y}) = \int L(O\,|\,\boldsymbol{\theta})p(\boldsymbol{\theta}\,|\,\boldsymbol{y})\mathrm{d}\boldsymbol{\theta}$$

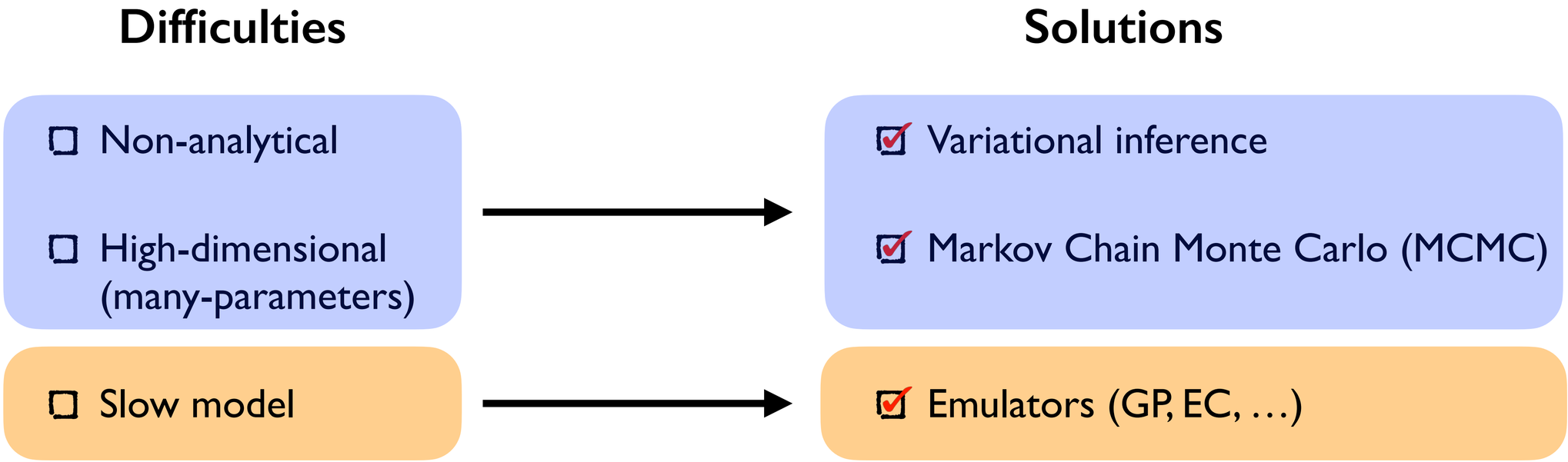


# Technique difficulties

$$p(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\boldsymbol{y} \mid \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$$



# Metropolis-Hastings Algorithm

- **Metropolis-Hastings algorithm** (Metropolis et al. 1953, Hastings 1970):

  Construct a Markov Chain for Θ by employing an auxiliary distribution that is easy to sample from.

- **Steps:**

  1. Choose a starting value $\theta^{(i)}$ $(i = 0)$;
  2. Produce a candidate value $\theta^*$ according to a **proposal distribution** $q(\theta^*|\theta^{(i)})$;
  3. Calculate the **acceptance probability**

$$a(\theta^{(i)}, \theta^*) = \min\left\{1, \frac{p(\theta^* \mid \mathcal{D}) q(\theta^{(i)} \mid \theta^*)}{p(\theta^{(i)} \mid \mathcal{D}) q(\theta^* \mid \theta^{(i)})}\right\}$$

  Ratio independent of the normalization factor

  4. Generate $u \sim U(0,1)$,
     If u < a, let $\theta^{(i+1)} = \theta^*$, else $\theta^{(i+1)} = \theta^i$ .
  5. i=i+1, go to the 2nd step.

- The Markov Chain converges to the stationary distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$

# Random-walk Metropolis-Hastings

- Take $q(\theta^*|\theta^{(i)})$ to be **symmetric**, i.e.,

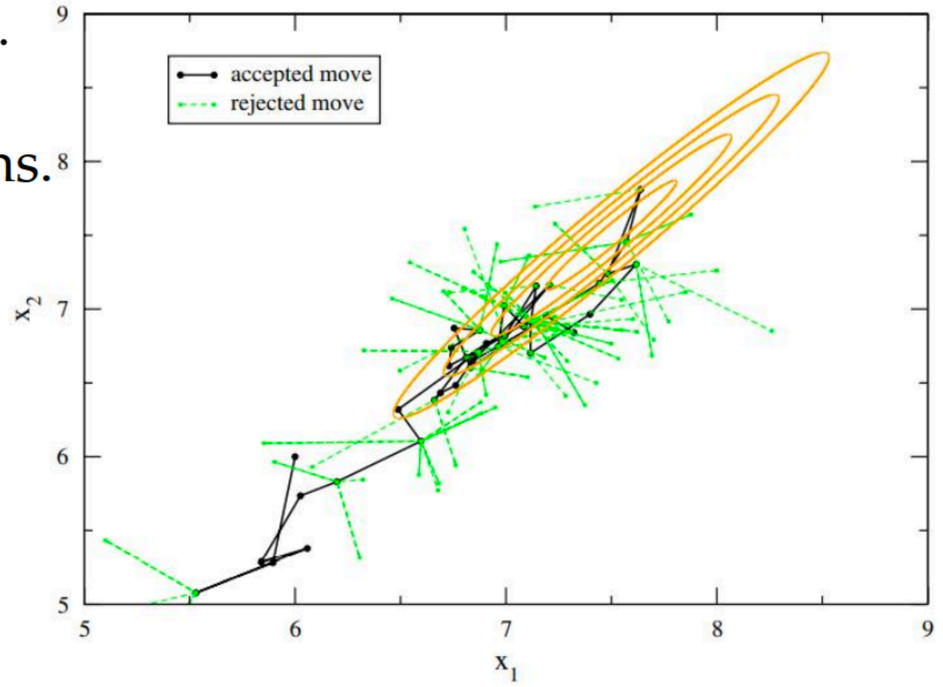
$$q(\theta^* \mid \theta^{(i)}) \;=\; q(\theta^{(i)} \mid \theta^*) \;= q(|\,\theta^* - \theta^{(i)}\,|).$$

  Popular choices are (multivariate) Gaussians or t-distributions.

- Draw $\epsilon \sim q$, $\theta^* = \theta^{(i)} + \epsilon$
- **Acceptance probability**

$$a(\theta^{(i)}, \theta^*) = \min\left\{1, \frac{p(\theta^* \mid \mathcal{D})}{p(\theta^{(i)} \mid \mathcal{D})}\right\}$$

- **Scale parameter** controls the acceptance rates.




U. von Toussaint, Rev. Mod. Phys. 83. 943 (2011)

- **Q: convergency? Autocorrelation?**

Interactive MCMC Sampling Visualizer by Chi Feng

https://chi-feng.github.io/mcmc-demo

# Uncertainty quantification & correlation analysis

- 边缘分布（marginal distribution）

$$p\left(\theta_i \mid \boldsymbol{y}\right) = \frac{\int p(\boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d} \prod_{j \neq i} \theta_j}{\int p(\boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d} \prod_j \theta_j}$$

- 可信区间（credible interval）

$$P(a \leq \theta_i \leq b) = \int_a^b p(\theta_i \mid \mathcal{D}) dx_i = 1 - \alpha$$

$$P(\theta_i < a) = P(\theta_i > b) = \alpha/2$$

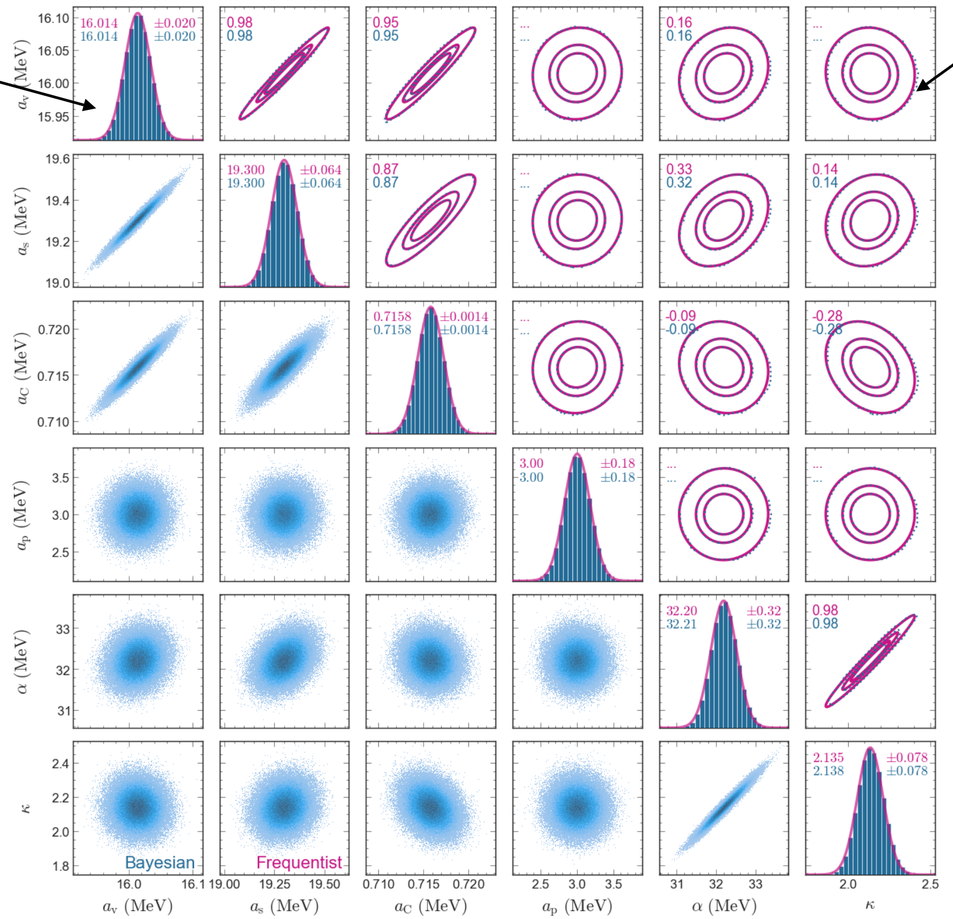
- 平均值 （mean value）

$$\left\langle \theta_i \right\rangle = \int \theta_i p\left(\theta_i \mid \boldsymbol{y}\right) d\boldsymbol{\theta}$$

- 中位数（median value）

$$P\left(\theta_i < U\right) = 0.5$$

- 最大后验(MAP)估计




联合分布（joint distribution）

$$p\left[\left(\theta_i, \theta_j\right) \mid \boldsymbol{y}\right] = \frac{\int p(\boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d} \prod_{k \neq i,j} \theta_k}{\int p(\boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d} \prod_k \theta_k}$$

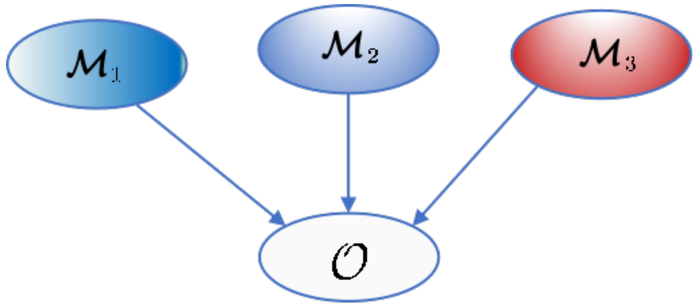
协方差(covariance)

$$\mathrm{Cov}\left(\theta_i, \theta_j\right) = \int \left(\theta_i - \left\langle \theta_i \right\rangle\right) \left(\theta_j - \left\langle \theta_j \right\rangle\right) p\left[\left(\theta_i, \theta_j\right) \mid \boldsymbol{y}\right] d\theta_i d\theta_j$$

相关系数（correlation coeffecient）

$$R = \frac{\mathrm{Cov}\left(\theta_i, \theta_j\right)}{\sigma_i \sigma_j}$$

# Inter-model uncertainty




*Which model should we trust?*

Determine a weighting factor $\omega_i$ for each model.

- **Model selection**: identify the best model(s) with the largest $\omega_i$
- **Model averaging**: averaging model predictions by

$$\mathcal{O}_{MA} = \frac{\sum_i \mathcal{O}_i \omega_i}{\sum_i \omega_i}$$

How to determine $\omega_i$?

Burnham & Anderson, Model Selection and Multimodel Inference: A Practical Information Theoretic Approach

Udo von Toussaint, Rev. Mod. Phys. 83, 943 (2011)

# Bayesian evidence

✦ **Evidence** (a larger evidence indicates a better model)

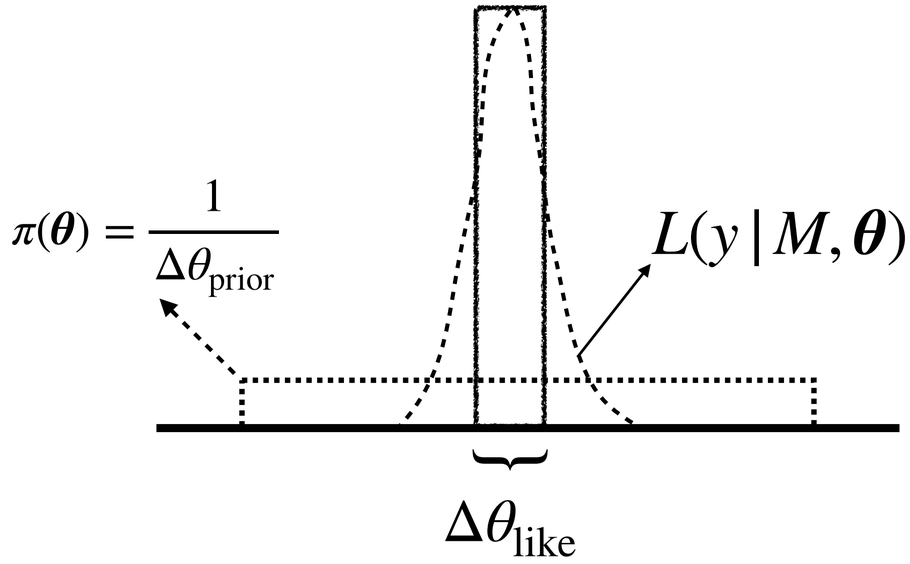
$$Z \equiv p(\boldsymbol{y} \mid M) = \int d\boldsymbol{\theta} L(\boldsymbol{y} \mid M, \theta) \pi(\boldsymbol{\theta} \mid M)$$

$$\approx L\left(\boldsymbol{y} \mid M, \theta_{\mathrm{ML}}\right) \left(\Delta\theta_{\mathrm{like}} / \Delta\theta_{\mathrm{prior}}\right)^{N_\theta}$$

Maximum likelihood

No. of parameter, model complexity

Udo von Toussaint, Rev. Mod. Phys. 83, 943 (2011)




✦ posterior predictive distribution:

$$Z_i^{\mathrm{pseu}} \equiv \int L\left(\boldsymbol{y}_{\mathrm{ev}} \mid \boldsymbol{\theta}, M_i\right) p\left(\boldsymbol{\theta} \mid \boldsymbol{y}, M_i\right) d\boldsymbol{\theta}$$

- Model calibration and evaluation use different data sets.
- Select $\boldsymbol{y}_{\mathrm{ev}}$ that are closely related to the target observables.
- Posterior from calibration → Prior for evaluation/prediction.

Neufcourt et al., PRL 122, 062502 (2019)
Neufcourt et al., PRC 101, 014319 (2020)
Kejzlar et al., JPG 47, 094001 (2020)
Cirigliano et al., JPG 49, 120502 (2022)


17

# Bayesian model averaging

- Model prior $\pi(M_i)$:
  our preference on $M_i$ before seeing the data.

- Model likelihood:
  (pseudo-)evidence $Z_i = p(\boldsymbol{y} \mid M_i)$ $(Z_i^{\text{pseu}} = p(y_{\text{ev}} \mid y, M_i))$

- Model posterior:

$$p(M_i \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid M_i)\pi(M_i)}{\sum_j p(\boldsymbol{y} \mid M_j)\pi(M_j)}$$

- $$p(M_i \mid \boldsymbol{y}_{ev}) = \frac{p(y_{\text{ev}} \mid y, M_i)\pi(M_i)}{\sum_j p(y_{\text{ev}} \mid y, M_j)\pi(M_j)}$$

- Model averaging for observable $\mathcal{O}$:

$$p(\mathcal{O} \mid \boldsymbol{y}) = \sum_i p(\mathcal{O} \mid \boldsymbol{y}, M_i) p(M_i \mid \boldsymbol{y})$$

PHYSICAL REVIEW LETTERS **122,** 062502 (2019)

**Neutron Drip Line in the Ca Region from Bayesian Model Averaging**

Léo Neufcourt,[1,2] Yuchen Cao (曹宇晨),[3] Witold Nazarewicz,[4] Erik Olsen,[2] and Frederi Viens[1]

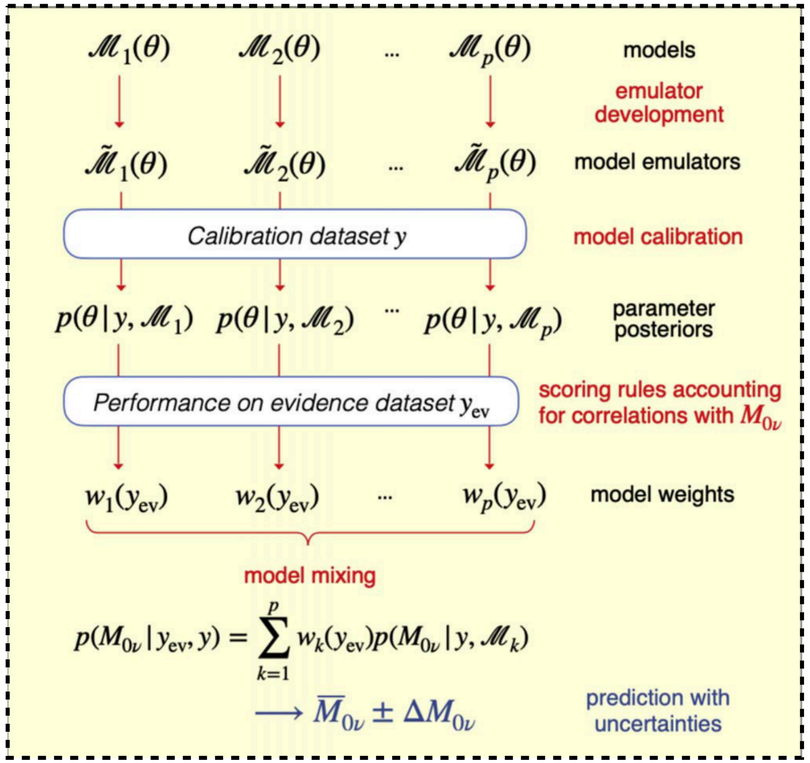
PHYSICAL REVIEW LETTERS **126,** 242301 (2021)

Editors' Suggestion

**Phenomenological Constraints on the Transport Properties of QCD Matter with Data-Driven Model Averaging**

D. Everett,[1] W. Ke,[2,3] J.-F. Paquet,[4] G. Vujanovic,[5] S. A. Bass,[4] L. Du,[1] C. Gale,[6] M. Heffernan,[6] U. Heinz,[1] D. Liyanage,[1] M. Luzum,[7] A. Majumder,[5] M. McNelis,[1] C. Shen,[5,8] Y. Xu,[4] A. Angerami,[9] S. Cao,[5] Y. Chen,[10,11] J. Coleman,[12] L. Cunqueiro,[13,14] T. Dai,[4] R. Ehlers,[13,14] H. Elfner,[15,16,17] W. Fan,[4] R. J. Fries,[18,19] F. Garza,[18,19] Y. He,[20] B. V. Jacak,[2,3] P. M. Jacobs,[2,3] S. Jeon,[6] B. Kim,[18,19] M. Kordell II,[18,19] A. Kumar,[5] S. Mak,[12] J. Mulligan,[2,3] C. Nattrass,[13] D. Oliinychenko,[3] C. Park,[6] J. H. Putschke,[5] G. Roland,[10,11] B. Schenke,[21] L. Schwiebert,[22] A. Silva,[13] C. Sirimanna,[5] R. A. Soltz,[5,9] Y. Tachibana,[5] X.-N. Wang,[20,2,3] and R. L. Wolpert[12]

(JETSCAPE Collaboration)




V. Cirigliano et al, J. Phys. G 49, 120502 (2022)

# Bayesian model averaging

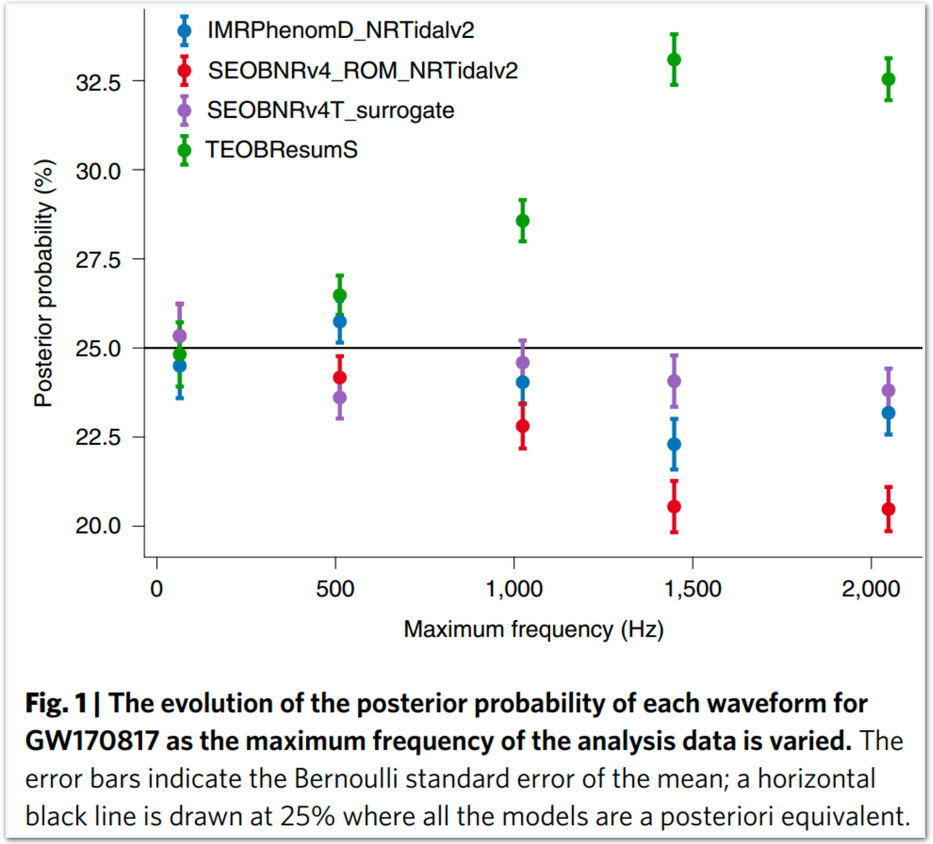



**Fig. 1 | The evolution of the posterior probability of each waveform for GW170817 as the maximum frequency of the analysis data is varied.** The error bars indicate the Bernoulli standard error of the mean; a horizontal black line is drawn at 25% where all the models are a posteriori equivalent.

nature astronomy

ARTICLES

https://doi.org/10.1038/s41550-022-01707-x

Check for updates

## The use of hypermodels to understand binary neutron star collisions

Gregory Ashton [1,2] ✉ and Tim Dietrich [3,4]

## Comparing gravitational waveform models for binary black hole mergers through a hypermodels approach

Anna Puecher [1,2], Anuradha Samajdar [2,3], Gregory Ashton [4], Chris Van Den Broeck [1,2], and Tim Dietrich [3,5]

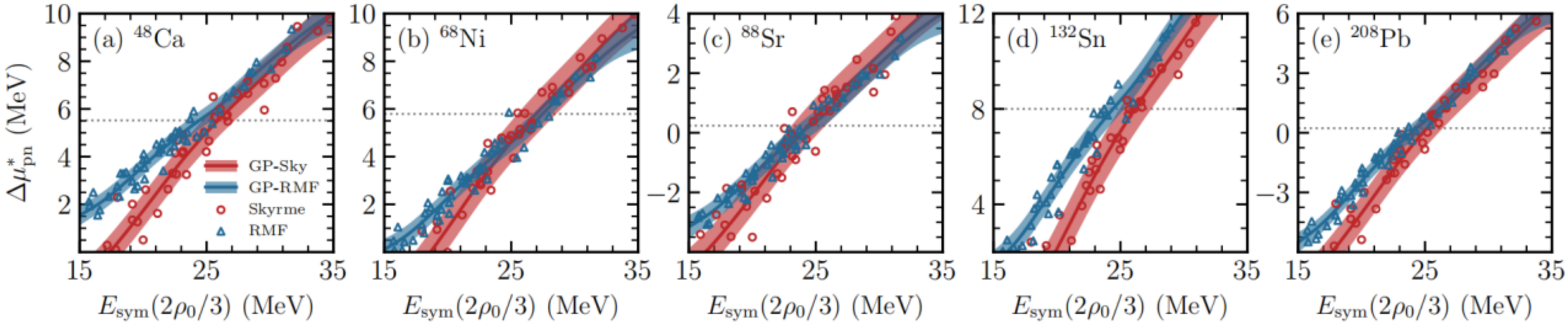
Show more

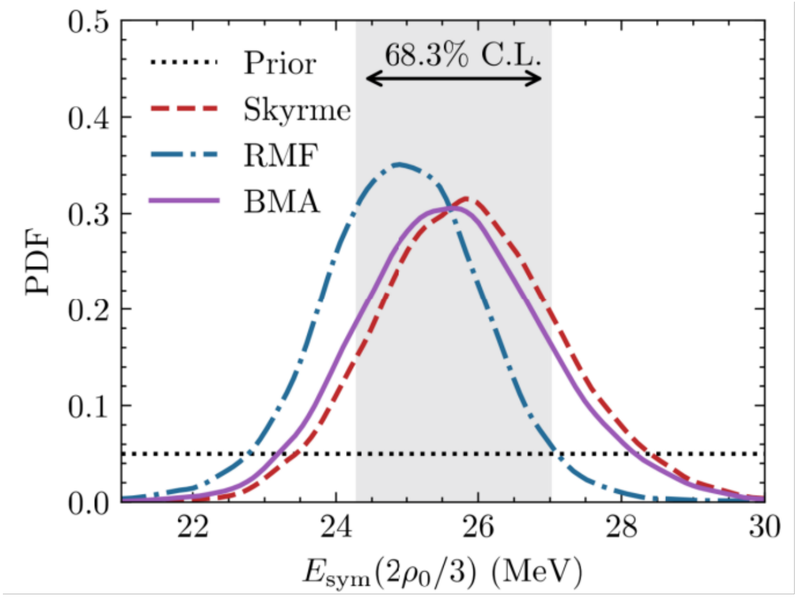
Phys. Rev. D **109**, 023019 – **Published 11 January, 2024**

DOI: https://doi.org/10.1103/PhysRevD.109.023019

# BMA for nuclear symmetry energy

$$\Delta\mu^*_{\rm pn} = \frac{1}{4}[B(N,Z+2) - B(N,Z-2) - B(N+2,Z) + B(N-2,Z)]$$




*M. Qiu, B.-J. Cai, L.-W. Chen, C.X.Yuan and **ZZ,** PLB 849, 138435 (2024)*

# Bayesian model selection

- Bayes' factor :

$$B_{12} = \frac{Z_1}{Z_2}$$

**Jeffreys' Scale (rule of thumb)**

- $B_{12} < 1$: support for $M_2$.
- $1 < B_{12} < 3$: barely worth mentioning.
- $3 < B_{12} < 10$: moderate evidence for $M_1$.
- $10 < B_{12} < 100$: strong evidence.
- $> 100$: decisive evidence.

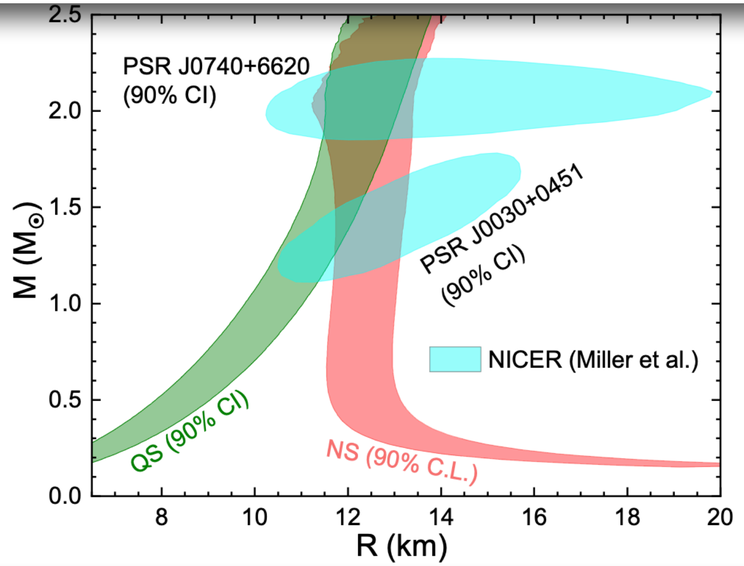
arXiv > astro-ph > arXiv:2308.16783

Astrophysics > High Energy Astrophysical Phenomena

[Submitted on 31 Aug 2023]

**Neutron Star vs Quark Star in the Multimessenger Era**

Zheng Cao, Lie-Wen Chen




arXiv > nucl-th > arXiv:2507.11394v1

Search...

Help | Ad

Nuclear Theory

[Submitted on 15 Jul 2025]

**Bayesian Model Selection and Uncertainty Propagation for Beam Energy Scan Heavy-Ion Collisions**

Syed Afrid Jahan, Hendrik Roch, Chun Shen

# Nested sampling

| Code | Methods | Dynamic | Languages | Field | Pub. Year |
|---|---|---|---|---|---|
| CosmoNest [60, 61] | ellipsoid | fixed | Fortran | Cosmology | 2006 |
| MultiNest [48, 84] | multi-ellipsoid | fixed | Fortran, C/C++, Python | Cosmology | 2008 |
| DIAMONDS [249] | multi-ellipsoid | fixed | C++ | Astrophysics | 2015 |
| nestle [250] | ellipsoid, multi-ellipsoid | fixed | Python | Astrophysics | 2015 |
| nessai [90, 91] | normalising flow ellipsoid | fixed | Python | Gravitational waves | 2021 |
| (dy)PolyChord [53, 65] | slice | dynamic | Fortran, C/C++, Python | Cosmology | 2015 |
| LALInferenceNest [180] | random walk, ensemble, differential evolution | fixed | C | Gravitational waves | 2015 |
| Nested_fit [104, 257, 258] | random walk | fixed | Fortran | Atomic physics | 2016 |
| cpnest [259] | slice, differential evolution, Gauss, Hamiltonian, ensemble | fixed | Python | Gravitational waves | 2017 |
| pymatnest [44] | random walk, Galilean, symplectic Hamiltonian | fixed | Python | Materials | 2017 |
| NNest [261] | normalising flow random walk | fixed | Python | Cosmology | 2019 |
| DNest5 [55] | user-defined, random walk | diffusive | C++ | Astrophysics | 2020 |
| BayesicFitting [263] | random walk, slice, Galilean, Gibbs | fixed | Python | Astronomy | 2021 |
| dynesty [52] | ellipsoid, multi-ellipsoid, MLFriends & Gauss, slice, Hamiltonian | dynamic | Python | Astrophysics | 2020 |
| UltraNest [92] | MLFriends + ellipsoid & Gauss, hit-and-run, slice | reactive | Python, Julia, R, C/C++, Fortran | Astrophysics | 2020 |
| jaxns [266] | multi-ellipsoid & slice | fixed | jax | Astronomy | 2021 |

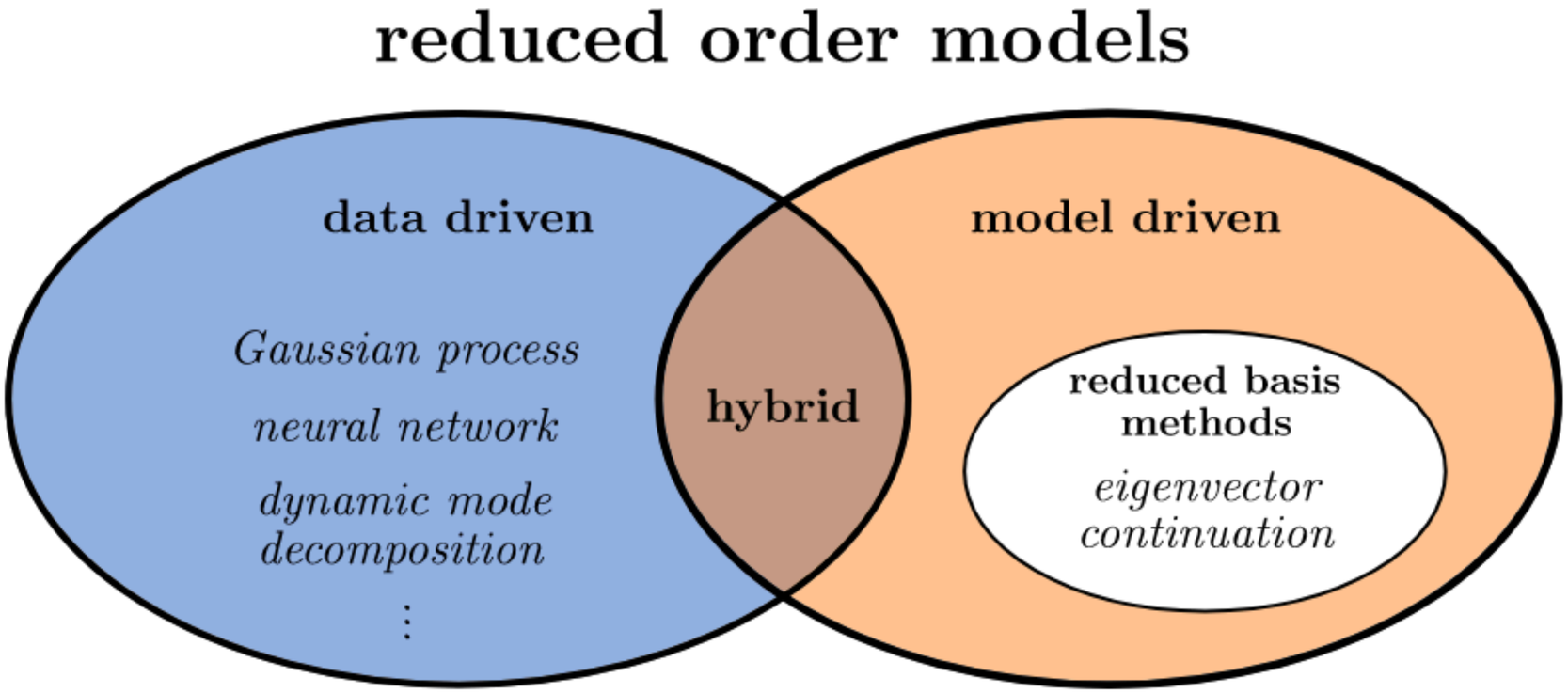
Table 2 | **Comparison of NS codes.** The first two groups are region samplers and step samplers, respectively, whereas the third group offers both. Dynamic implementations allow the number of live points to be changed during a run. We show the language in which the NS code was written followed by any additional languages for which interfaces exist, and the field from which the code originated (though most are general purpose codes).

Ashton et al., Nat Rev Methods Primers 2, 39 (2022)

# Surrogate Emulators

- Gaussian process (GP)
- Reduced basis method (RBM)
- Parametric Matrix Model (PMM)

# Emulator




T. Duguet, et al., Rev. Mod. Phys. 96, 031002 (2024)

# Gaussian process

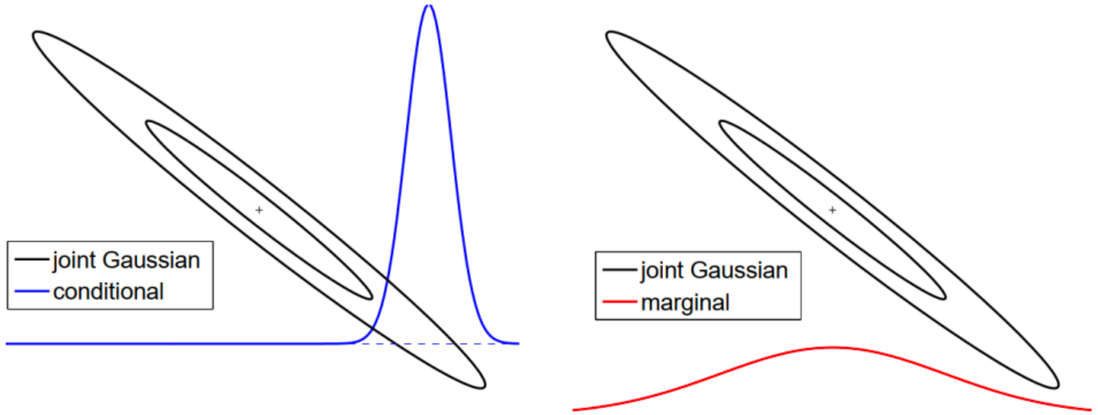
- **Definition**: *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

  Rasmussen and Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006

- **Multivariate Gaussian distribution**

$$p(\boldsymbol{x};\boldsymbol{\mu},\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

where μ is the mean vector and Σ is the covariance matrix.




Conditional and marginal distributions of a multivariate Gaussian are gaussian

The Gaussian Processes Web Site

http://www.gaussianprocess.org

# Gaussian process emulator

- **GP** can serve as a fast surrogate of a slow model.

  Given training data from model calculations, a GP can guess model prediction $y_*$ at $x_*$.

  $X = (\vec{x}_1, \cdots, \vec{x}_m)$, $\boldsymbol{y} = (y_1, \cdots, y_m)$

- Considering $\boldsymbol{y}$ and $y_*$ as **random variables**, seek for $p(y_* | \boldsymbol{y})$.
  - $y_*$ and $\boldsymbol{y}$ should be correlated and the correlation depends on $|x_* - x_i|$.

- **Prior:**

Covariance matrix

$$\begin{bmatrix} \boldsymbol{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Sigma^0(X,X) & \Sigma^0(X,\vec{x}_*) \\ \Sigma^0(\vec{x}_*,X) & \Sigma^0(\vec{x}_*,\vec{x}_*) \end{bmatrix} \right)$$

- Conditional distribution:

$$y_* \,|\, \boldsymbol{y} \sim \mathcal{N}(\mu_*, \sigma_*^2)$$

$$\mu_* \,|\, \boldsymbol{y} = \Sigma^0(x_*, X)\Sigma^0(X,X)^{-1}\boldsymbol{y}$$

$$\sigma_*^2 = \sigma(x_*, x_*) - \Sigma^0(x_*, X)\Sigma^0(X,X)^{-1}\Sigma^0(X, x_*)$$

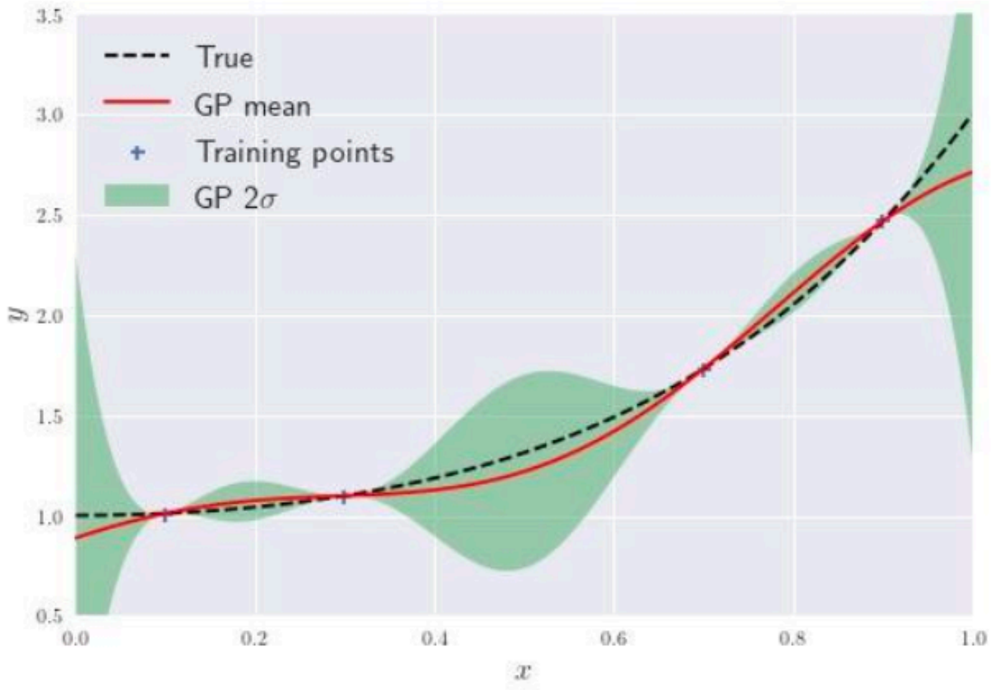
A typical covariance function:

$$\Sigma^0(X,X)=\begin{pmatrix}\sigma(\vec{x}_1,\vec{x}_1) & \cdots & \sigma(\vec{x}_1,\vec{x}_m)\\ \vdots & \ddots & \vdots\\ \sigma(\vec{x}_m,\vec{x}_1) & \cdots & \sigma(\vec{x}_m,\vec{x}_m)\end{pmatrix},\quad \sigma(\vec{x}_i,\vec{x}_j)=\sigma_{\mathrm{GP}}^2\exp\left(-\sum_k\frac{(x_{i,k}-x_{j,k})^2}{2l_k^2}\right)+\sigma_n^2\delta_{ij}$$

Decrease with distance

***Hyper-parameters:***

- Noise $\sigma_n$ = uncertainty of a stochastic model
- Amplitude $\sigma_{GP}$ only affects $\sigma_*$ but not affect $\mu_*$ if $\sigma_n$=0
- length scale $l_k$ : usually $\mu_*$ is relatively insensitive to $l_k$




- Reproduce the training points without noise
- Make predictions with statistical uncertainties

1. How to determine the hyper parameters?
2. How to choose training data?
3. How to treat multiple observables?

# Hyper-parameters in GP

- Log marginal likelihood :

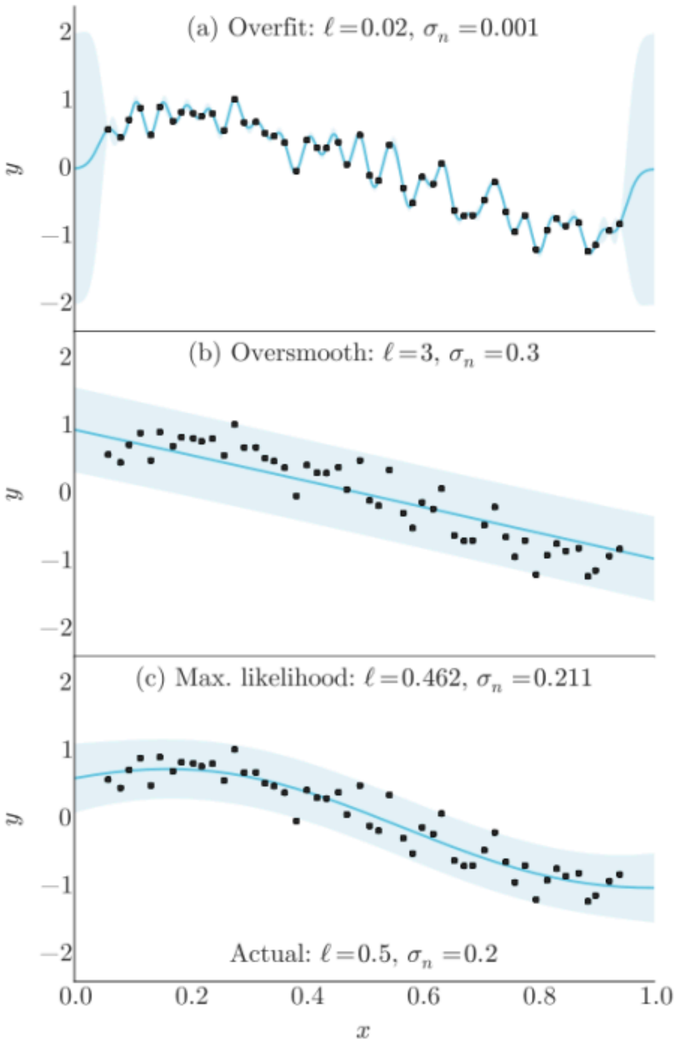
$$\log P(\mathbf{y} \mid X, \theta) = -\frac{1}{2}\mathbf{y}^{\top}(\Sigma^{0})^{-1}\mathbf{y} - \frac{1}{2}\log|\Sigma^{0}| - \frac{m}{2}\log 2\pi$$

- Hyper-parameters can be determined by maximizing the marginal likelihood

- Example:

$$\sigma(x, x') = \exp\left(-\frac{|x - x'|^2}{2\ell^2}\right) + \sigma_n^2 \delta_{xx'}$$

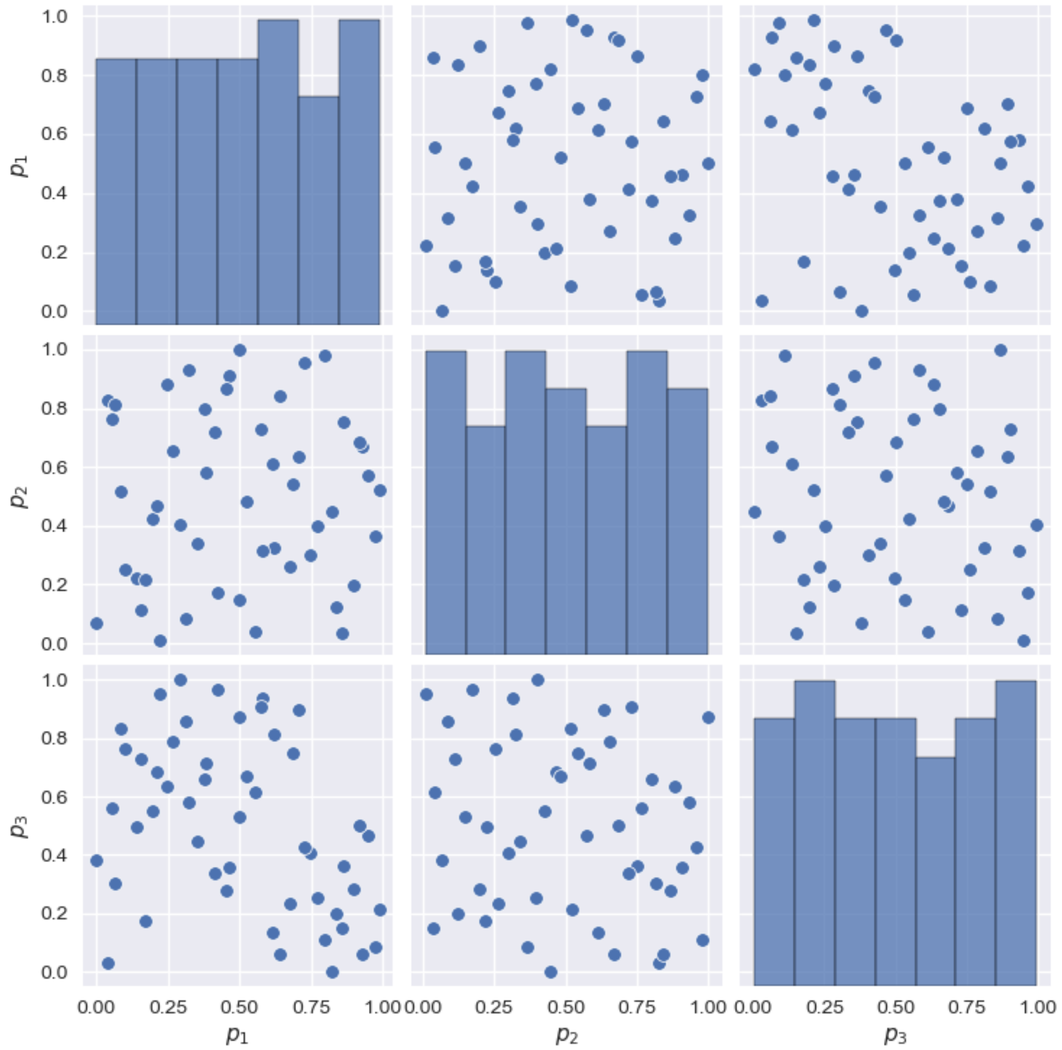
*Bernhard et al., PRC 91, 054910 (2015)*



# How to choose training points

- Latin Hypercube sampling

```
from scipy.stats import qmc
sampler = qmc.LatinHypercube(d=3)
sample = sampler.random(n=50)
```




- Greedy approach

C. E. Rasmussen & C. K. I. Williams,
Gaussian Processes for Machine Learning, the MIT Press, 2006,

# Principal component analysis (PCA)

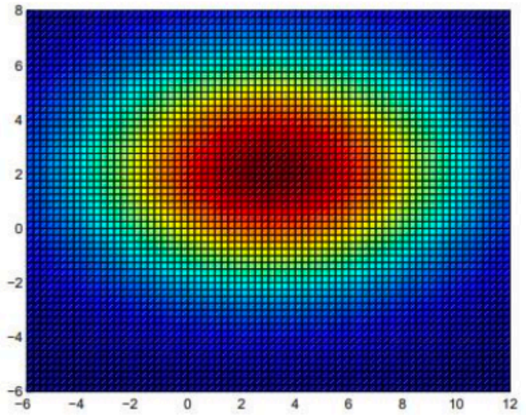
- How to treat multiple observables?

The central idea of principal component analysis is to reduce the dimensionality of a data set in which there are a large number of **interrelated variables**, while retaining as much as possible of the variation present in the data set. This reduction is achieved by transforming to a new set of variables, the principal components, which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables.

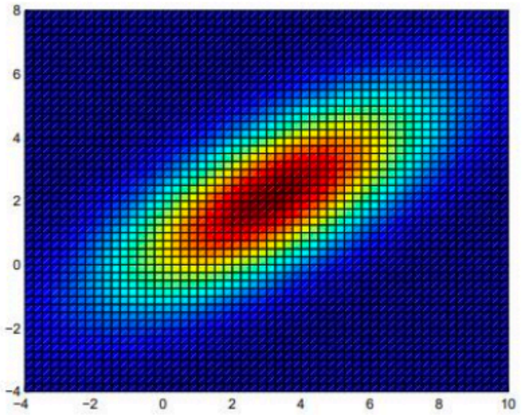
I.T. Jolliffe, *Principal Component Analysis*, Second Edition

uncorrelated

$$\mathcal{N}\left(\mu=\begin{bmatrix}3\\2\end{bmatrix},\Sigma=\begin{bmatrix}25 & 0\\0 & 9\end{bmatrix}\right)$$




correlated




$$\mathcal{N}\left(\mu=\begin{bmatrix}3\\2\end{bmatrix},\Sigma=\begin{bmatrix}10 & 5\\5 & 5\end{bmatrix}\right)$$

# PCA procedure

- m correlated observables/model output $o_i$
- $o_i$ are functions of model parameters, but linear combinations of $o_i$ may be independent of parameters.

$$\tilde{o}_i = \frac{o_i - \langle o_i \rangle}{\sigma_i}$$ average over training data

- Covariance matrix:

$$M = \langle \tilde{o}_i \tilde{o}_j \rangle \ , \quad (\langle \tilde{o}_i \rangle = \langle \tilde{o}_j \rangle = 0)$$

- Eigenvalue decomposition

$$M = U \Lambda U^T$$

$$\Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{bmatrix}$$

Eigenvalue

$$U = (\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_m)$$

Normalized Eigenvectors

# PCA

$$M = U \Lambda U^T$$
$$UU^T = I$$

| Physics output | Principal component |
|---|---|
| $\tilde{\boldsymbol{o}}$ | $\boldsymbol{z} = \tilde{\boldsymbol{o}} U$ |
| $\langle \tilde{\boldsymbol{o}} \rangle = 0$ | $\langle \boldsymbol{z} \rangle = 0$ |
| $\langle \tilde{o}_i \tilde{o}_j \rangle = \mathrm{M}$ | $\langle z_i z_j \rangle = U^T \langle \tilde{o}_i \tilde{o}_j \rangle U = \Lambda$ |
| Correlated | Uncorrelated |

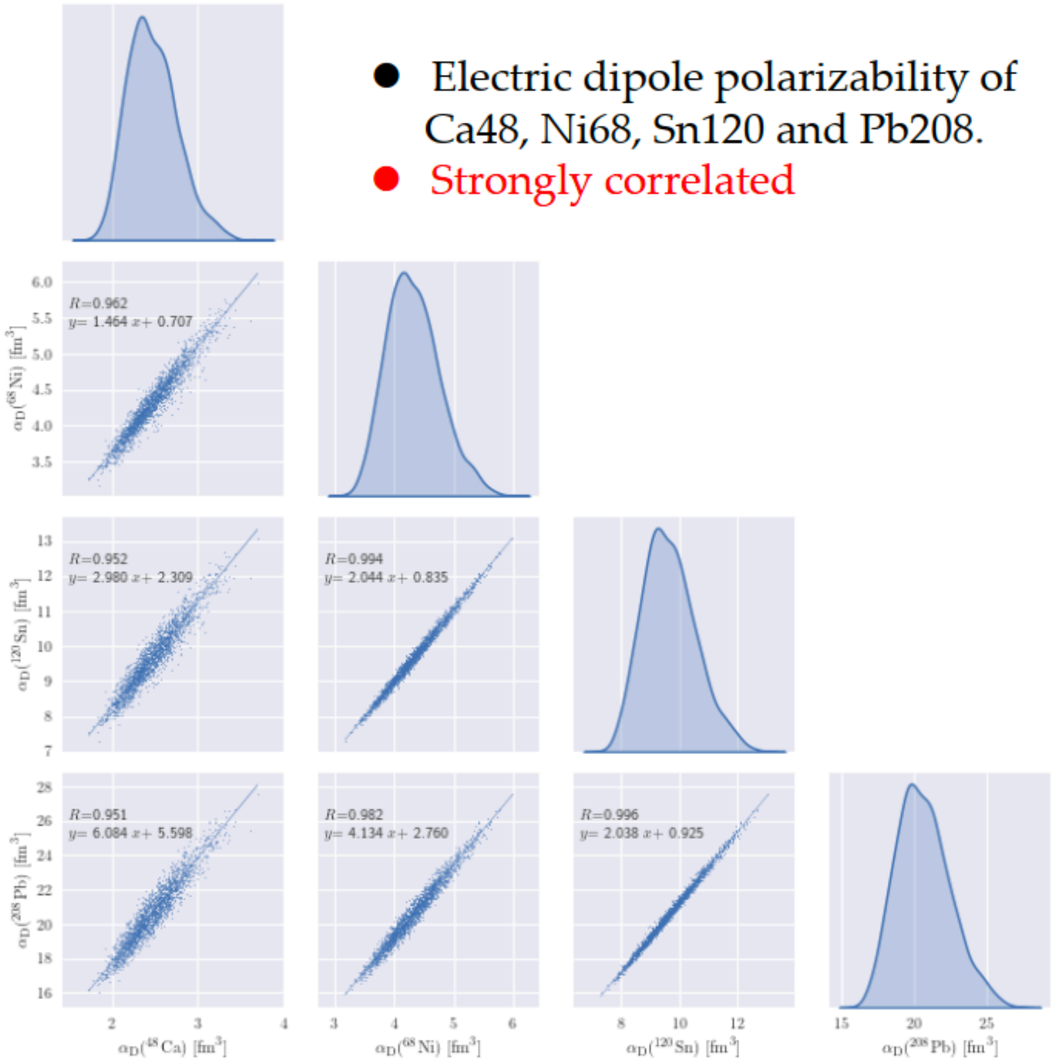
**Covariance Matrix**

$$\mathcal{L} \propto \exp\left[ - \sum_i \frac{(o_i - o_i^{\exp})^2}{2\sigma_i^2} \right]$$
$$= \exp\left[ - \frac{1}{2} \sum_i (\tilde{o}_i - \tilde{o}_i^{\exp})^2 \right]$$
$$= \exp\left[ - \frac{1}{2} \sum_i (z_i - z_i^{\exp})^2 \right]$$

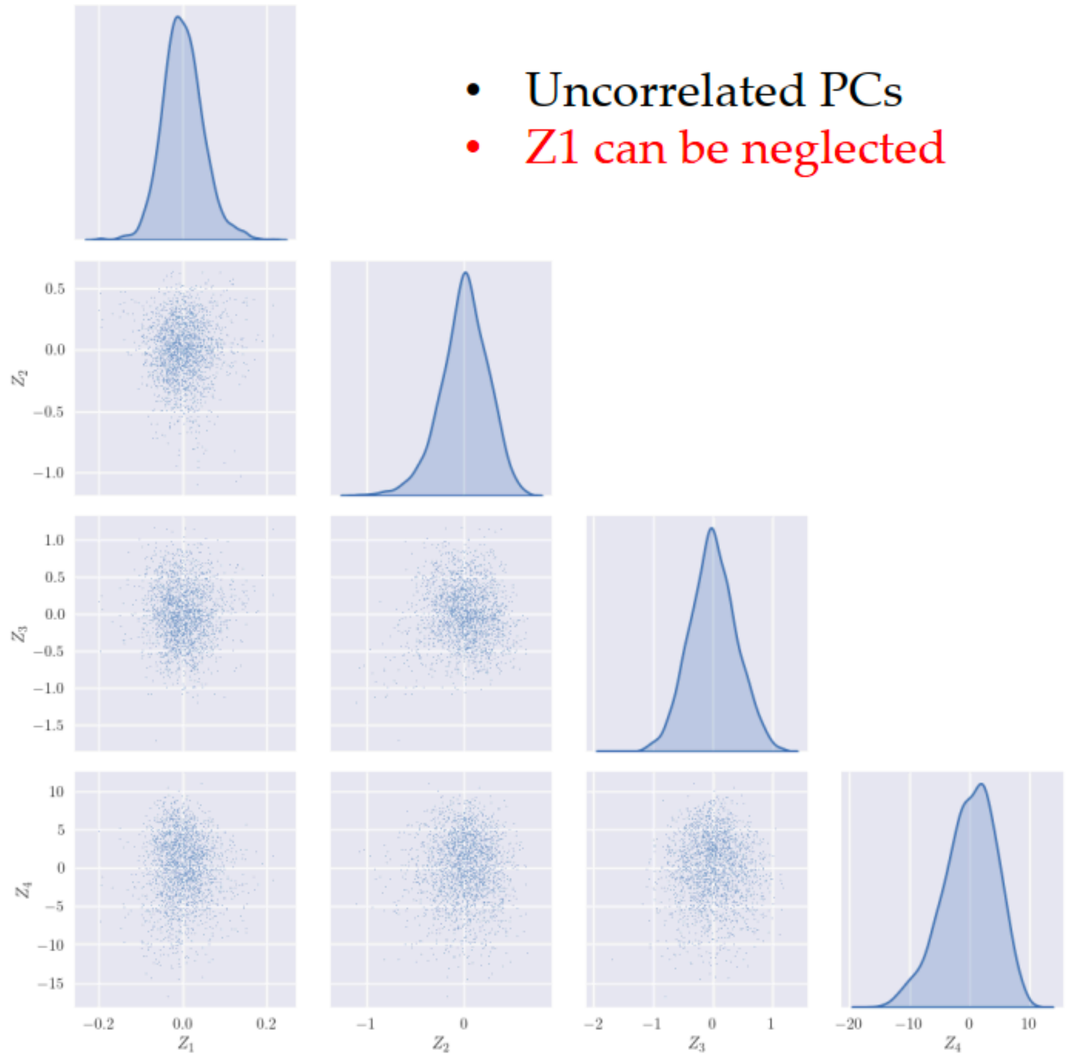
- $\lambda_i$ is the variance of $z_i$.
- Retain the first $q$ PCs with the largest eigenvalues.
- m correlated outputs → $q$ uncorrelated PCs.

$$\sum_{i=1}^{q} \lambda_i \Big/ \sum_{i=1}^{m} \lambda_i \leq V$$

# PCA example



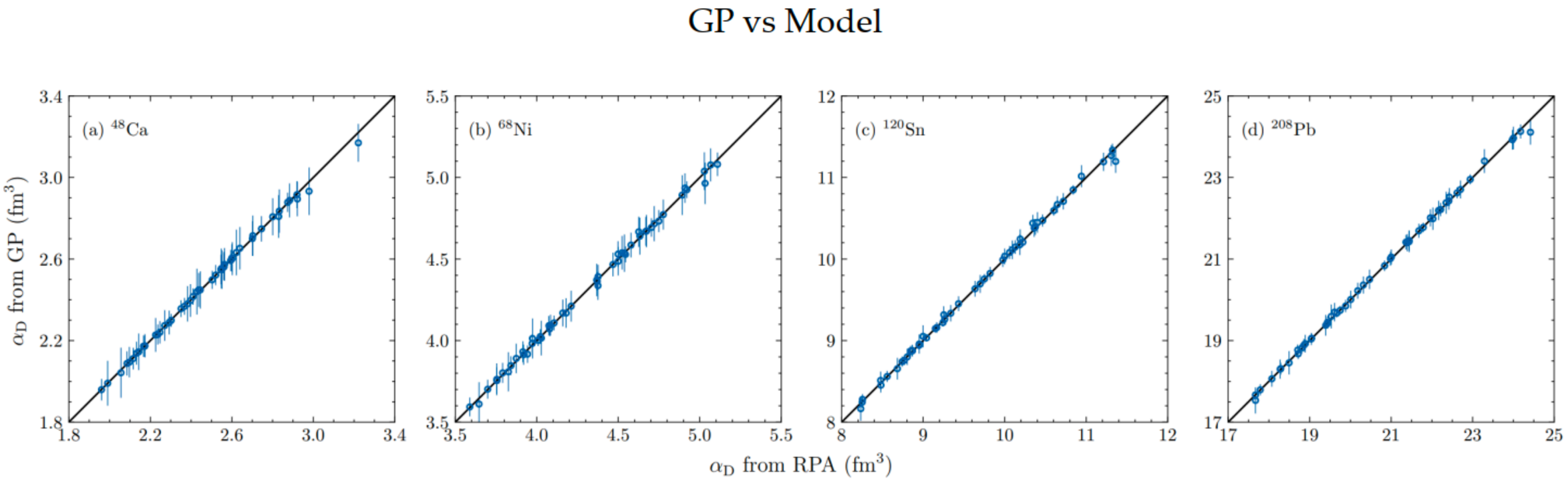

- Electric dipole polarizability of Ca48, Ni68, Sn120 and Pb208.
- Strongly correlated




- Uncorrelated PCs
- Z1 can be neglected

~ 0.2 ~ 1 ~ 1 ~ 10

# PCA+GP




- Tune GPs for dominant PCs instead of all model outputs.
- PCA+GPs => a multivariate emulator.

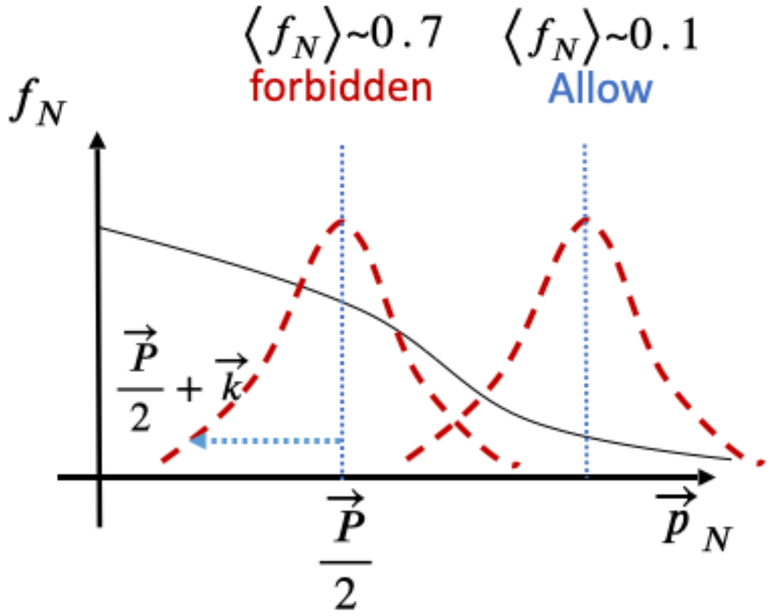
# Example: Bayesian analysis of light-nuclei production in HICs

**Model**: LBUU transport model

R. Wang, **ZZ**, L.-W. Chen & Y.-G. Ma, Front. Phys. 8:330 (2020)
R. Wang et al., PRC108, L031601 (2023)

**Parameters:** $f^{\rm cut}_{A=2,3,4}$

$$\langle f_N \rangle_i (\vec{P}) \equiv \int f_N^{\rm tot}\left(\frac{\vec{P}}{A_i}+\vec{p}\right)\left|\phi_i(\vec{p})\right|^2 \mathrm{d}\vec{p} \leqslant f^{\rm cut}_{A=A_i}$$

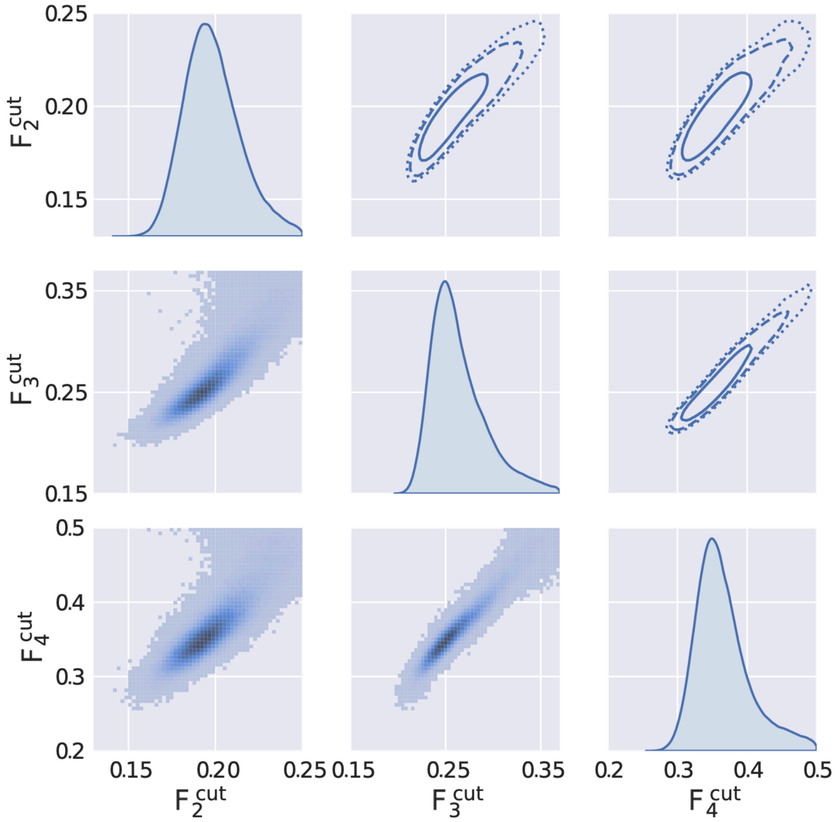


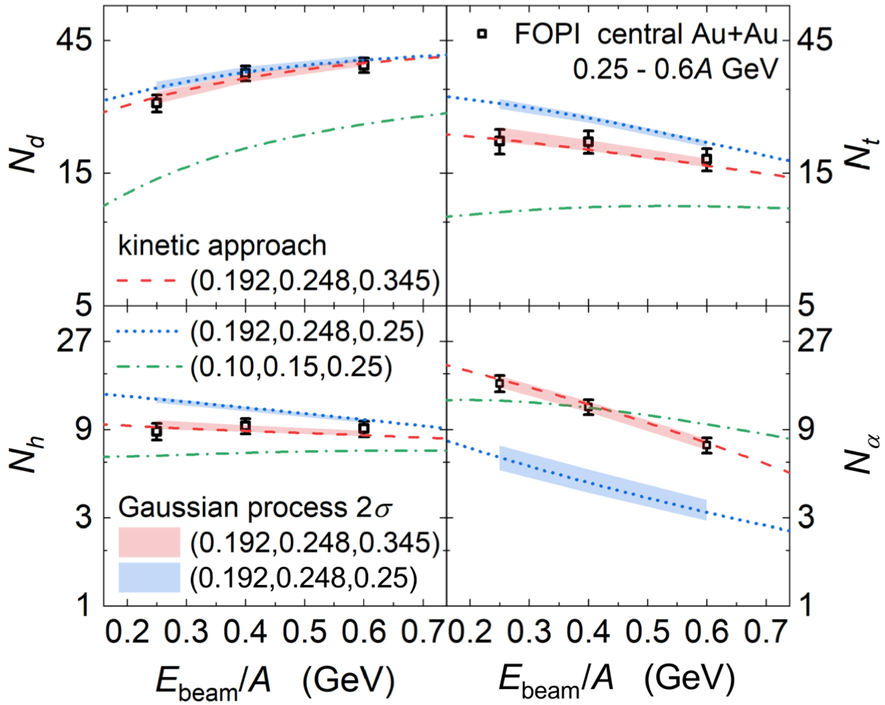

**Data**: d, t, h, $\alpha$ yields in Au+Au central collisions at energies of 0.25 - 0.6*A* GeV.

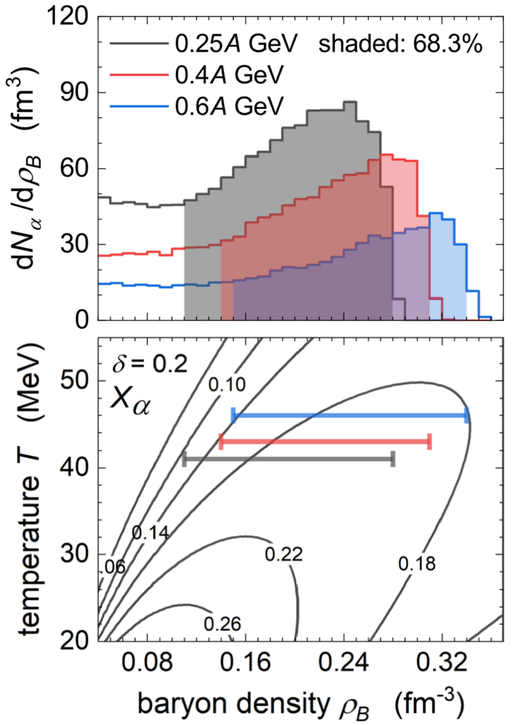
- 30 parameter sets from **Latin Hypercube Sampling**
- **Emulator**: Principal component analysis + Gaussian process
- Metropolis-Hastings (MH) algorithm for **MCMC**

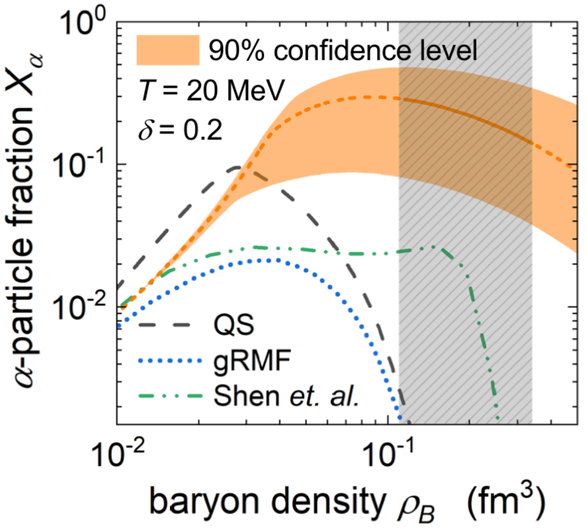
BAND
Bayesian Analysis of Nuclear Dynamics

surmise 0.2.1
pip install surmise



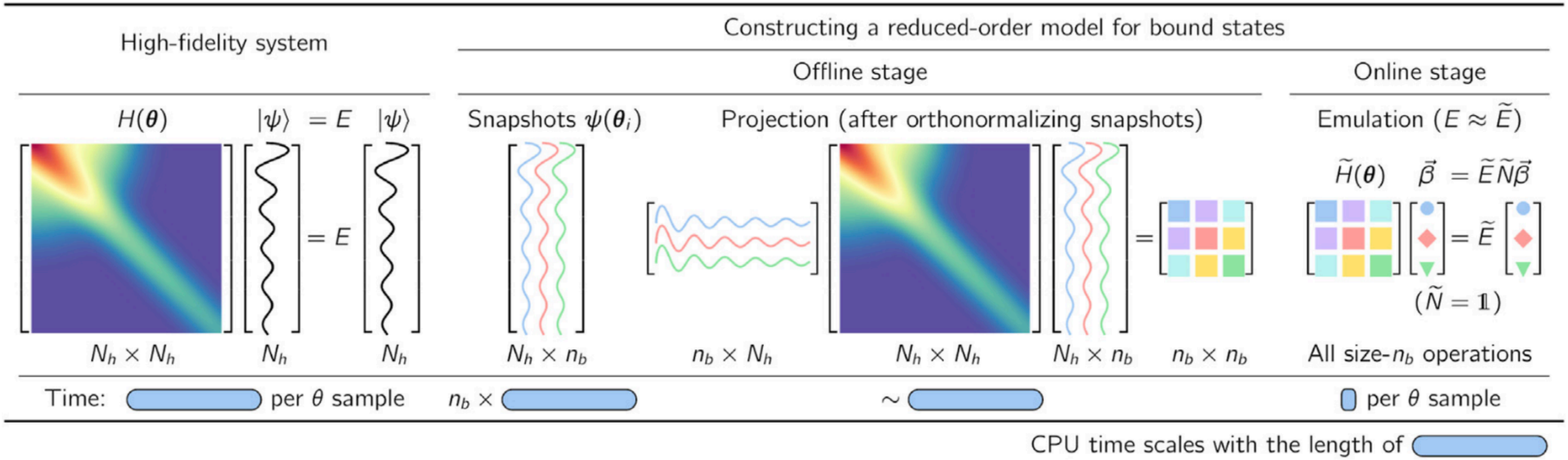




✦ Unexpectedly abundant α clustering in hot and dense nuclear matter.

R. Wang, **ZZ**, et al., arXiv:2507.16613

# Reduced basis method (RBM)

C. Drescher et al., Front. Phys. 10,1092931 (2022)




**高维($N_h$) 问题**

$$H(\theta)|\psi\rangle = E|\psi\rangle$$

- 3D finite difference $n_h \sim 100^3$

**Off-line 阶段**

对$n_b$个参数$\boldsymbol{\theta}_i$严格求解得到$\psi_i$(snapshots)

**降维 ($N_h \longrightarrow N_b$)**

- 正交化$\psi_i$得到低维基矢 $V = \phi_1, \phi_2, \cdots, \phi_{n_b}$
- 投影$H_{N_h \times N_h} \rightarrow \tilde{H}_{n_b \times n_b}$
- 可用PCA 进一步降维
- EC 通常不做正交化

**Online阶段**

- 在$\phi_i$或$\psi_i$为基矢的子空间中求解 $\tilde{H}(\boldsymbol{\theta})\vec{\beta} = \tilde{E}(\boldsymbol{\theta})\tilde{N}\vec{\beta}$
- 用$\tilde{E}$近似$E$
- 对于$\phi_i$, $\tilde{N}$为单位矩阵
- For $\psi_i$ $\tilde{N}_{ij} = \langle\psi_i|\psi_j\rangle$

Eigenvector continuation (EC) approaches are a subset of RBMs.

D. Frame et al., Phys. Rev. Lett. 121, 032501 (2019).

T. Duguet et al., Rev. Mod. Phys. 96, 031002 (2024)

# How to choose the snapshots?

1. 利用LHC方法在参数空间抽样得到$n_d$ 组参数$\boldsymbol{\theta}_i$.

2. 利用参数$\boldsymbol{\theta}_i$ 求解得到$\psi_i$.

3.奇异值分解(SVD)，并按奇异值截断，得到$n_b \leq n_d$ 个Emualtor basis.

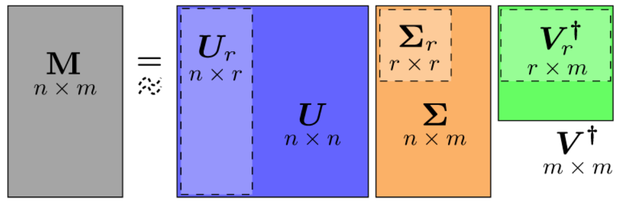
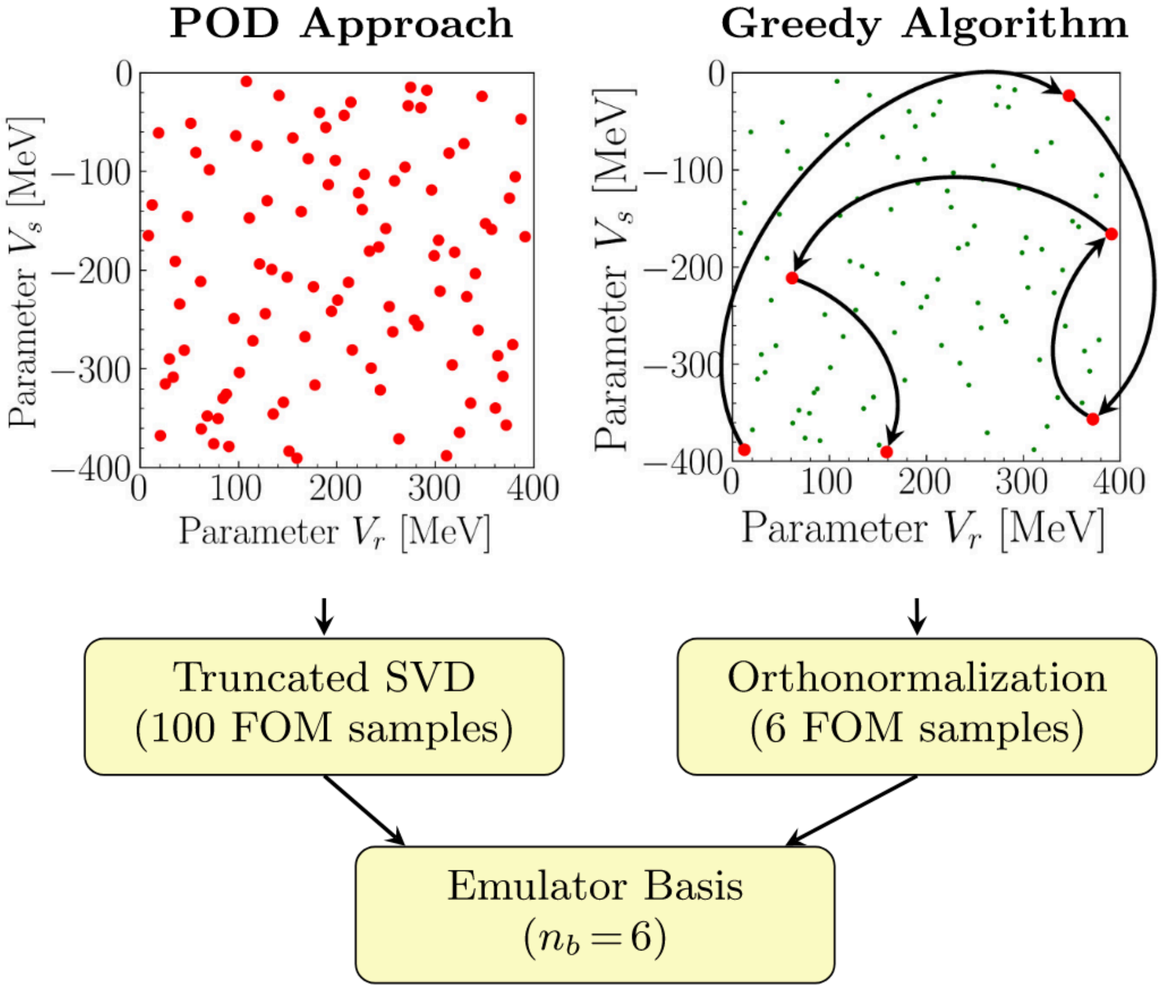



FIG. 3. Illustrations of the SVD (solid outline) and truncated SVD (dashed outline) applied to a real or complex matrix **M**: **U** and **V** are unitary matrices (e.g., $\mathbf{U}^\dagger\mathbf{U} = \mathbf{U}\mathbf{U}^\dagger = \mathbf{1}$) containing the singular vectors; **Σ** is a diagonal matrix containing the singular values in descending order. The truncated SVD truncates the singular vectors corresponding to the $r$ smallest singular values, resulting in a best possible rank-$r$ approximation (in the Frobenius norm) to the original **M**, also called low-rank approximation. The POD is based on a truncated SVD of the snapshot matrix, taking only the (orthonormal) left singular vectors associated with the $r$ largest singular values. The matrix dimensions are annotated.




J. M. Maldonado PRC 112, 024002 (2025)

1. 利用LHC方法在参数空间抽样得到$n_d$ 组参数$\boldsymbol{\theta}_i$.

2. 利用参数$\boldsymbol{\theta}_i$ 求解得到$\psi_i$.

3. 选择起始点构建emulator

4.for $n_b$ =1, 2, …
计算emulator在$\boldsymbol{\theta}_i$ 处误差
加入误差最大点($n_b = n_b + 1$)
构建$n_b$个基矢的emulator

# Applications of RBM

- No-core shell-model
  Konig, S., et al., Phys. Lett. B 810, 135814 (2020).
  Wesolowski, S., et al., Phys. Rev. C 104, 064001(2021).
  Djarv, T., et al, Phys. Rev. C 105, 014005 (2022).
- Coupled cluster
  Ekstrom, A., and G. Hagen, Phys. Rev. Lett. 123, 252501 (2019).
  Hu, B., et al., Nat. Phys. 18, 1196–1200 (2022).
  Jiang, W. G., et al, Phys. Rev. C 109, 064314 (2024).
- Quantum Monte Carlo
  Frame, D. et al. Phys. Rev. Lett. 121, 032501 (2018)
  Sarkar, A et al. Phys. Rev. Lett. 131, 242503 (2023)
- Phenomenological shell model
  Yoshida, S., and N. Shimizu, Prog. Theor. Exp. Phys. 053D02 (2022)
  O. C. Gorton & K. Kravvaris, Phys.Rev.C 112 (2025) 1, 014302

- Energy density functional
  P. Giuliani et al., Front. Phys. 10:1054524.
  X. Zhang et al., Phys.Rev.C 112 (2025) 2, L021302
- Quantum scattering
  Furnstahl, R. J, et al, Phys. Lett. B 809, 135719 (2020).
  Drischler, C, Phys. Lett. B 823, 136777 (2021)
  Garcia, A. J, Phys. Rev. C 107, 054001 (2023)
  J. Liu, J. Lei & Z. Ren, Phys. Lett. B 858 (2024) 139070
  K. Hagino et al., Phys.Rev.C 112 (2025) 2, 024618
- Resonances
  Yapa, N et al, Phys. Rev. C 107, 064316 (2023).
  Yapa, N., and S. Konig, Phys. Rev. C 106, 014309 (2022).

.....

# ROSE: A reduced-order scattering emulator for optical models

PDF

Share

D. Odell[1,*], P. Giuliani[2,3,†], K. Beyer[4,‡], M. Catacora-Rios[2,5], M. Y.-H. Chan[6,§], E. Bonilla[7,‖], R. J. Furnstahl[8,¶], K. Godbey[2,#], and F. M. Nunes[2,5,**]

Show more

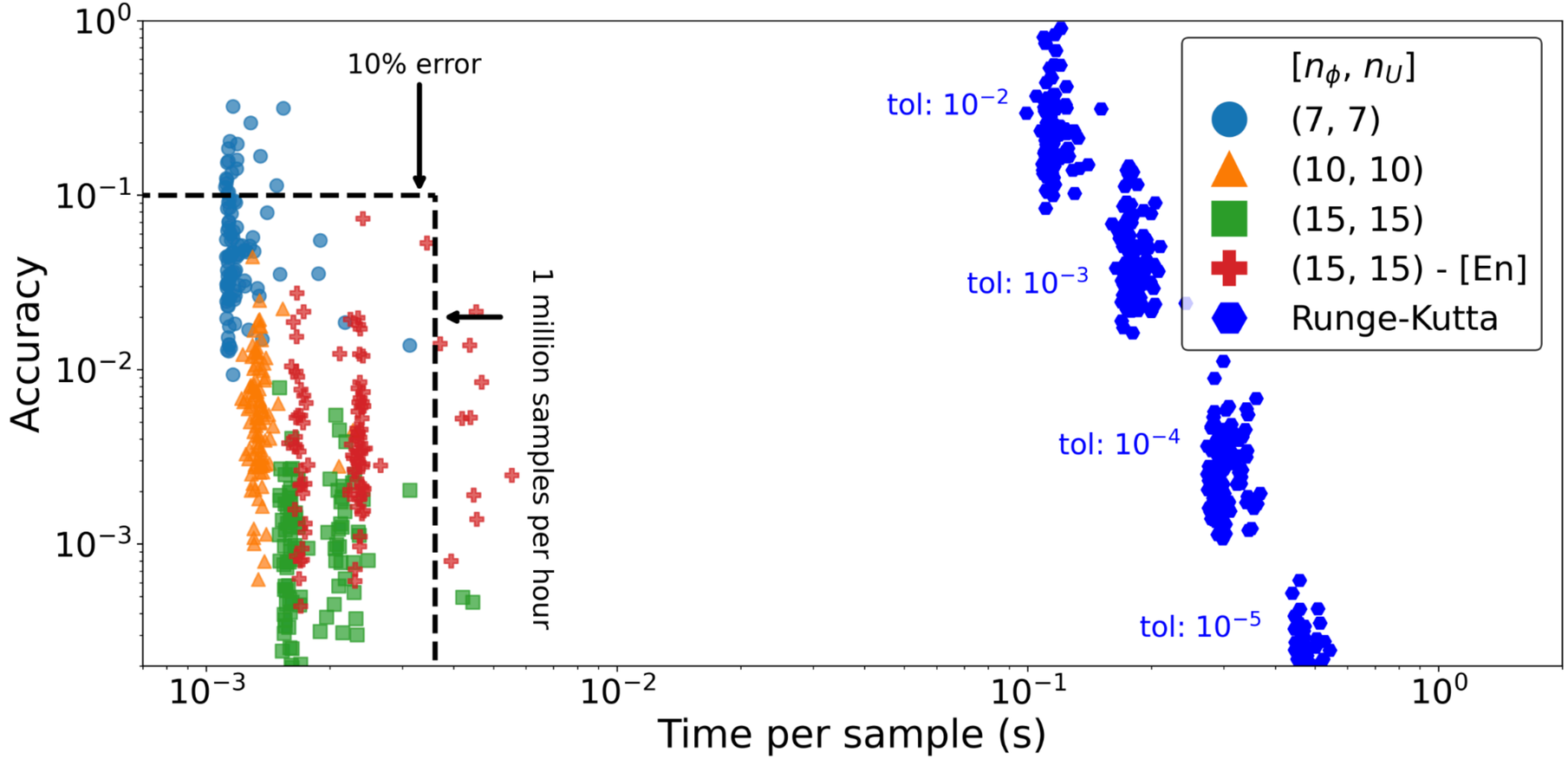
nuclear-rose 1.1.7

pip install nuclear-rose

Phys. Rev. C **109**, 044612 – **Published 11 April, 2024**

DOI: https://doi.org/10.1103/PhysRevC.109.044612

Export Citation



# GP vs RBM

**Gaussian process**

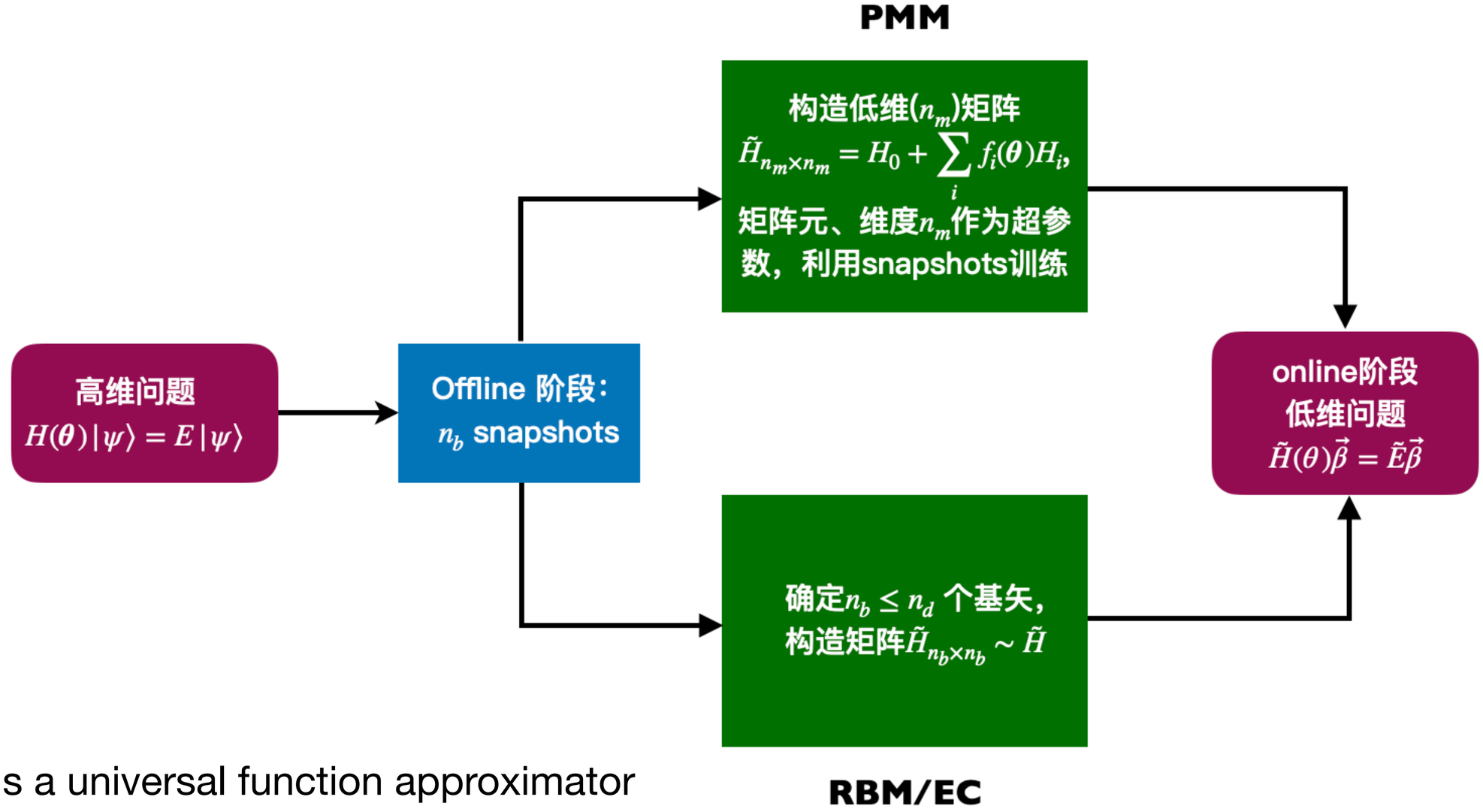
- Data-driven
  - 易用、通用
- 预测带误差

**Reduced Basis method**

- Model-driven
  - 需要根据具体问题/模型构建
- 精度高
  - 误差随$n_b$指数衰减
- 外推能力强

A. Sarkar & D. Lee, PRL 126, 032501 (2021)

# Parametric matrix model (PMM)

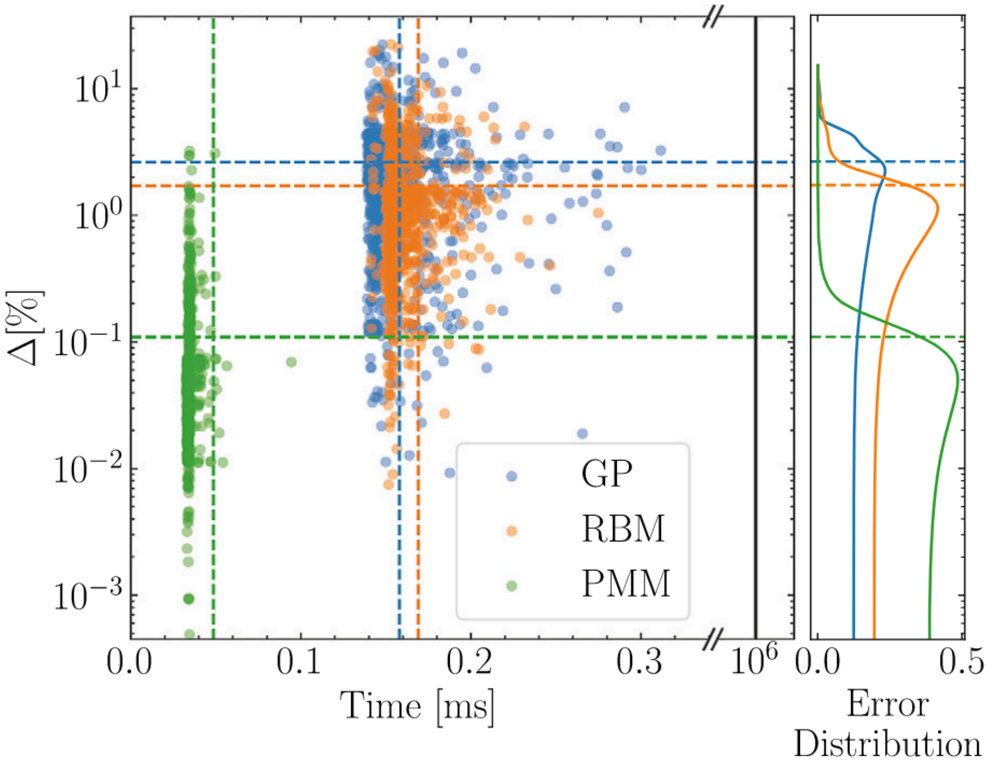



✦PMM is a universal function approximator

P. Cook, D. Jammooa, M. Hjorth-Jensen, D.D. Lee, D. Lee, Parametric matrix models, arXiv:2401.11694

# Applications of PMM

## AFDMC calculations for Deuteron




emulator errors of only ≈ 0.1%

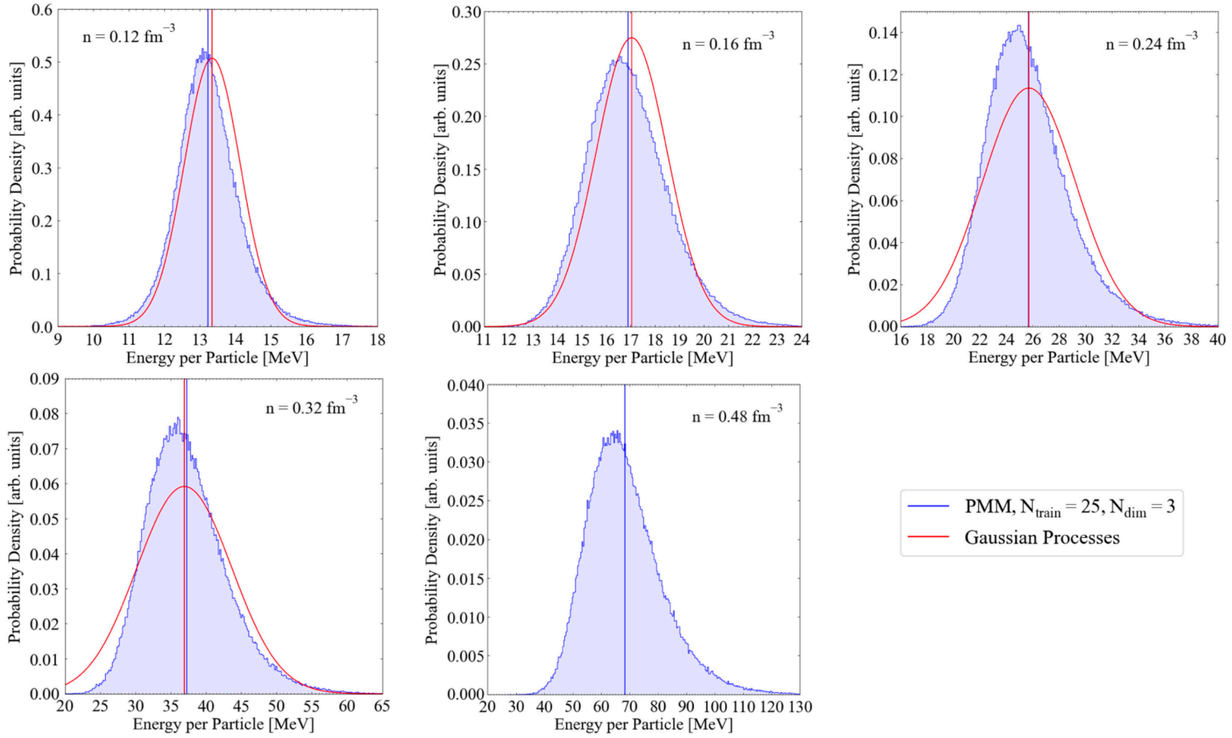
speed-up factors of ≈ $10^7$

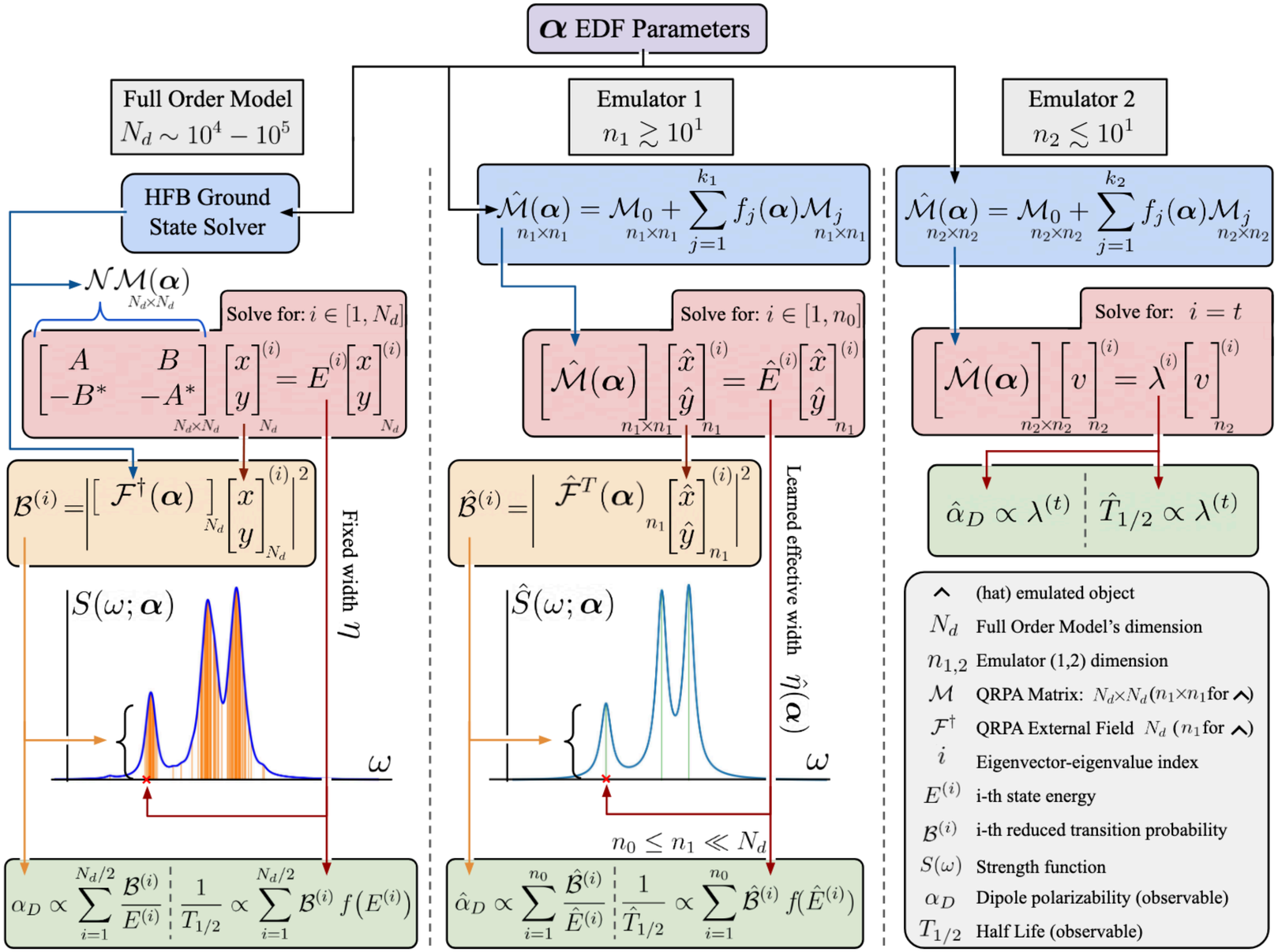
R. Somasundaram et al., Phys. Lett. B 866 (2025) 139558



**Emulators for Scarce and Noisy Data: Application to Auxiliary-Field Diffusion Monte Carlo for Neutron Matter**

Cassandra L. Armstrong,[1] Pablo Giuliani,[2] Kyle Godbey,[2] Rahul Somasundaram,[3,4] and Ingo Tews[3,*]






# PMM for linear response

L. Jin et al. arXiv:2510.13989

# Summary

- **Emulator+Bayesian inference form a fast, external, and uncertainty-aware accelerator for modern nuclear theory.**

- Other approaches besides emulator?

Physical Review Letters



# Fast Parameter Inference on Pulsar Timing Arrays with Normalizing Flows

PDF

Share

David Shih [1], Marat Freytsis [2], Stephen R. Taylor [3], Jeff A. Dror [4,5], and Nolan Smyth [5]

Show more

# Acknowledgments

**Collaborators**:

- Lie-Wen Chen (SJTU)
- Bao-Jun Cai, Yu-Gang Ma, Kai-Jia Sun (FDU)
- Mengying Qiu, Cen-Xi Yuan, Jun Zeng (SYSU)
- Che-Ming Ko (TAMU)
- Rui Wang (INFN)

Thanks for your attentions