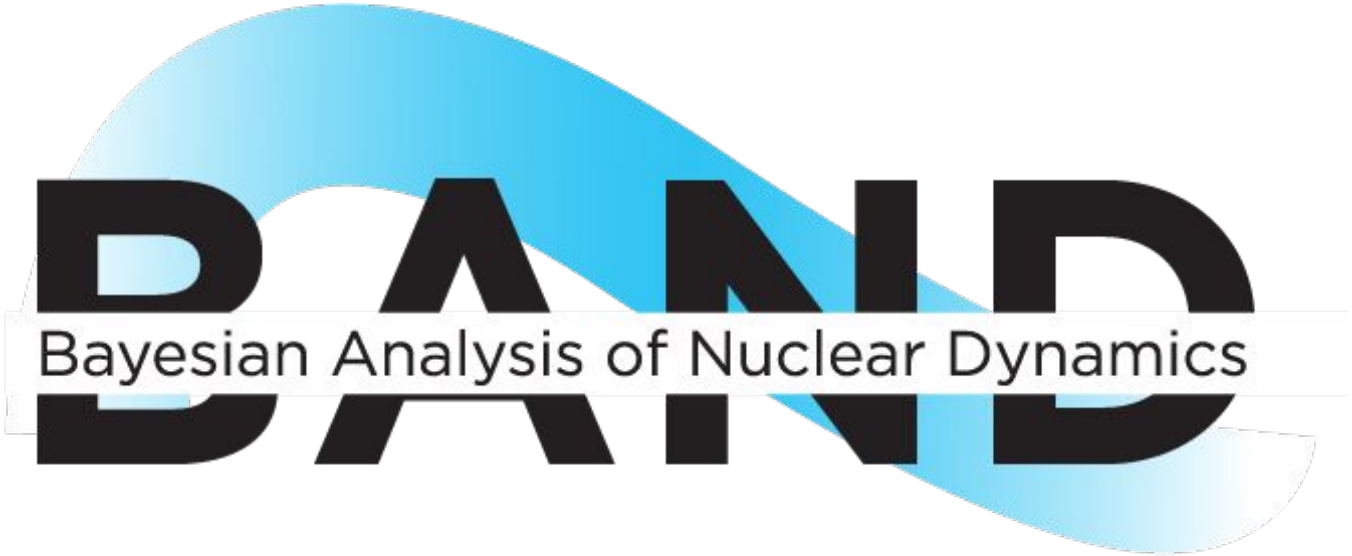
The BAND Software Framework

Kyle Godbey

Slides with videos:
https://docs.google.com/presentation/d/1XdE1vG8VEe97BCzEaW8fgwQkZimLL2gJcubDXwzl4XE/edit?usp=sharing

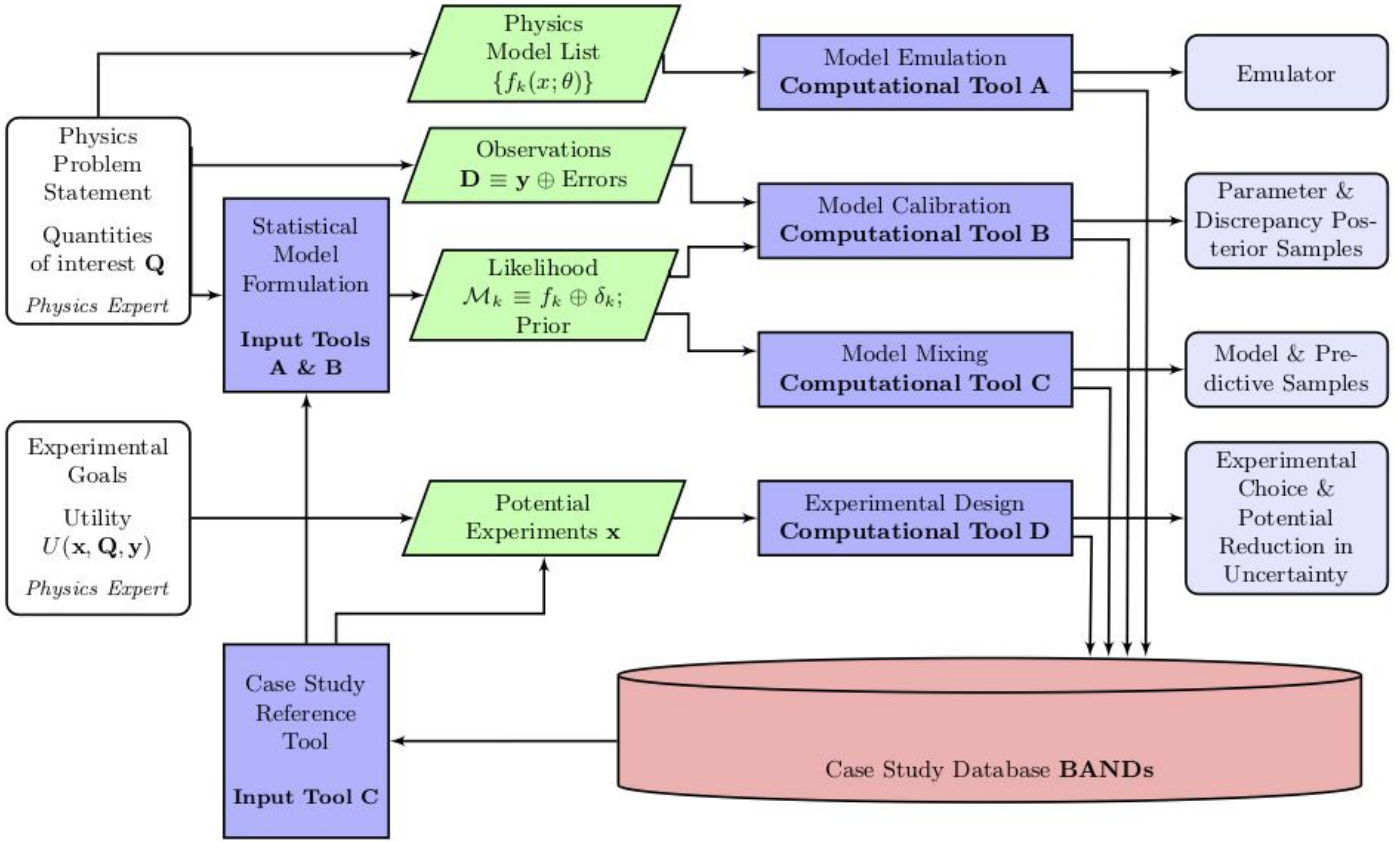
WANDA | 2024 WANDA Meeting
Arlington, VA

Introducing: BAND

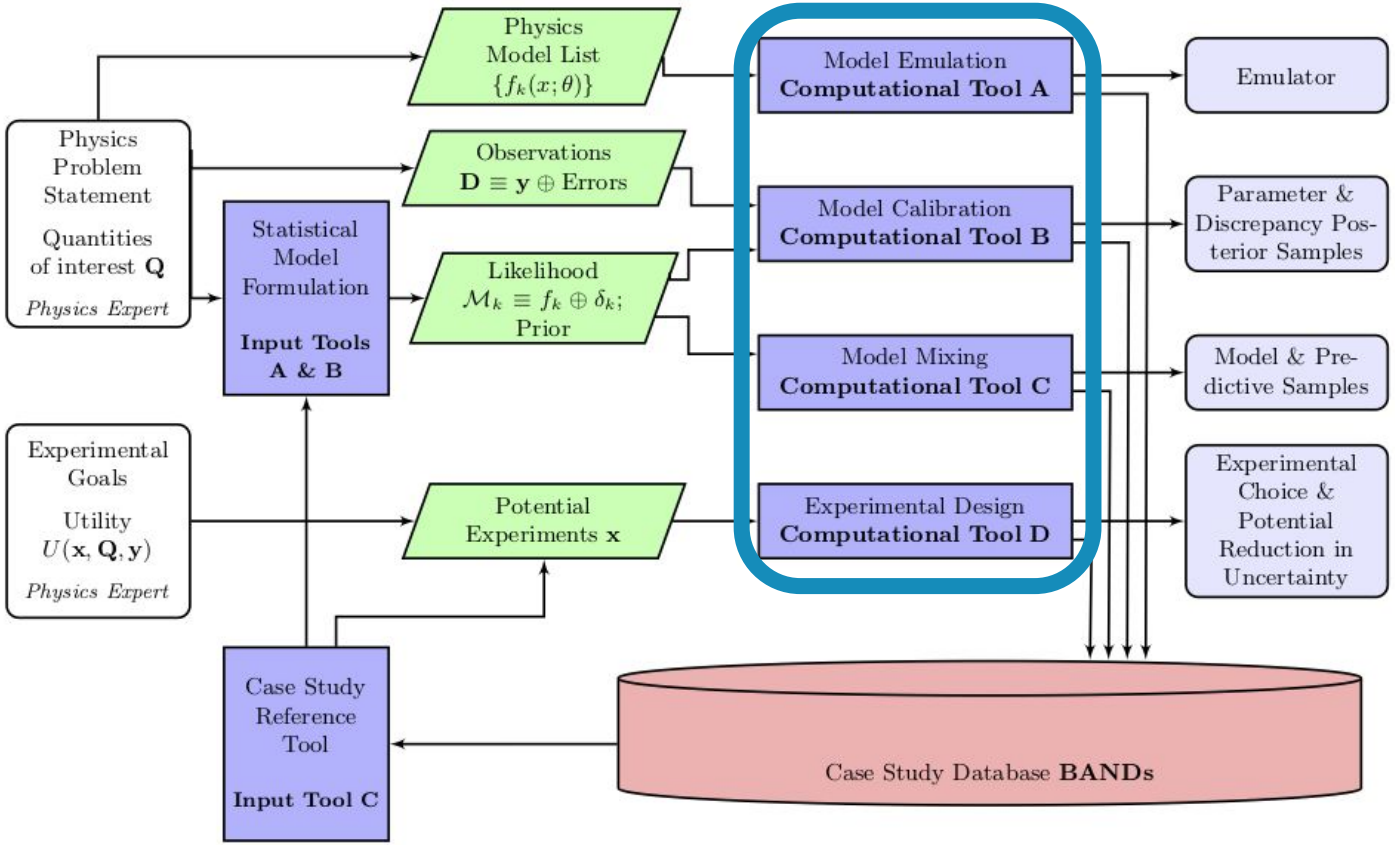


Supported by the NSF CSSI program under grant OAC-2004601

The Framework



The Framework



The Framework

Online:
https://bandframework.github.io/

On GitHub:
https://github.com/bandframework/bandframework

Software

External code delivery will be from the bandframework github repository

surmise

A Python package designed to provide a surrogate model interface for calibration, uncertainty quantification, and other tools.

O. Surer, M. Plumlee, S.M. Wild, M. Y-H. Chan
surmise Read the Docs

Taweret

A versatile Python package containing multiple model mixing techniques for a variety of use cases.

K. Ingles, D. Liyanage, A. C. Semposki, J. C. Yannotty
Taweret documentation

SAMBA

The SAndbox for Mixing using Bayesian Analysis, developed as a testing ground for multivariate model mixing on a toy model setup.

A. C. Semposki, R. J. Furnstahl, D. R. Phillips
SAMBA repository

ParMOO

ParMOO is a parallel multiobjective optimization solver that seeks to exploit simulation-based structure in objective and constraint functions.

T.H. Chang, S.M. Wild, H. Dickinson
parmoo Read the Docs

BMEX

The Bayesian Mass Explorer (BMEX) is a user-focused web application that provides a one-stop-shop for quantified theoretical model predictions of nuclear masses and related quantities.

K. Godbey, L. Buskirk, P. Giuliani
BMEX Web Application

ROSE

The Reduced Order Scattering Emulator (ROSE) is a Python package for building emulators using reduced basis methods for calculating nuclear scattering observables for user-defined interactions, including optical potentials.

D. Odell, P. Giuliani, K. Godbey, K. Beyer, M. Y. Chan
ROSE Github

The Focus

ROSE – A scattering emulator for fast theoretical simulation without sacrificing accuracy

Use cases include model calibration, quick evaluation, efficient model mixing, experimental design, science gateways, and more (presumably)

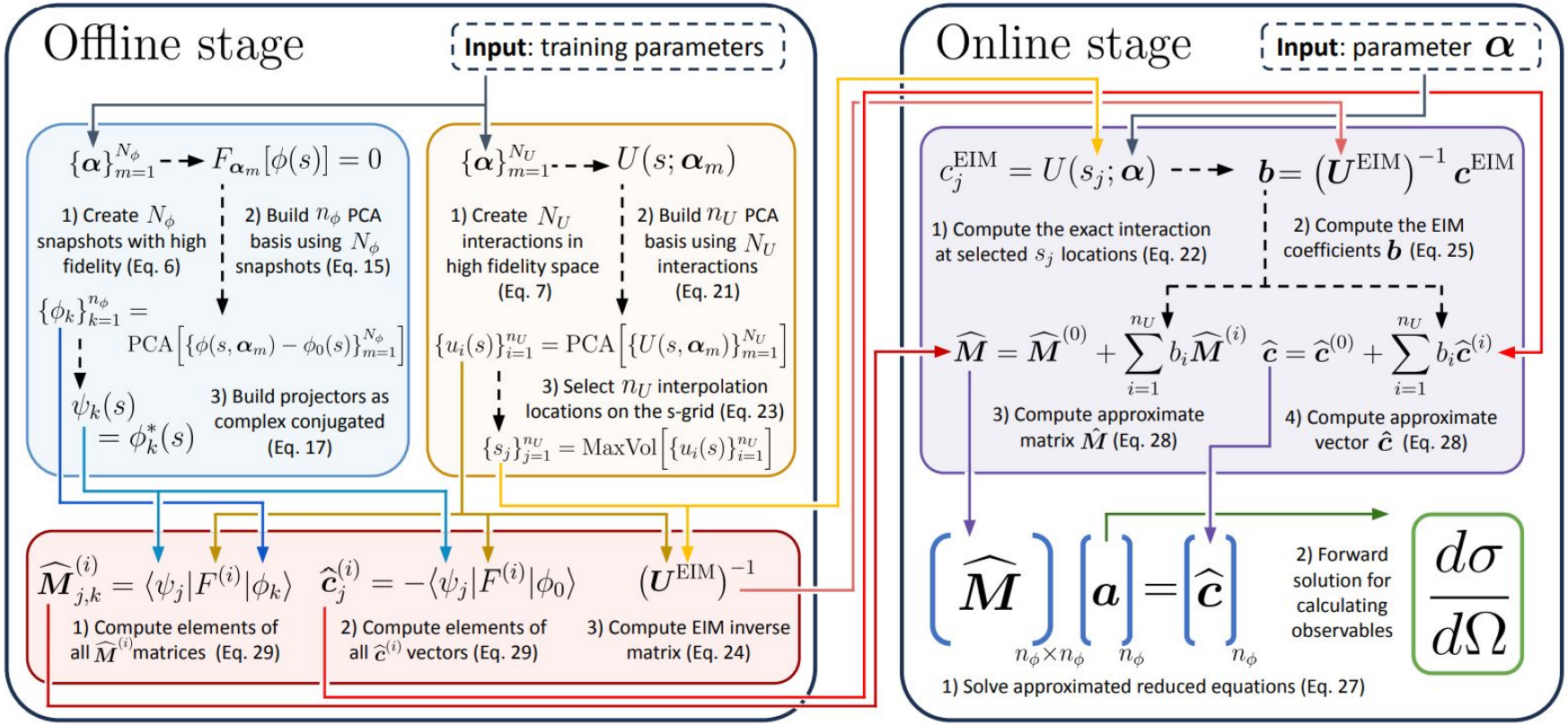
ROSE

The Reduced Order Scattering Emulator (ROSE) is a Python package for building emulators using reduced basis methods for calculating nuclear scattering observables for user-defined interactions, including optical potentials.

D. Odell, P. Giuliani, K. Godbey, K. Beyer, M. Y. Chan

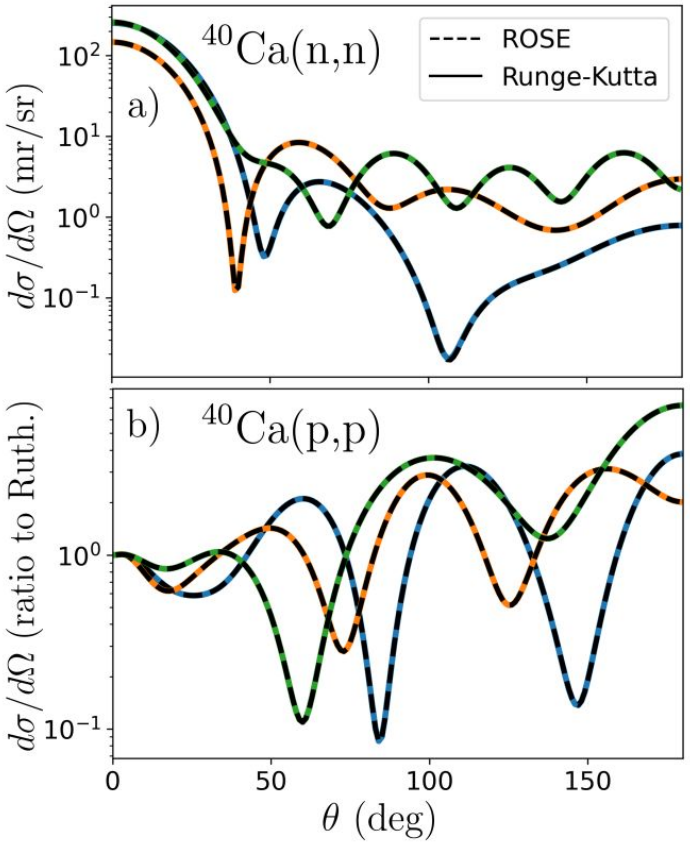
ROSE Github

The Focus

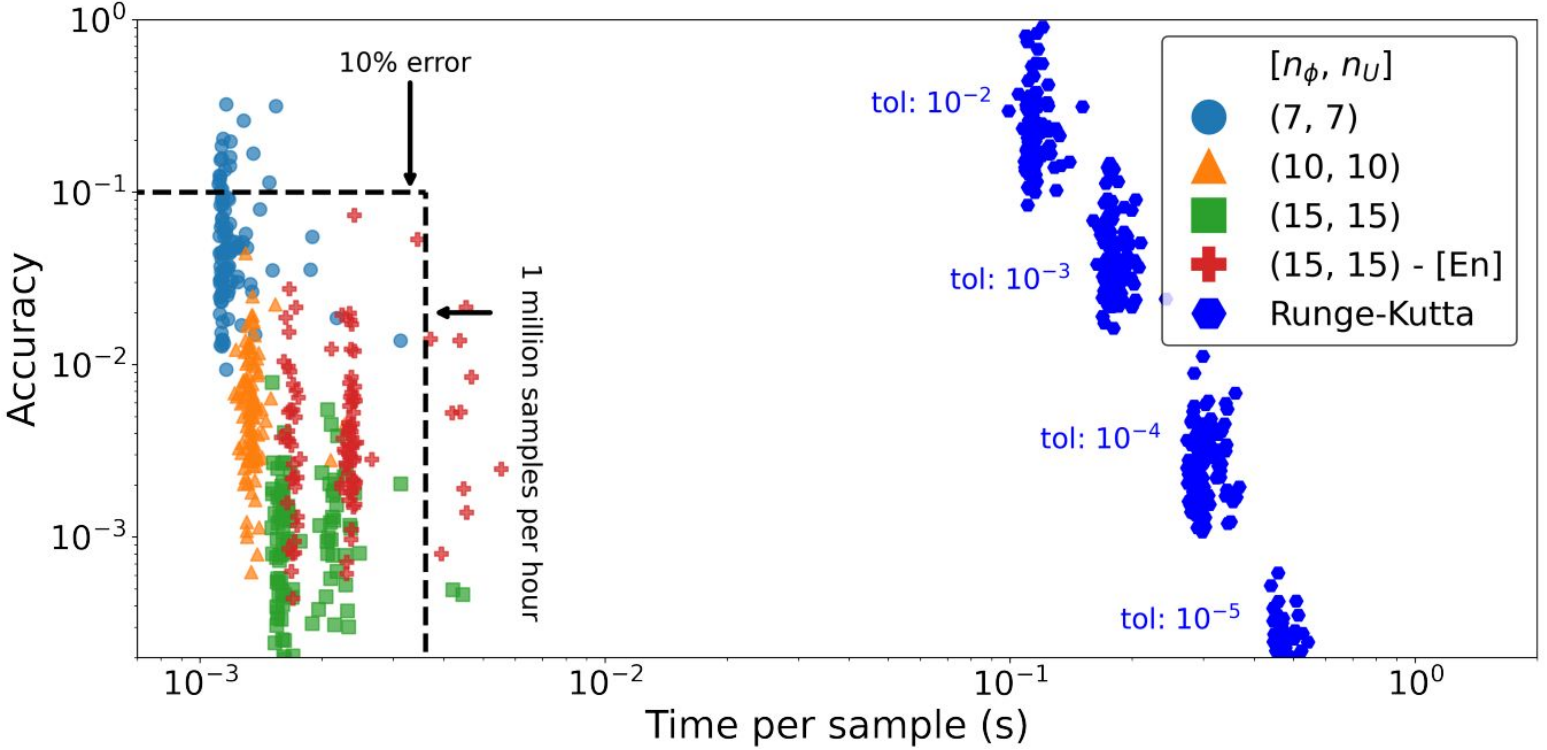

Offline stage
Input: training parameters
$\{\boldsymbol{\alpha}\}_{m=1}^{N_\phi} \dashrightarrow F_{\boldsymbol{\alpha}_m}[\phi(s)] = 0$
1) Create N_ϕ snapshots with high fidelity (Eq. 6)
2) Build n_ϕ PCA basis using N_ϕ snapshots (Eq. 15)
$\{\phi_k\}_{k=1}^{n_\phi} = \mathrm{PCA}\left[\{\phi(s,\boldsymbol{\alpha}_m) - \phi_0(s)\}_{m=1}^{N_\phi}\right]$
$\psi_k(s) = \phi_k^*(s)$
3) Build projectors as complex conjugated (Eq. 17)
$\{\boldsymbol{\alpha}\}_{m=1}^{N_U} \dashrightarrow U(s;\boldsymbol{\alpha}_m)$
1) Create N_U interactions in high fidelity space (Eq. 7)
2) Build n_U PCA basis using N_U interactions (Eq. 21)
$\{u_i(s)\}_{i=1}^{n_U} = \mathrm{PCA}\left[\{U(s,\boldsymbol{\alpha}_m)\}_{m=1}^{N_U}\right]$
3) Select n_U interpolation locations on the s-grid (Eq. 23)
$\{s_j\}_{j=1}^{n_U} = \mathrm{MaxVol}\left[\{u_i(s)\}_{i=1}^{n_U}\right]$
$\widehat{\boldsymbol{M}}_{j,k}^{(i)} = \langle\psi_j|F^{(i)}|\phi_k\rangle$
1) Compute elements of all $\widehat{M}^{(i)}$ matrices (Eq. 29)
$\hat{\boldsymbol{c}}_j^{(i)} = -\langle\psi_j|F^{(i)}|\phi_0\rangle$
2) Compute elements of all $\hat{c}^{(i)}$ vectors (Eq. 29)
$\left(\boldsymbol{U}^{\mathrm{EIM}}\right)^{-1}$
3) Compute EIM inverse matrix (Eq. 24)
Online stage
Input: parameter $\boldsymbol{\alpha}$
$c_j^{\mathrm{EIM}} = U(s_j;\boldsymbol{\alpha}) \dashrightarrow \boldsymbol{b} = \left(\boldsymbol{U}^{\mathrm{EIM}}\right)^{-1}\boldsymbol{c}^{\mathrm{EIM}}$
1) Compute the exact interaction at selected s_j locations (Eq. 22)
2) Compute the EIM coefficients $\boldsymbol{b}$ (Eq. 25)
$\widehat{\boldsymbol{M}} = \widehat{\boldsymbol{M}}^{(0)} + \sum_{i=1}^{n_U} b_i \widehat{\boldsymbol{M}}^{(i)}$
3) Compute approximate matrix $\hat{\boldsymbol{M}}$ (Eq. 28)
$\hat{\boldsymbol{c}} = \hat{\boldsymbol{c}}^{(0)} + \sum_{i=1}^{n_U} b_i \hat{\boldsymbol{c}}^{(i)}$
4) Compute approximate vector $\hat{\boldsymbol{c}}$ (Eq. 28)
$\widehat{\boldsymbol{M}}\ \boldsymbol{a} = \hat{\boldsymbol{c}}$
$n_\phi \times n_\phi$ n_ϕ n_ϕ
1) Solve approximated reduced equations (Eq. 27)
2) Forward solution for calculating observables
$\frac{d\sigma}{d\Omega}$

The Goal

Supplementing high-fidelity simulations with ‘almost as good’ replacements

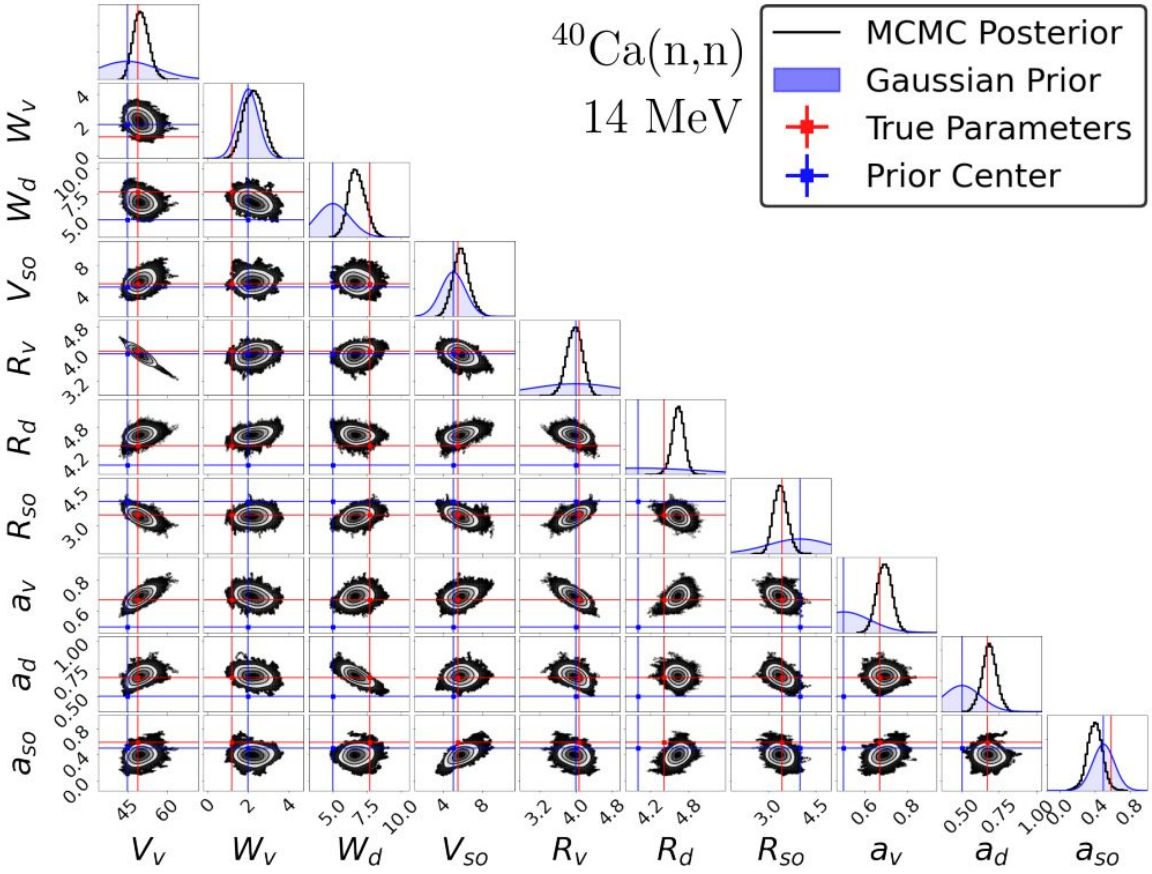


The Goal


10% error
1 million samples per hour
Accuracy
Time per sample (s)
tol: 10^{-2}
tol: 10^{-3}
tol: 10^{-4}
tol: 10^{-5}
$[n_\phi, n_U]$
(7, 7)
(10, 10)
(15, 15)
(15, 15) - [En]
Runge-Kutta

The Outcome

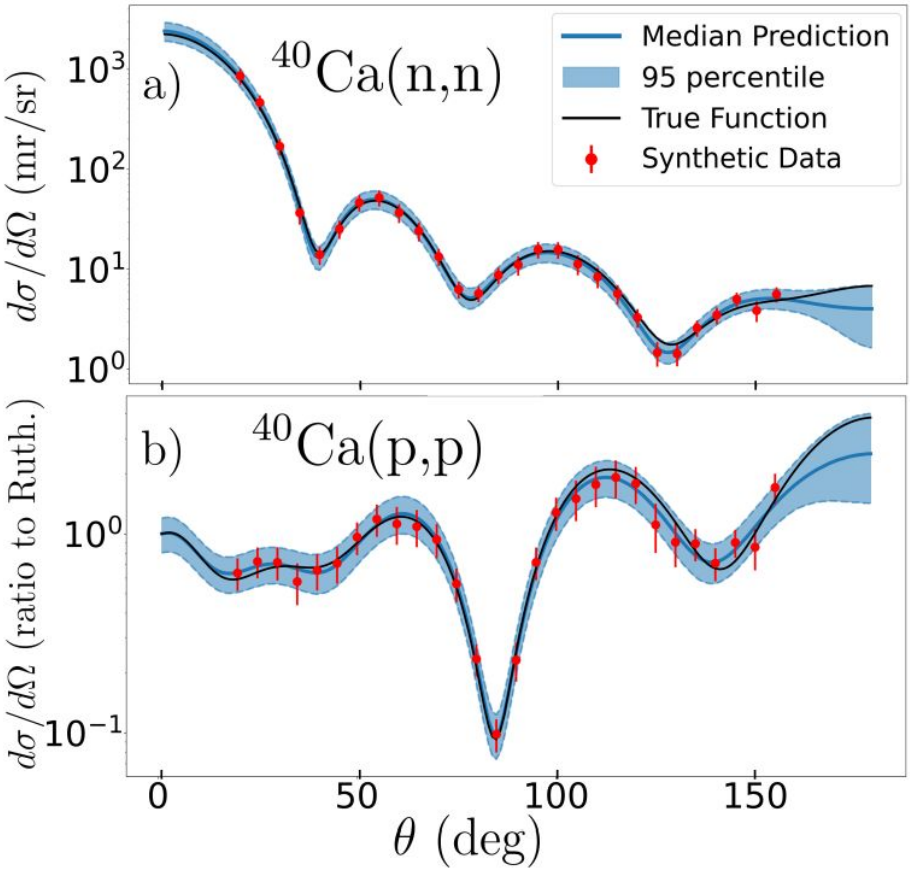
Principled Bayesian calibration is more accessible than ever before



The Outcome

The point is to get predictions, uncertainties, and covariances quickly

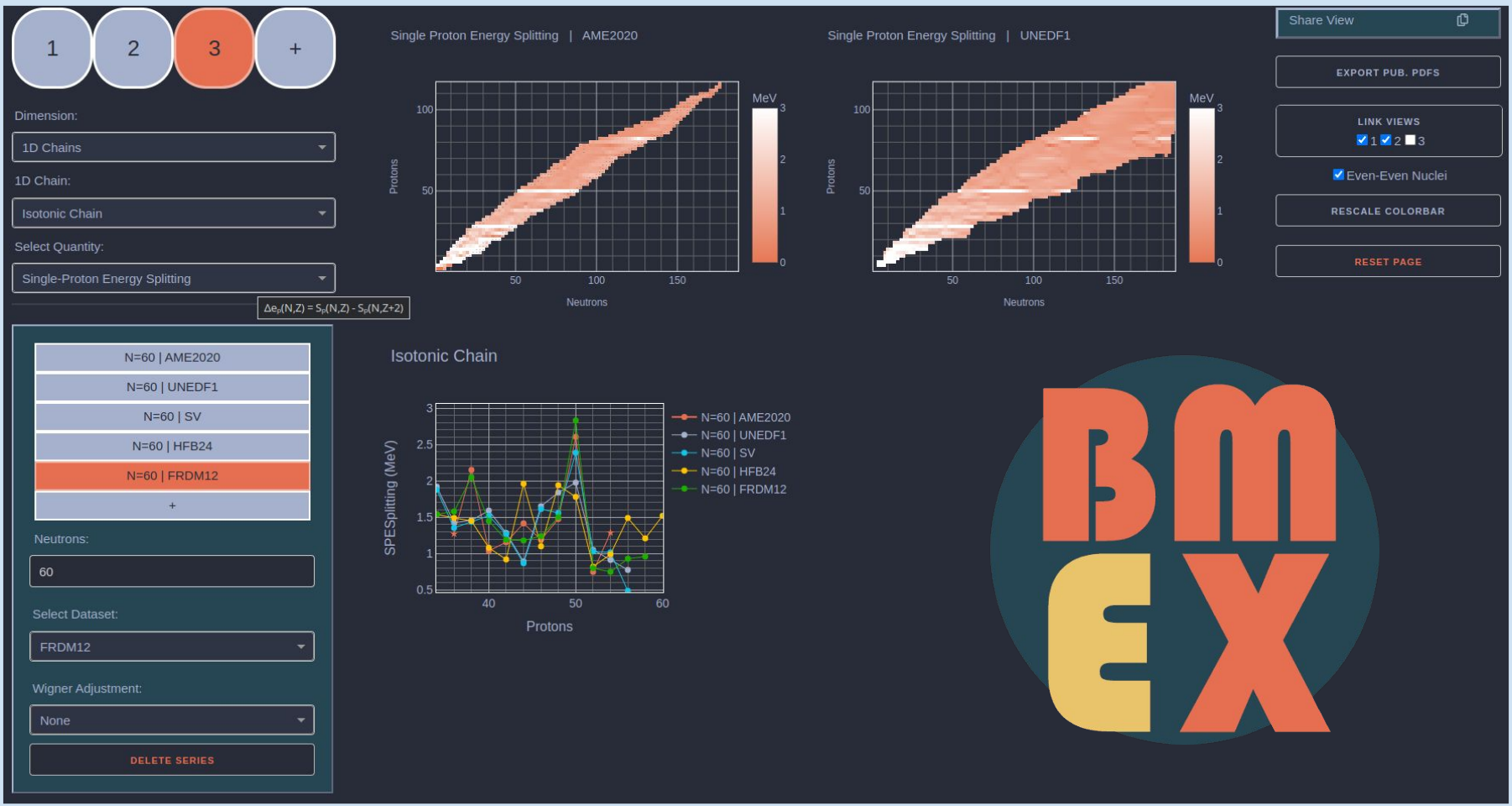
Integration with model mixing tools will help consider the wisdom of many models



BAND
Bayesian Analysis of Nuclear Dynamics

Outlook

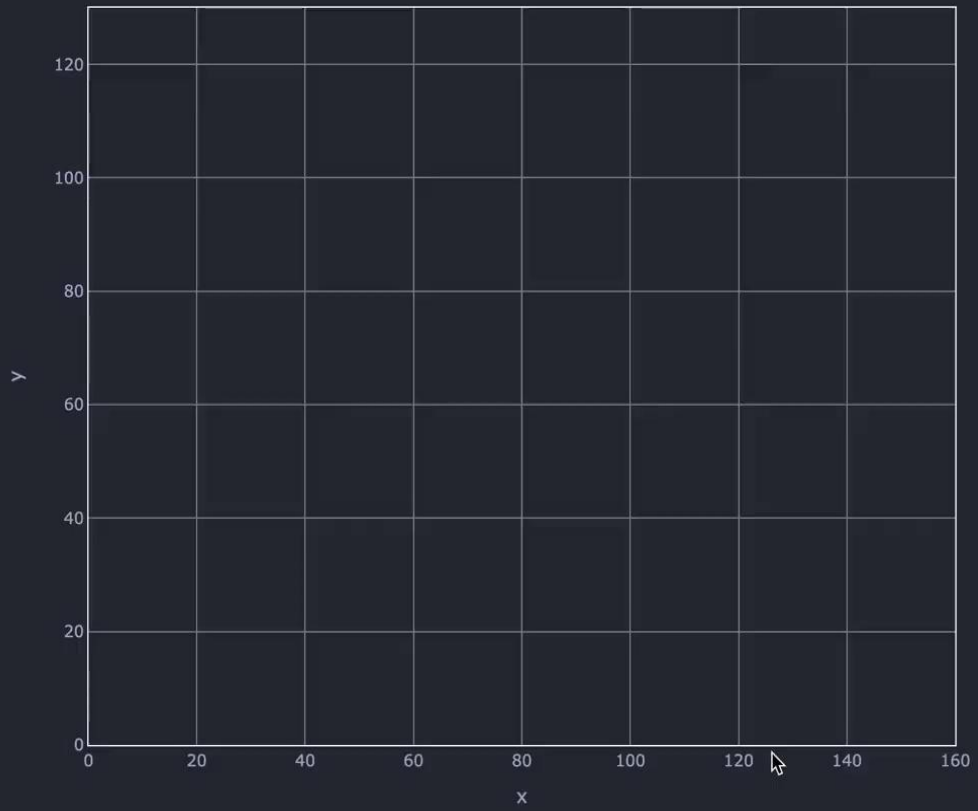
- Continued feature development in both physics and emulation
  - Coupled-channels emulation
  - Black-box dimensionality reduction
- Expanded integration within and without the BAND framework
  - Bayesian model mixing
- Explore novel applications and user interaction modality
  - Cloud-enabled backends
  - Web applications


1
2
3
+
Dimension:
1D Chains
1D Chain:
Isotonic Chain
Select Quantity:
Single-Proton Energy Splitting
$\Delta e_p(N,Z) = S_p(N,Z) - S_p(N,Z+2)$
N=60 | AME2020
N=60 | UNEDF1
N=60 | SV
N=60 | HFB24
N=60 | FRDM12
+
Neutrons:
60
Select Dataset:
FRDM12
Wigner Adjustment:
None
DELETE SERIES
Single Proton Energy Splitting | AME2020
Protons
Neutrons
MeV
Single Proton Energy Splitting | UNEDF1
Protons
Neutrons
MeV
Share View
EXPORT PUB. PDFS
LINK VIEWS
1 2 3
Even-Even Nuclei
RESCALE COLORBAR
RESET PAGE
Isotonic Chain
SPESplitting (MeV)
Protons
N=60 | AME2020
N=60 | UNEDF1
N=60 | SV
N=60 | HFB24
N=60 | FRDM12
BM
EX

FIT

CLEAR

	x	y	δy
x			



Our Request

Continued feature development in both physics and emulation

What features are needed? What models are highest impact?

Expanded integration within and without the BAND framework

What does harmonious integration look like?

Explore novel applications and user interaction modality

What **tools** have the highest potential for impact across the pipeline?

Conclusion

We want to build a broadly useful, open-source framework that meets the needs of the community

To that end we welcome suggestions, feedback, critiques, requests, code, etc.