Bayesian methods for uncertainty quantification and model mixing in the precision era of nuclear physics

Alexandra C. Semposki

2026 FRIB-TA Topical Program

6 July 2026

GRETA photo credit: Stephen Gillespie

The Ohio State University

INSTITUTE OF NUCLEAR & PARTICLE PHYSICS INPP

OHIO UNIVERSITY

The “precision era” of nuclear physics

Need **quantified uncertainties** to understand how wrong our models are and how to improve our results

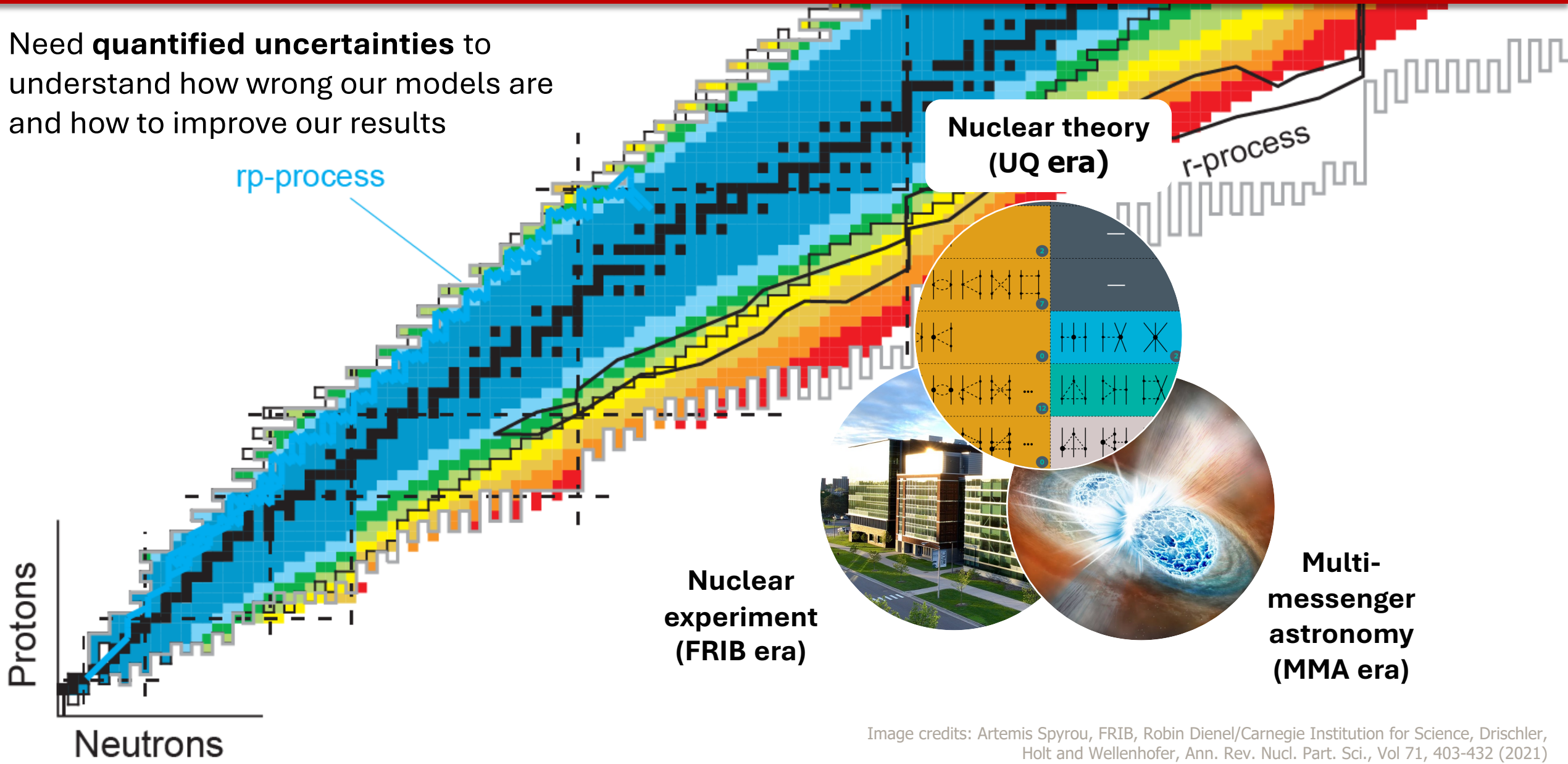


Image credits: Artemis Spyrou, FRIB, Robin Dienel/Carnegie Institution for Science, Drischler, Holt and Wellenhofer, Ann. Rev. Nucl. Part. Sci., Vol 71, 403-432 (2021)

Some introduction: Bayesian statistics

Bayes theorem: *everything* is a probability distribution!

Distribution represents the **degree of belief** of our results

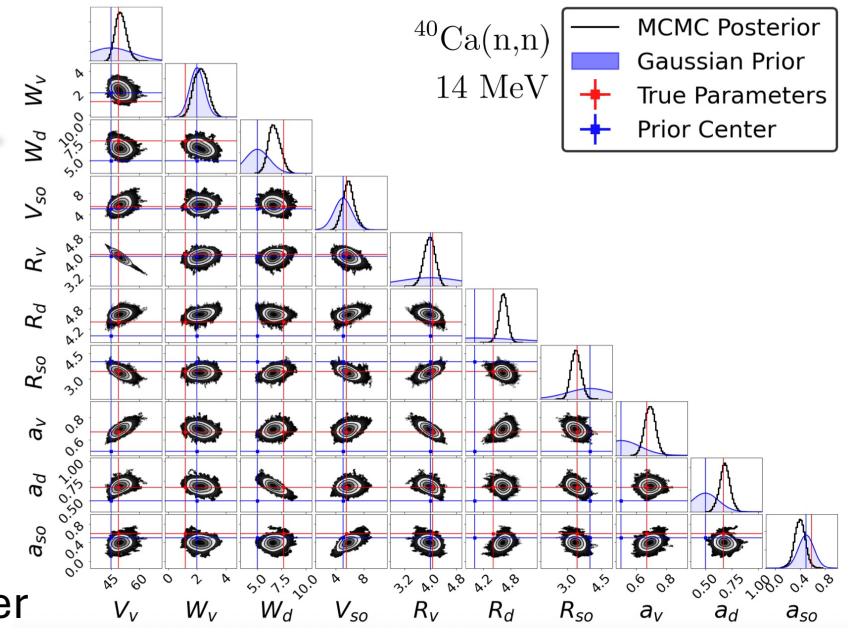
Use it to find distribution of parameters:

Likelihood function

Posterior $\longrightarrow$ $p(\boldsymbol{\theta}|\mathbf{D}) \propto p(\mathbf{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ $\longleftarrow$ Prior on desired parameters

Parameter estimation of an **optical potential!**

D. Odell et al. PRC **109** 044612 (2024)



And now, for something completely different—suppose you have an **experimental result** and **model prediction** that have various uncertainties to account for:

$$Y_{exp}(x) + \varepsilon_{exp}(x) = f(x) = Y_{model}(x,\theta) + \varepsilon_{trunc}(x) + \varepsilon_{MD}(x)$$

+ parameter uncertainties!

Experimental result + error

Underlying theory

Theory prediction + truncation error + model discrepancy

How do we deal with all these uncertainties?

What kinds of Bayesian tools can we use?

Type of problem	Bayesian and/or ML tools
Experimental systematics	Build statistical model
Lots of data but lack of well-behaved models	Model discrepancy, model mixing, neural networks
Models with lots of unknown parameters	Model calibration
Model is missing physics and does not fit well to data	Model mixing, model discrepancy, model discovery
Discrepant models for the same physics	Model mixing/combination
Lack of models/data to describe observable over range	Model mixing, Gaussian processes
Expensive model evaluations	Emulator development
Inverse problems	Emulators, neural networks

See **Pablo's talk** tomorrow!

Recap: model calibration

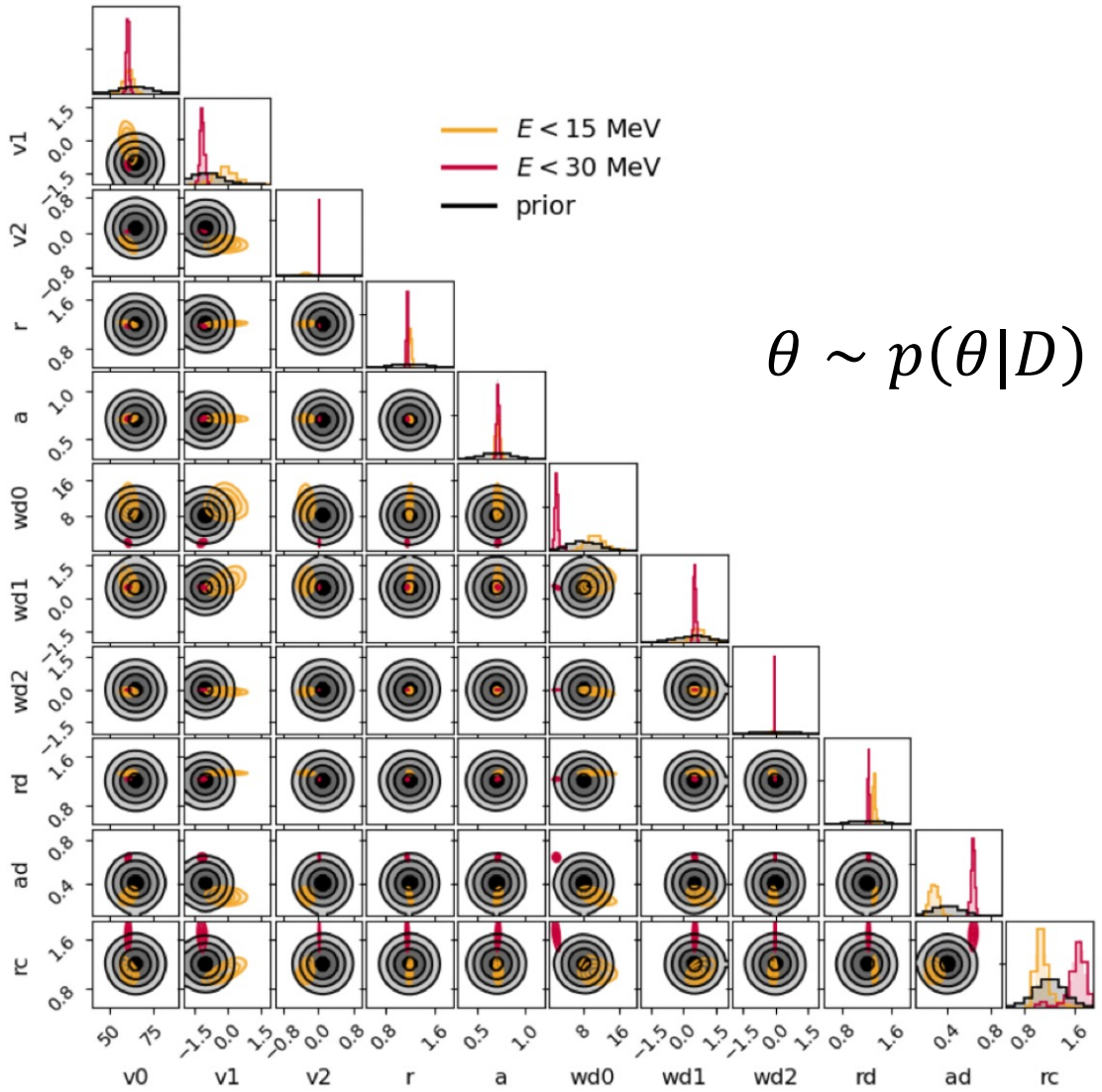
How do I fit my model to data so that it explains the data in the most accurate way?

Select a model with unknown parameters to fit to the data (assume this model is correct)*

Define priors over your parameters, e.g., Gaussian, truncated Gaussian, uniform, etc.

Define likelihood for the data (usually a chi-squared but the form can change—**ask Kyle**!)

Sample the posteriors using MCMC methods



$$\theta \sim p(\theta|D)$$

Crash course: Gaussian processes (GPs)

"Set of random variables, any subset of which possesses a Gaussian distribution"

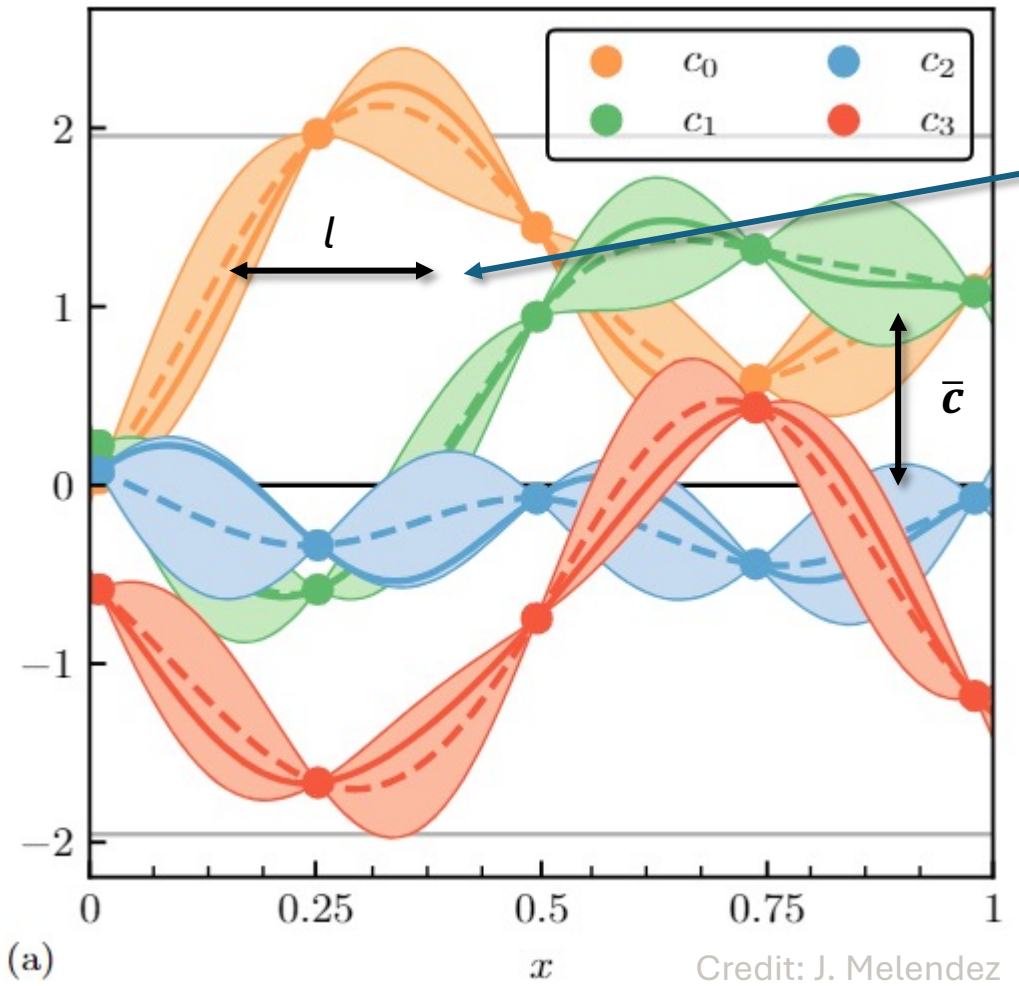
Less abstract:
Defined by mean function and covariance function (*kernel*)

$$f(x) \sim \mathcal{GP}[m(x), \kappa(x, x')]$$

Contains dependence on variance and lengthscale (RBF, Matérn, etc.)

Common form: squared-exponential radial basis function (RBF)

$$k_{\mathrm{RBF}}(x, x'; \ell) = \exp\left(\frac{-(x - x')^2}{2\ell^2}\right)$$



Credit: J. Melendez et al. PRC (2019)

Correlation structure parameterized by a lengthscale and marginal standard deviation

Hyperparameters that *can* be determined by Bayesian parameter estimation

Defined over *function space;* more flexible fit to data than other methods

Can be fitted to **data + uncertainties** and used to predict at new points

ε_{MD}: using GPs for model discrepancies

What if you have a model and its parameters are calibrated to data, but it is ***consistently discrepant***?

$$\underbrace{Y_{exp}(x) + \varepsilon_{exp}(x)}_{\text{Assumed known}} = f(x) = \underbrace{Y_{model}(x,\theta)}_{\text{Calibrated to the data}} + \varepsilon_{MD}(x)$$

How to model this?

Model this discrepancy using some function $\delta_{MD}(x)$

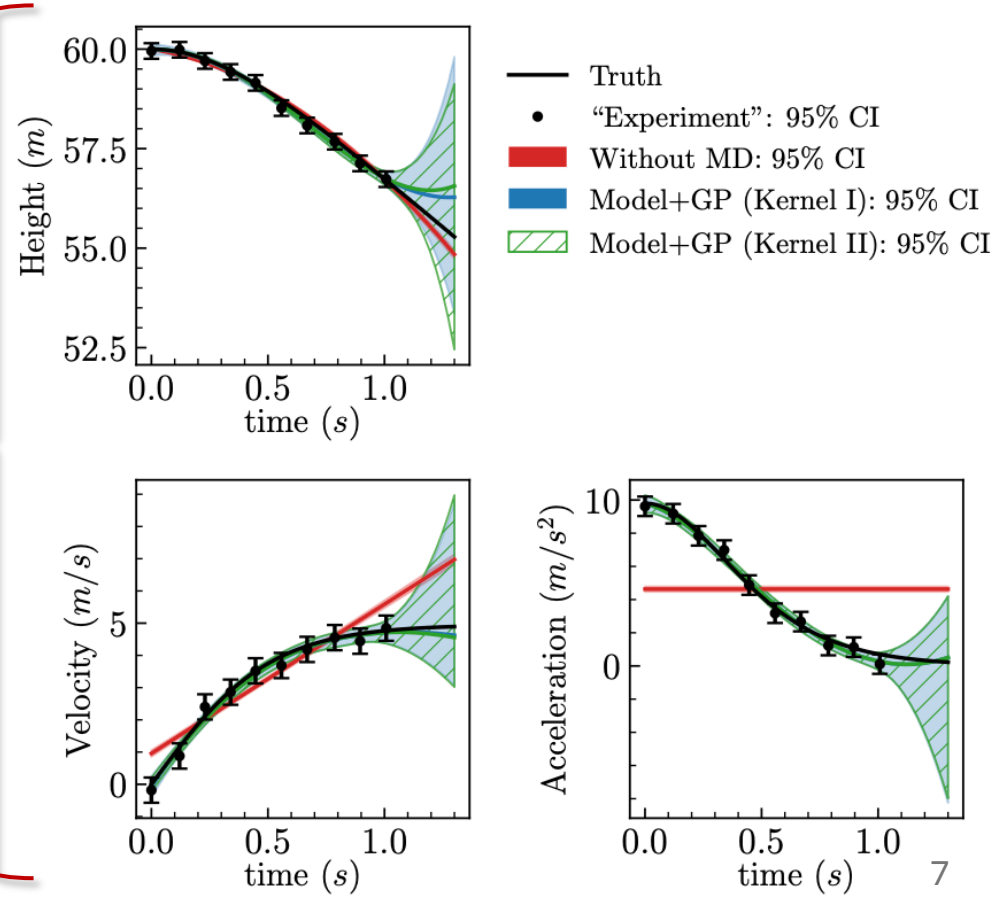
Perfect place for **GPs**! Estimate kernel based on prior knowledge of the problem:

$$\delta(\cdot \mid \boldsymbol{\phi}) \sim \mathrm{GP}\left(\mathbf{0}, K(\cdot,\cdot \mid \boldsymbol{\phi})\right)$$

Choices available:

1. Priors on the model parameters and hyperparameters of the kernel
2. GP covariance kernel **functional form**

"Ball-drop" experiment: simple toy case where **drag** is considered the discrepancy



Simultaneously infer model parameters **AND** GP hyperparameters

Original paper: S. Jaiswal et al., PLB **870**, 139946 (2025)

ε_{trunc}: model truncation error

Employ the BUQEYE truncation error model (**gsum**)

$$y_k(x) = y_{\text{ref}}(x) \sum_{n=0}^{k} c_n(x) Q^n(x)$$

Known orders of expansion

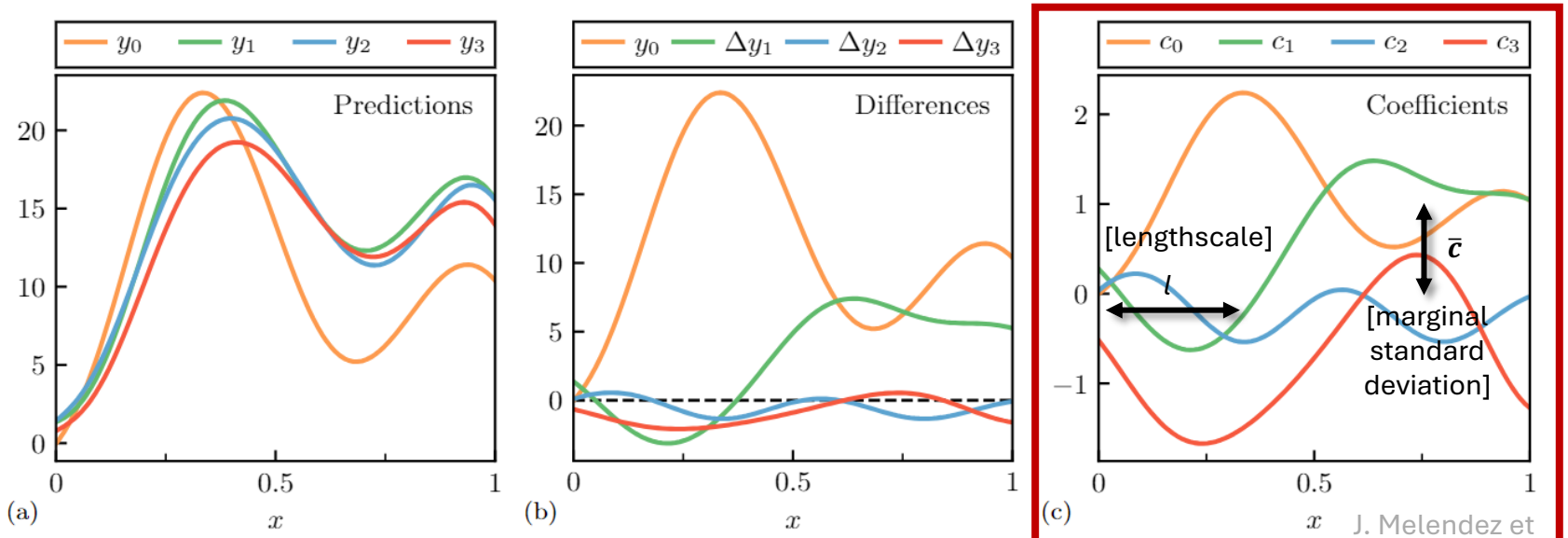
Estimations for unknown orders

$$\delta y_k(x) = y_{\text{ref}}(x) \sum_{n=k+1}^{\infty} c_n(x) Q^n(x)$$

Dimensional analysis

Natural i.i.d. curves

Expansion parameter (suppresses higher orders)



Gaussian Processes: learn coefficient distributions

Incorporates correlations in y(x), continuous model, priors constrain parameters of GP

Easily test with *diagnostics* (Mahalanobis distance, pivoted Cholesky decomposition,...)

J. A. Melendez, S. Wesolowski, R. J. Furnstahl, D. R. Phillips, M. Pratola, PRC 100, 044001 (2019)

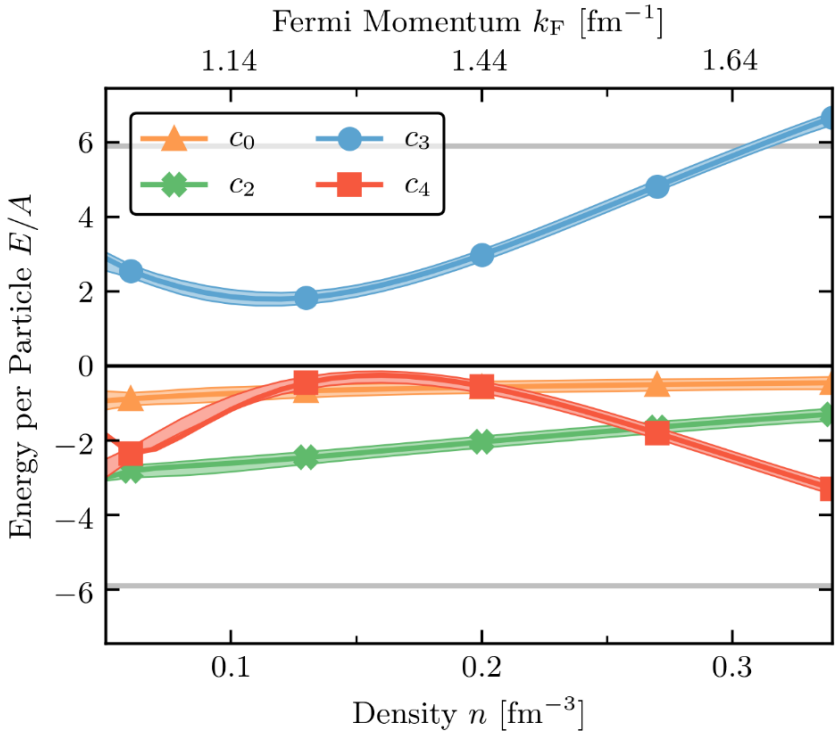
Case study: model truncation error for dense matter

Use the truncation error model to train on EFT coefficients of known orders to obtain theory uncertainties



$$Q(k_{\mathrm{F}}) = \frac{k_{\mathrm{F}}}{\Lambda_b} \quad \approx 600\ \mathrm{MeV}$$

$$y_{\mathrm{ref}}(k_{\mathrm{F}}) = 16\,\mathrm{MeV} \times \left(\frac{k_{\mathrm{F}}}{k_{\mathrm{F},0}}\right)^2$$



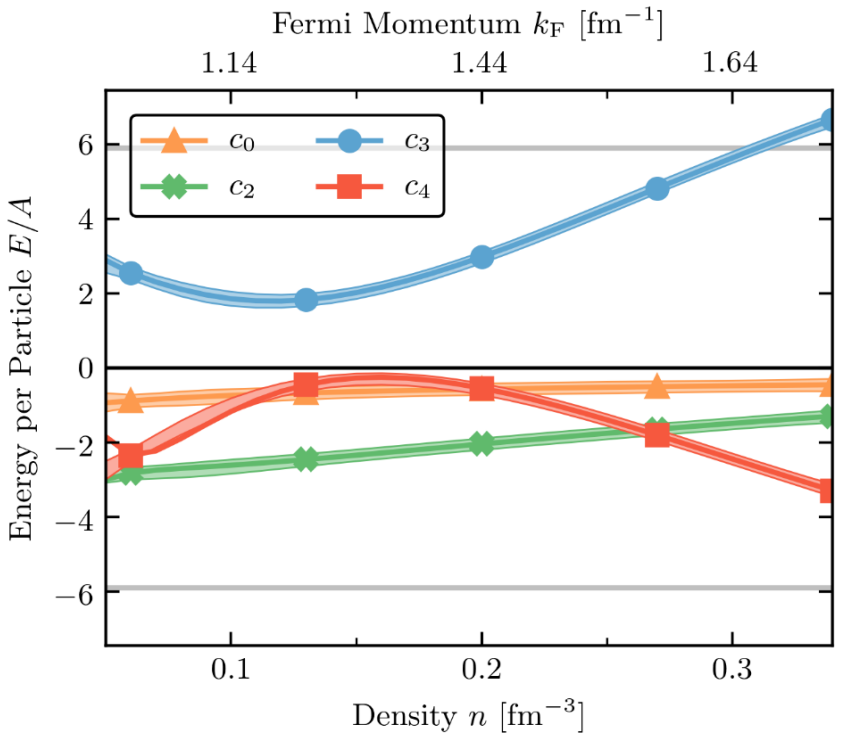
C. Drischler, R. J. Furnstahl, J. A. Melendez, D. R. Phillips, PRL **125**, 202702 (2020)

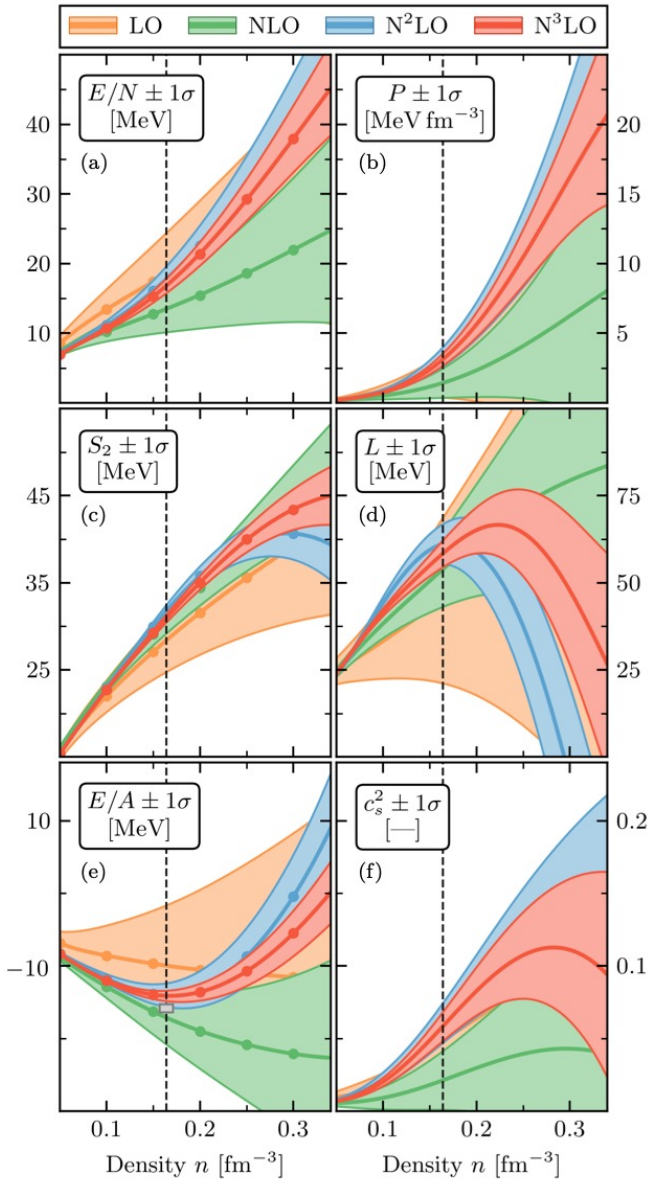
Case study: model truncation error for dense matter

Use the truncation error model to train on EFT coefficients of known orders to obtain theory uncertainties

$$Q(k_\mathrm{F}) = \frac{k_\mathrm{F}}{\Lambda_b} \quad \approx 600\ \mathrm{MeV}$$

$$y_\mathrm{ref}(k_\mathrm{F}) = 16\,\mathrm{MeV} \times \left(\frac{k_\mathrm{F}}{k_{\mathrm{F},0}}\right)^2$$





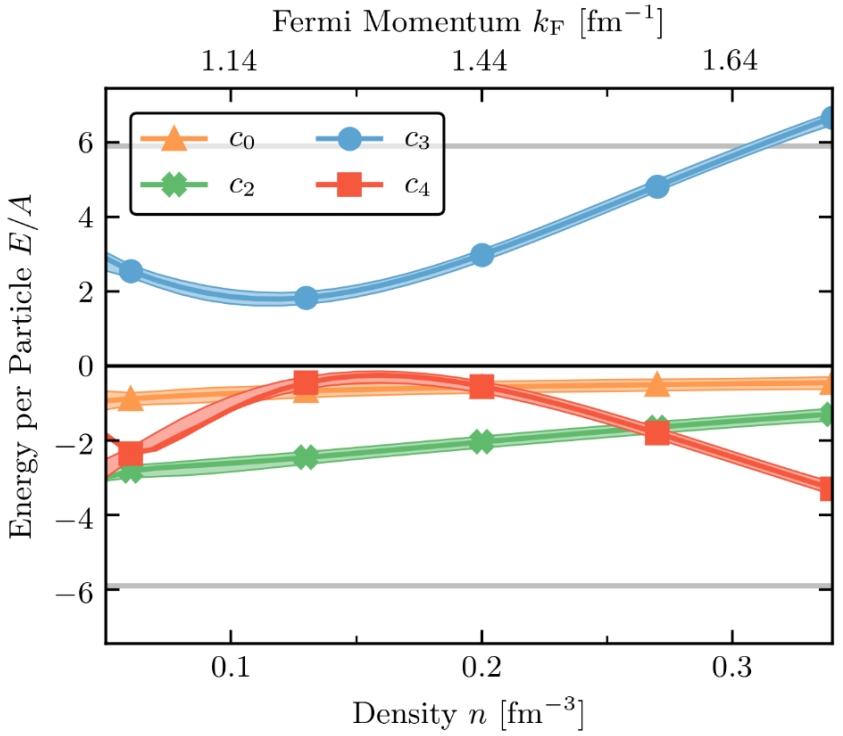
C. Drischler, R. J. Furnstahl, J. A. Melendez, D. R. Phillips, PRL **125**, 202702 (2020)

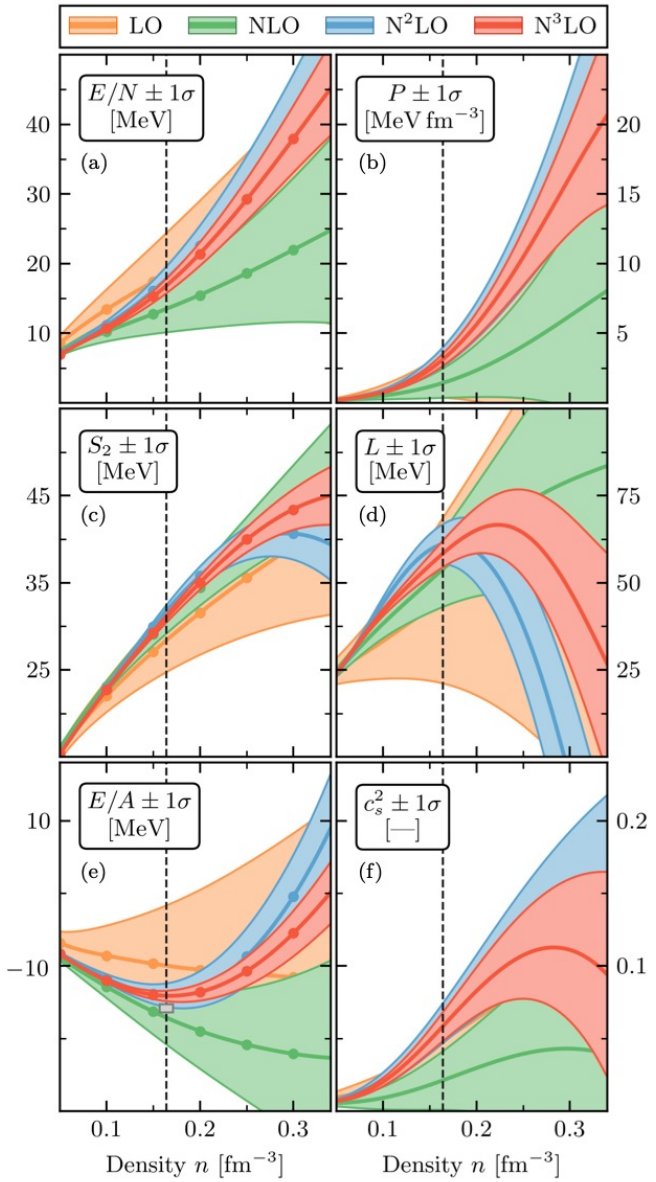
Case study: model truncation error for dense matter

Use the truncation error model to train on EFT coefficients of known orders to obtain theory uncertainties

$$Q(k_\mathrm{F}) = \frac{k_\mathrm{F}}{\Lambda_b} \quad \approx 600\ \mathrm{MeV}$$

$$y_\mathrm{ref}(k_\mathrm{F}) = 16\,\mathrm{MeV} \times \left(\frac{k_\mathrm{F}}{k_{\mathrm{F},0}}\right)^2$$





Derivatives of GPs carry UQ along with them!

$$P(n) = n^2 \frac{\mathrm{d}}{\mathrm{d}n}\frac{E}{A}(n)\ , \quad S_2(n) \approx \frac{E}{N}(n) - \frac{E}{A}(n), \text{ etc.}$$

[Further constrains the EOS and its properties]

C. Drischler, R. J. Furnstahl, J. A. Melendez, D. R. Phillips, PRL **125**, 202702 (2020)

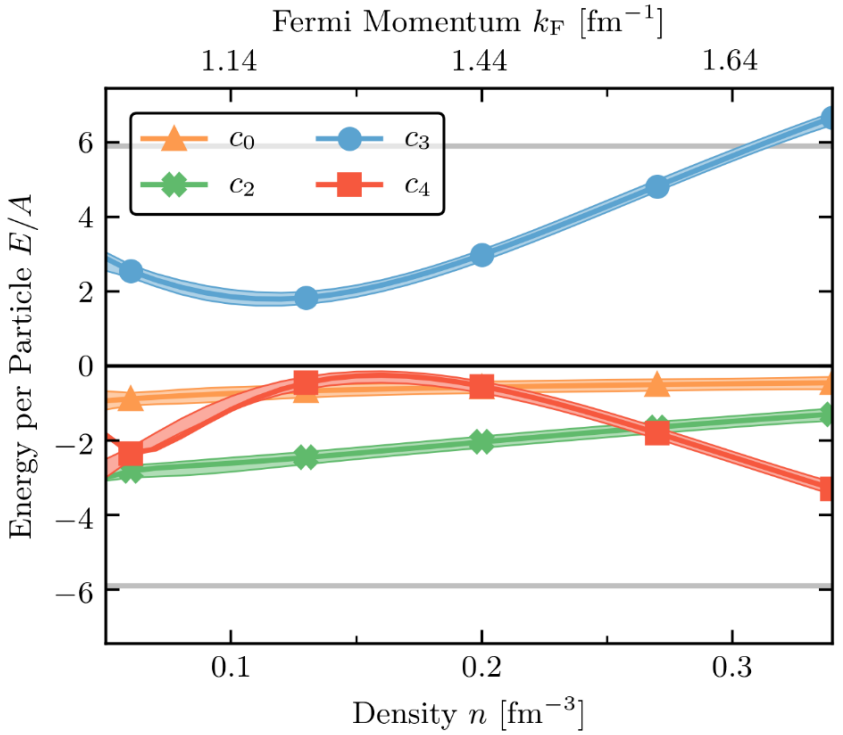
Case study: model truncation error for dense matter

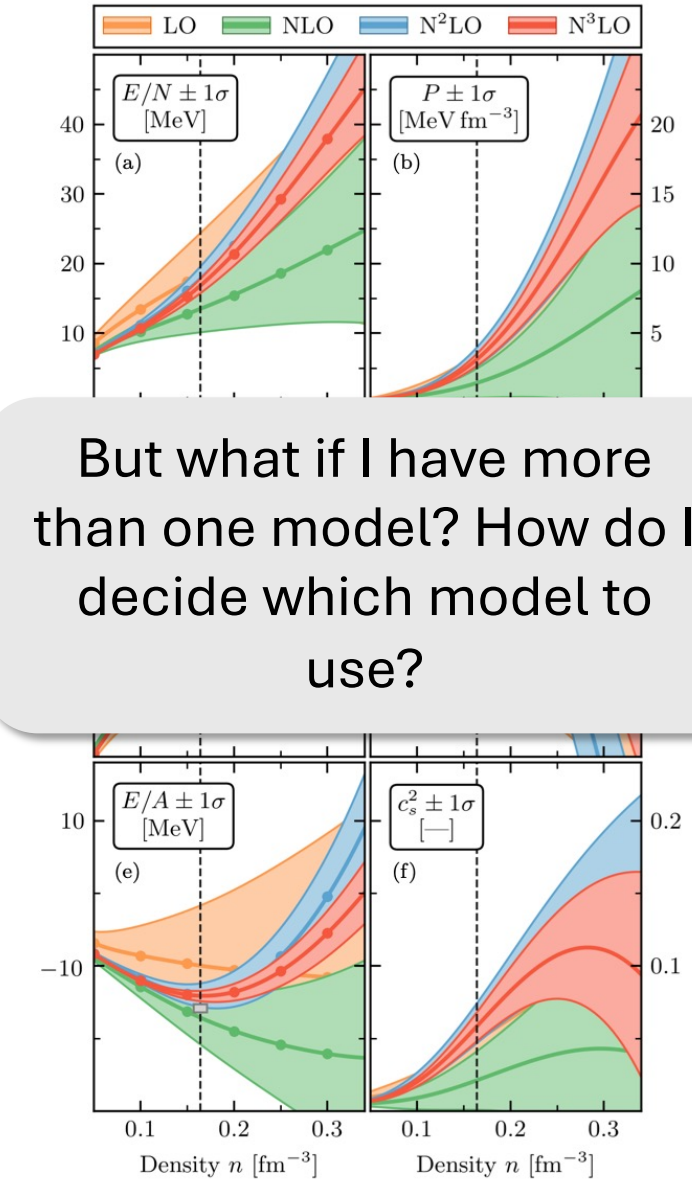
Use the truncation error model to train on EFT coefficients of known orders to obtain theory uncertainties



$$Q(k_F) = \frac{k_F}{\Lambda_b} \approx 600\ \text{MeV}$$

$$y_{\text{ref}}(k_F) = 16\,\text{MeV} \times \left(\frac{k_F}{k_{F,0}}\right)^2$$





But what if I have more than one model? How do I decide which model to use?

Derivatives of GPs carry UQ along with them!

$$P(n) = n^2 \frac{\mathrm{d}}{\mathrm{d}n}\frac{E}{A}(n)\ ,\quad S_2(n) \approx \frac{E}{N}(n) - \frac{E}{A}(n)\text{, etc.}$$

[Further constrains the EOS and its properties]

C. Drischler, R. J. Furnstahl, J. A. Melendez, D. R. Phillips, PRL **125**, 202702 (2020)

Focus: Bayesian model mixing

*How do I reliably quantify model uncertainties and make predictions using **more than one model**?*

1 Set of models: $\mathcal{M}_k : y_i = f_k(x_i; \boldsymbol{\theta}) + \varepsilon_{i,k}$ ⟶ Discrepancy term (theory error), can be modelled

2 Integrate over unknown parameters (if any exist): $p(\tilde{\mathbf{y}}|\mathbf{y}, \boldsymbol{x}, \mathcal{M}_k) = \int p(\tilde{\mathbf{y}}|\mathbf{y}, \boldsymbol{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{x}) d\boldsymbol{\theta}$

New predictions

3 Model mix with a choice of weights: $p(\tilde{\mathbf{y}}|\mathbf{y}, \boldsymbol{x}) = \sum_{k=1}^{K} \hat{w}_k(x) p(\tilde{\mathbf{y}}|\mathbf{y}, \boldsymbol{x}, \mathcal{M}_k)$

Two major types of BMM:

Density-based vs. Moment-based

Full distribution mixing

Mean and variance mixing

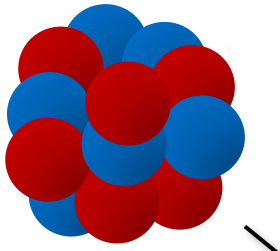
Types explored in BAND:

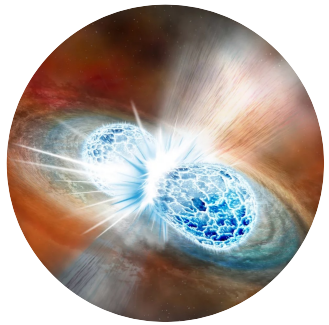
1. *Linear mixing*
2. *Regression trees*
3. **GP-based**
4. PCA model combination

BMM application: the dense matter equation of state

Interested in the **strong force** and its properties, especially in **nuclear matter** and **neutron stars**

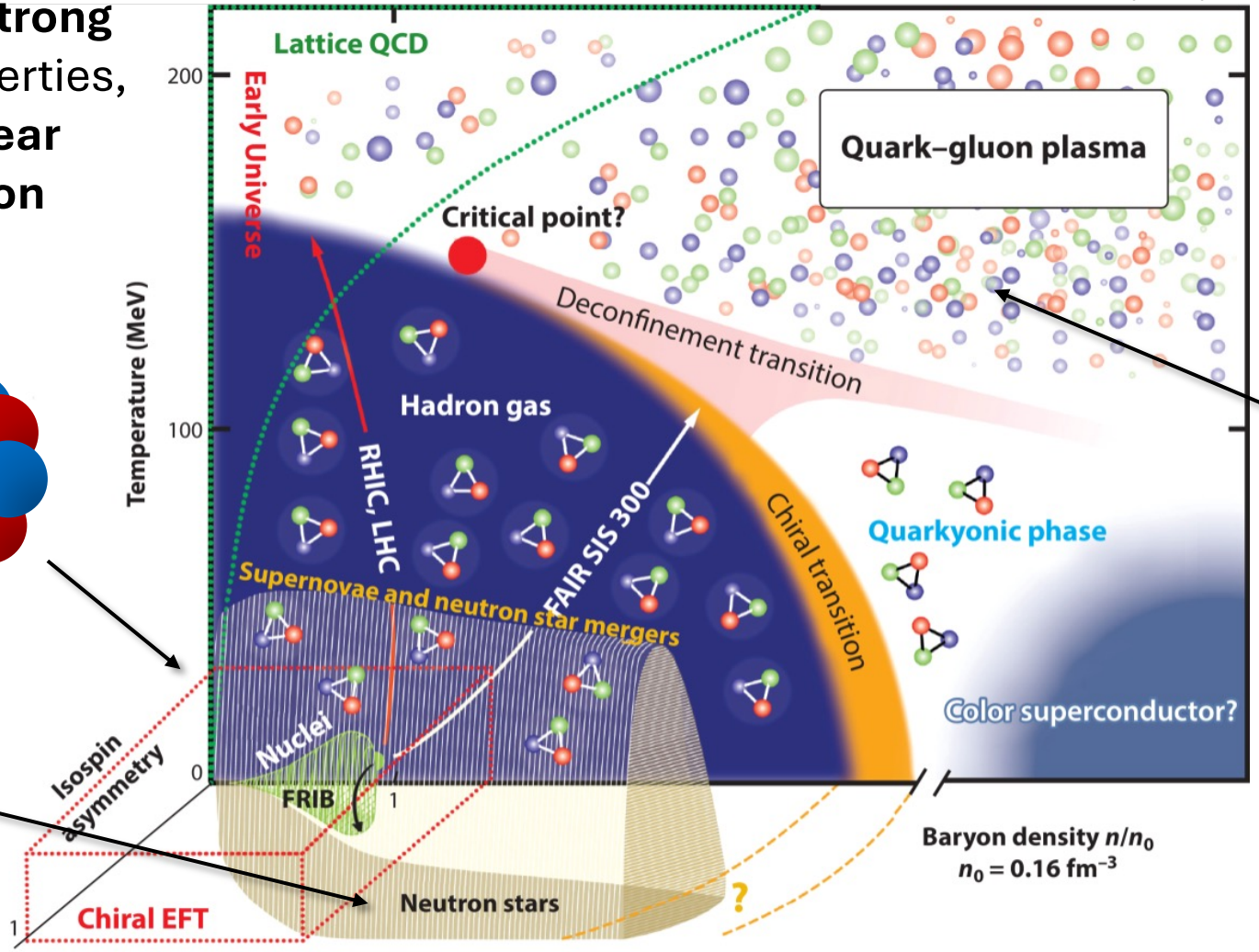
Nuclear physics





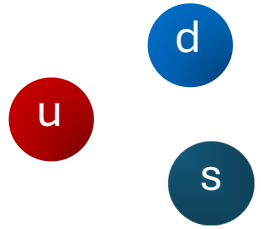
Neutron stars

C. Drischler, J. W. Holt, and C. Wellenhofer, Annu. Rev. Nucl. Part. Sci. (2021)



The **equation of state (EOS)** describes the state of matter from **nuclei** to neutron stars to **quarks and deconfinement**

Quark matter



How do we determine the unified EOS for this phase diagram?

BMM application: the dense matter equation of state

Energy density from chiral EFT:

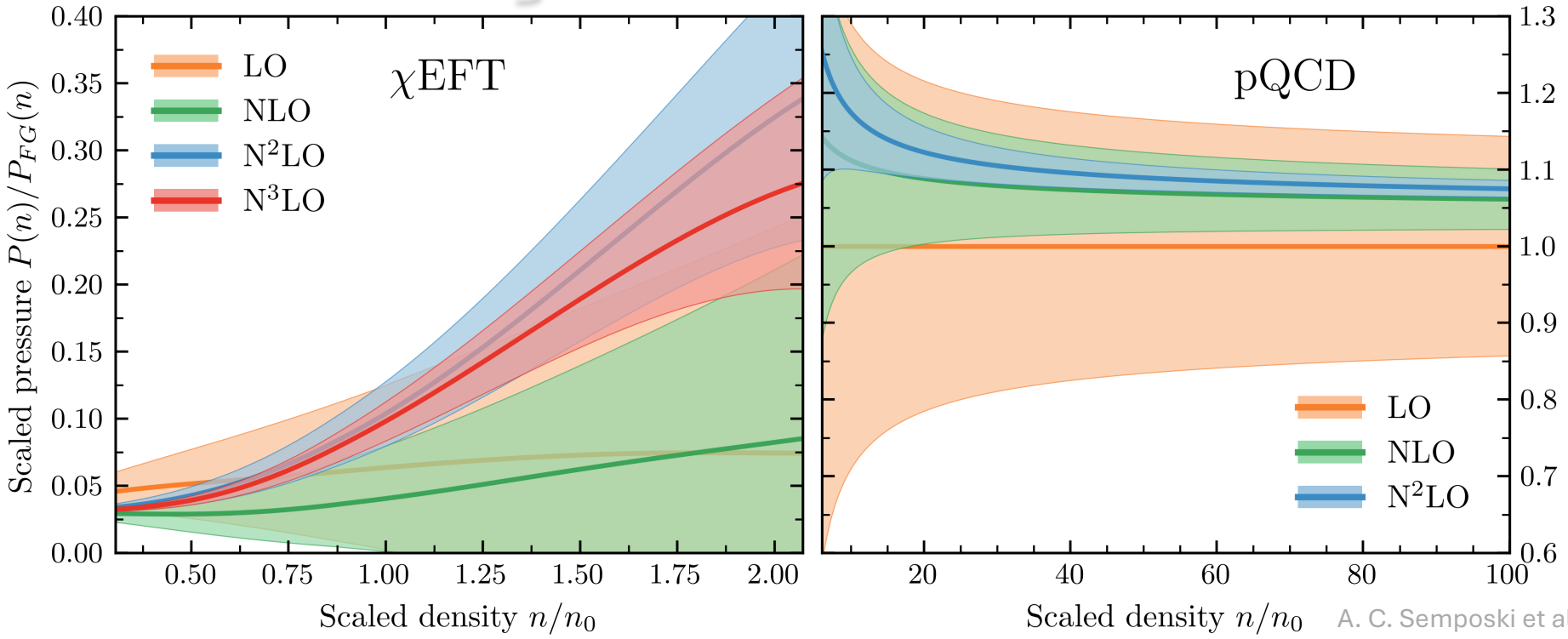
$$\varepsilon_{\rm NSM}(n_B, x, n_e, n_\mu) = (1-2x)^2 \varepsilon_{\rm PNM}(n_B) + 4x(1-x)\varepsilon_{\rm SNM}(n_B) + \varepsilon_e(n_e) + \varepsilon_\mu(n_\mu).$$

Energy per particle from symmetric matter, pure neutron matter, and the symmetry energy

pQCD: perturbative description of high-density EOS with 3 quark flavors

$$P(n) = P_{FG}(n)\left[c_0 + c_1 Q(\bar{\Lambda}_{FG}) + c_2(n) Q^2(\bar{\Lambda}_{FG}) + \ldots\right]$$

Chiral EFT uncertainties include full **density and observable correlations**



BUQEYE methods used for truncation error of both EOSs

A. C. Semposki et al. PRC **113** 015808 (2026)

How to combine these theories?

BMM application: the dense matter equation of state

1 **EOS models** written as:

$$Y^{(i)}(x) = F(x) + \delta Y^{(i)}(x) \longrightarrow \sim \mathrm{GP}[0, \kappa_y^{(i)}(x, x')]$$

Truncation, many-body error, etc.

Full covariance matrix from EOS models

2 **GP prior on underlying theory** written as:

$$F(x)|I \sim \mathcal{GP}[m(x), \kappa_f(x, x')] \longrightarrow \kappa(x, x'; \boldsymbol{\theta}) = \bar{c}^2 k(x, x'; \boldsymbol{\theta})$$

$$k_{\mathrm{RBF}}(x, x'; \ell) = \exp\left(\frac{-(x - x')^2}{2\ell^2}\right)$$

Make kernel choice

or other forms (Matérn, rational quadratic, non-stationary kernels, etc.)

3 Include physics constraints as **hyperpriors**:

$$\ell \,|\, I \sim \mathcal{U}(a_\ell, b_\ell) \times \mathcal{N}\left(\mu_\ell, \sigma_\ell^2\right)$$

or another appropriate form

Very important!

4

Assume Gaussian form and use Bayes theorem, produce posterior mean and covariance matrix

$$F_e \,|\, \{(y_t)_i\}_{i=1}^N, K_{y,tt}, I \sim \mathcal{N}[\mu_e, \Sigma_{ee}]$$

with:

$$\mu_e = K_{f,et}(K_{f,tt} + K_{y,tt})^{-1}(y_t - m_t) + m_t,$$
$$\Sigma_{ee} = K_{f,ee} - K_{f,et}(K_{f,tt} + K_{y,tt})^{-1}K_{f,te}.$$

[GP form]

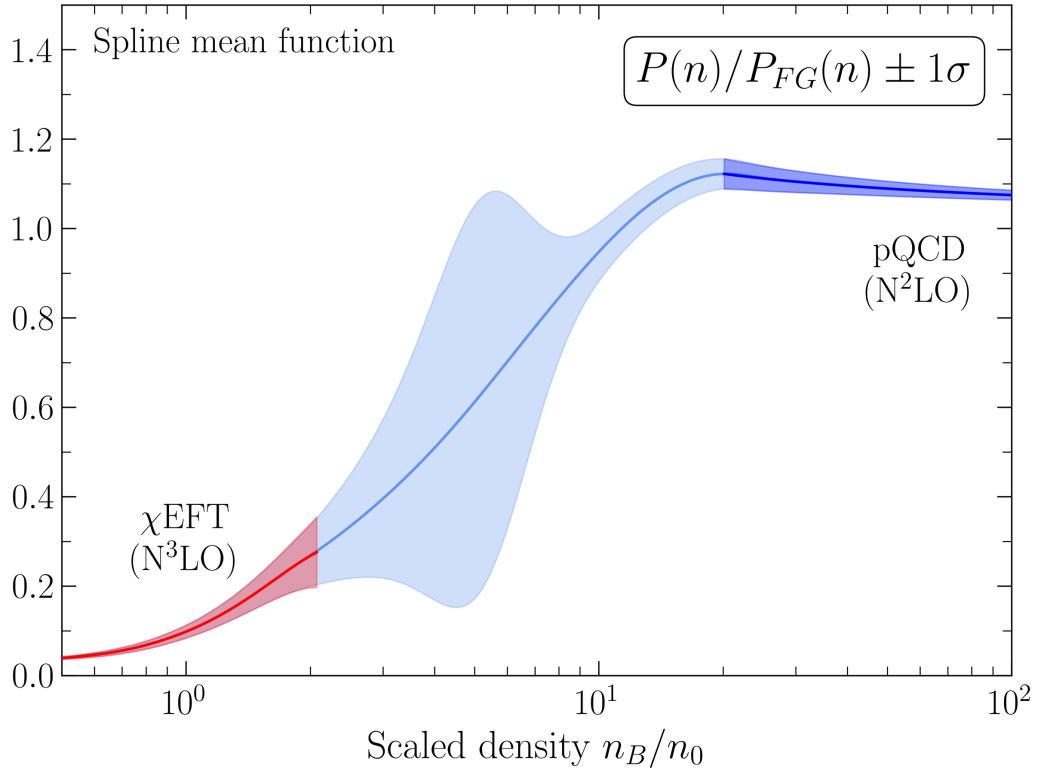
BMM application: the dense matter equation of state

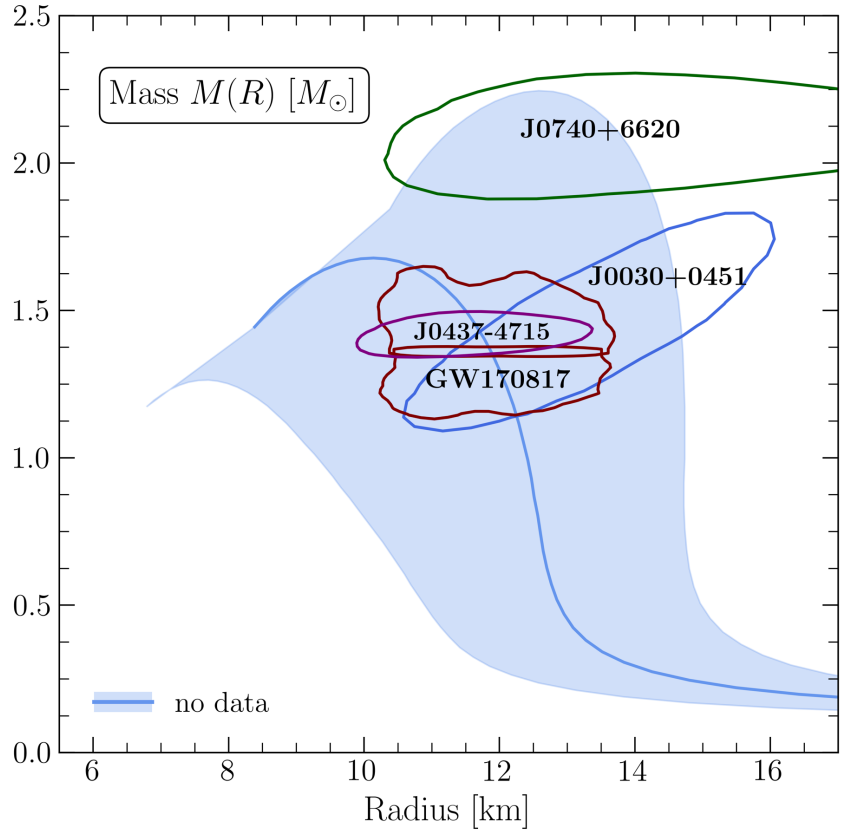
Smoothly transition between two stationary RBF kernels using a switching function

$$\kappa_f(x,x';\ell_1,\ell_2,\bar{c}_1^2,\bar{c}_2^2,\boldsymbol{\xi}) = [1-\alpha(x,\boldsymbol{\xi})] \times \kappa_1(x,x';\bar{c}_1^2,\ell_1)[1-\alpha(x',\boldsymbol{\xi})] + \alpha(x,\boldsymbol{\xi})\kappa_2(x,x';\bar{c}_2^2,\ell_2)\alpha(x',\boldsymbol{\xi})$$

$$\alpha_{\text{sigmoid}}(x;x_0,w) = \left[1+\exp\left(-\frac{(x-x_0)}{w}\right)\right]^{-1}$$

Use a "greedy approach" to sequentially add heavy-ion data and future model predictions to the EOS





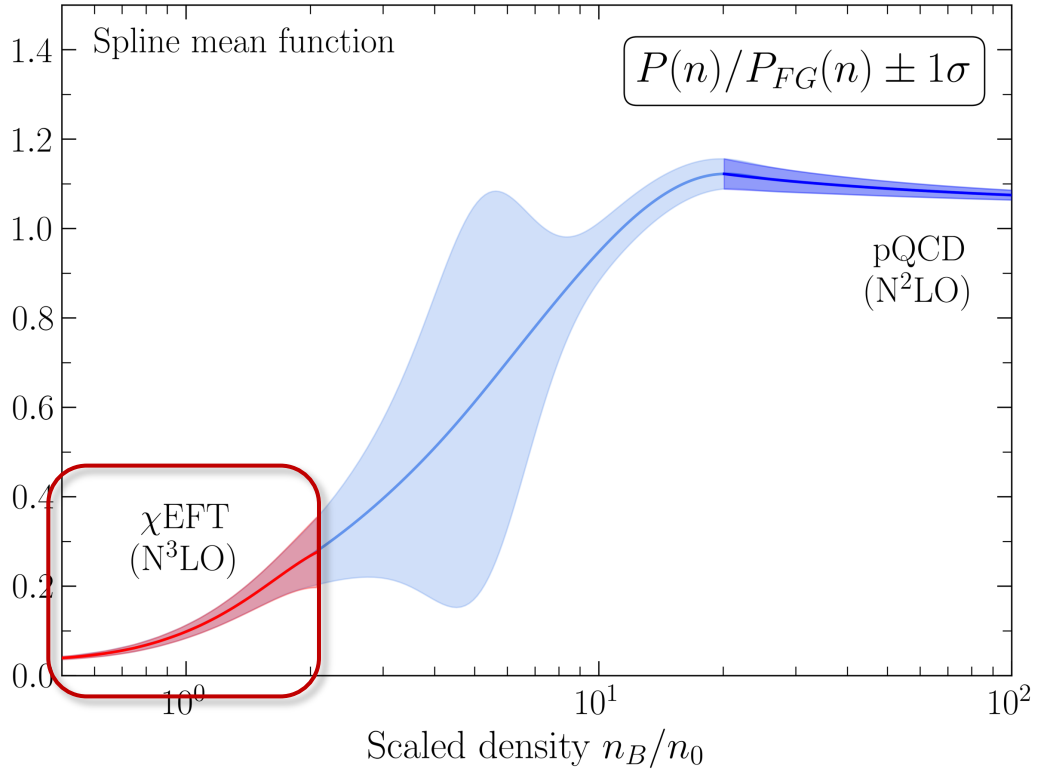
BMM application: the dense matter equation of state

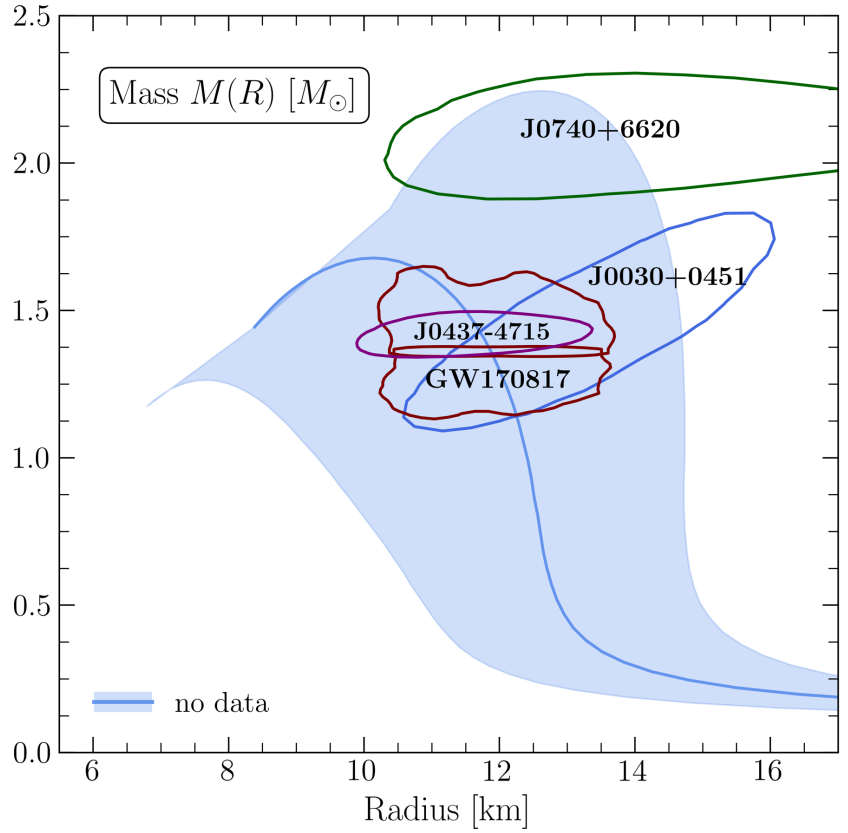
Smoothly transition between two stationary RBF kernels using a switching function

$$\kappa_f(x, x'; \ell_1, \ell_2, \bar{c}_1^2, \bar{c}_2^2, \boldsymbol{\xi}) = [1 - \alpha(x, \boldsymbol{\xi})] \times \kappa_1(x, x'; \bar{c}_1^2, \ell_1)[1 - \alpha(x', \boldsymbol{\xi})] + \alpha(x, \boldsymbol{\xi})\kappa_2(x, x'; \bar{c}_2^2, \ell_2)\alpha(x', \boldsymbol{\xi})$$

$$\alpha_{\text{sigmoid}}(x; x_0, w) = \left[1 + \exp\left(-\frac{(x - x_0)}{w}\right)\right]^{-1}$$

Use a "greedy approach" to sequentially add heavy-ion data and future model predictions to the EOS





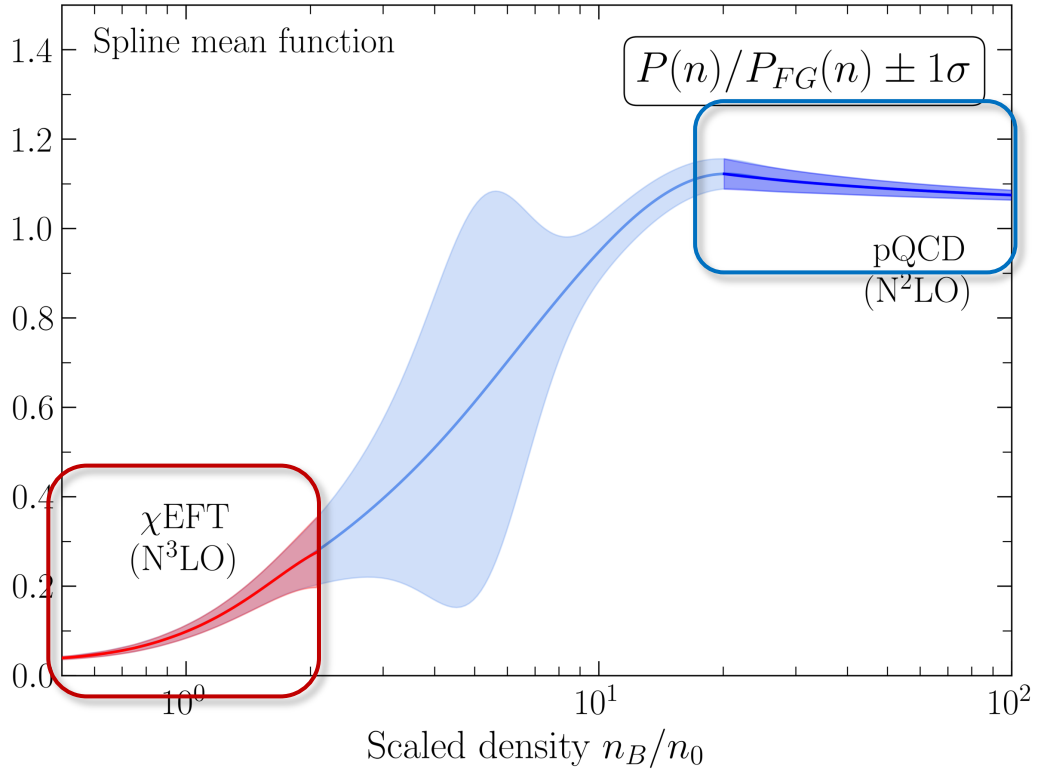
BMM application: the dense matter equation of state

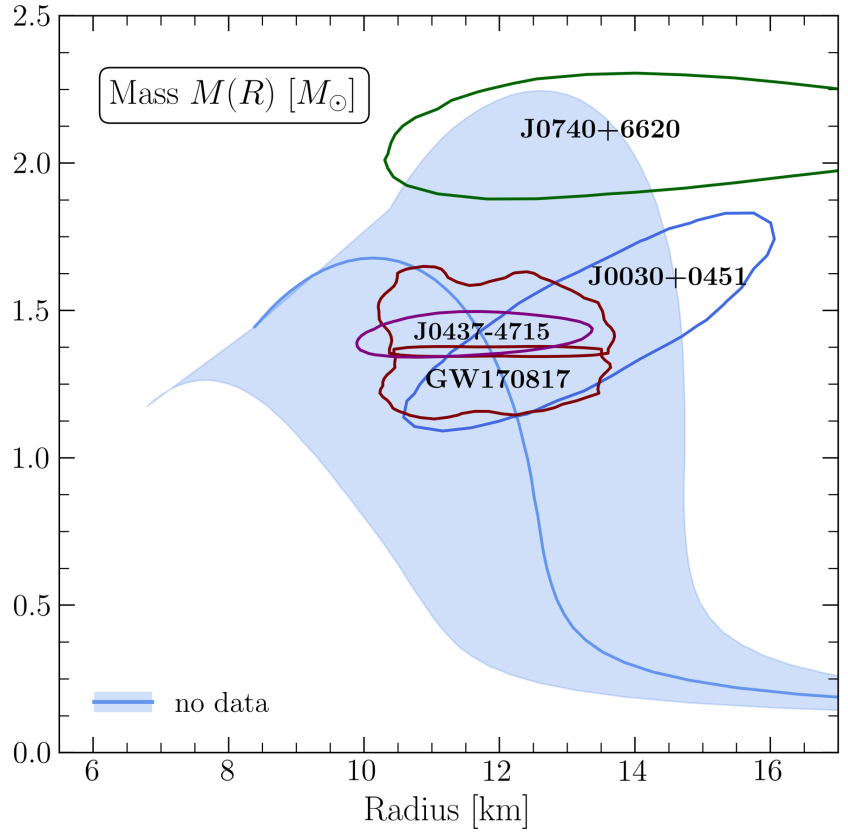
Smoothly transition between two stationary RBF kernels using a switching function

$$\alpha_{\text{sigmoid}}(x; x_0, w) = \left[1 + \exp\left(-\frac{(x - x_0)}{w}\right)\right]^{-1}$$

$$\begin{aligned}\kappa_f(x, x'; \ell_1, \ell_2, \bar{c}_1^2, \bar{c}_2^2, \boldsymbol{\xi}) = [1 - \alpha(x, \boldsymbol{\xi})] \\ \times \kappa_1(x, x'; \bar{c}_1^2, \ell_1)[1 - \alpha(x', \boldsymbol{\xi})] \\ + \alpha(x, \boldsymbol{\xi})\kappa_2(x, x'; \bar{c}_2^2, \ell_2)\alpha(x', \boldsymbol{\xi})\end{aligned}$$

Use a "greedy approach" to sequentially add heavy-ion data and future model predictions to the EOS





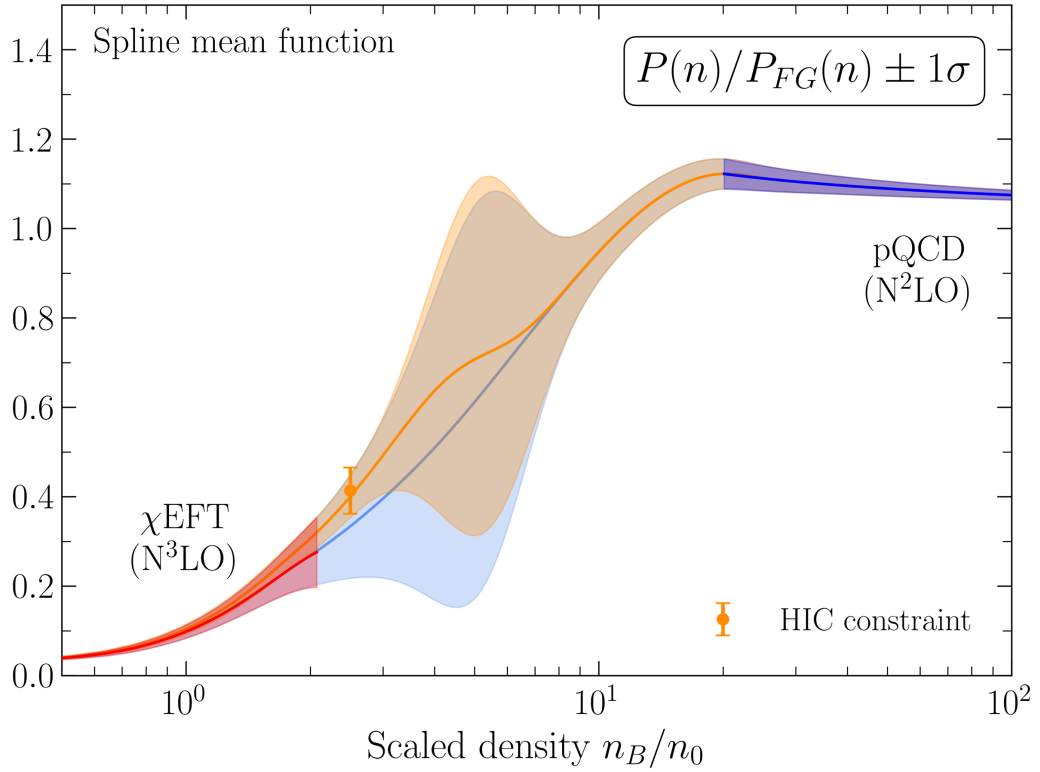
BMM application: the dense matter equation of state

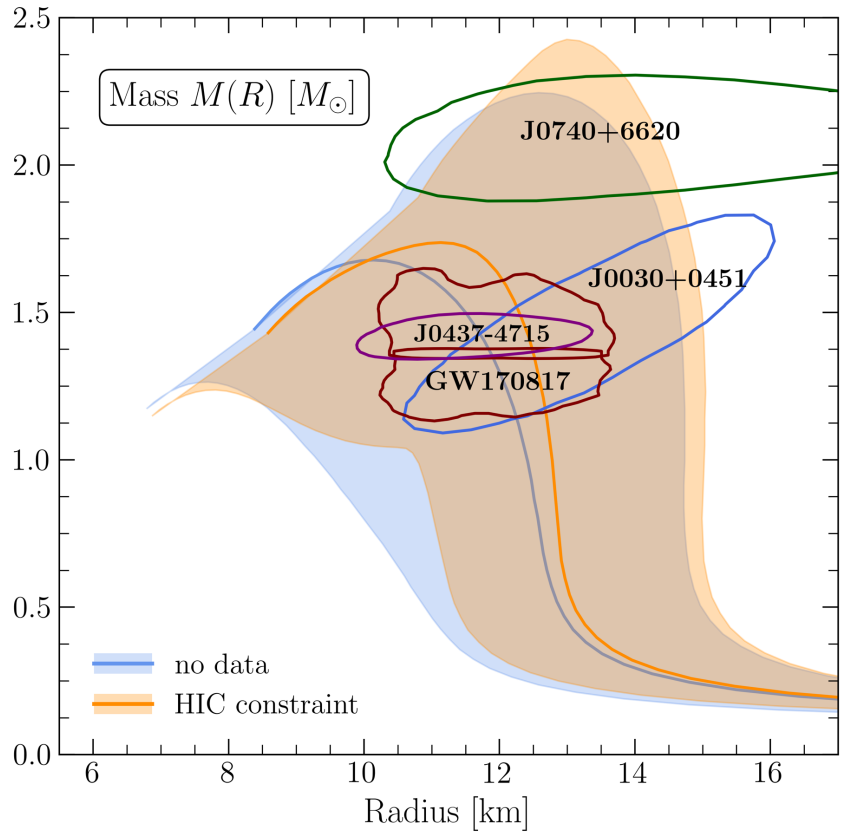
Smoothly transition between two stationary RBF kernels using a switching function

$$\kappa_f(x,x';\ell_1,\ell_2,\bar{c}_1^2,\bar{c}_2^2,\boldsymbol{\xi}) = [1-\alpha(x,\boldsymbol{\xi})] \times \kappa_1(x,x';\bar{c}_1^2,\ell_1)[1-\alpha(x',\boldsymbol{\xi})] + \alpha(x,\boldsymbol{\xi})\kappa_2(x,x';\bar{c}_2^2,\ell_2)\alpha(x',\boldsymbol{\xi})$$

$$\alpha_{\text{sigmoid}}(x;x_0,w) = \left[1+\exp\left(-\frac{(x-x_0)}{w}\right)\right]^{-1}$$

Use a "greedy approach" to sequentially add heavy-ion data and future model predictions to the EOS





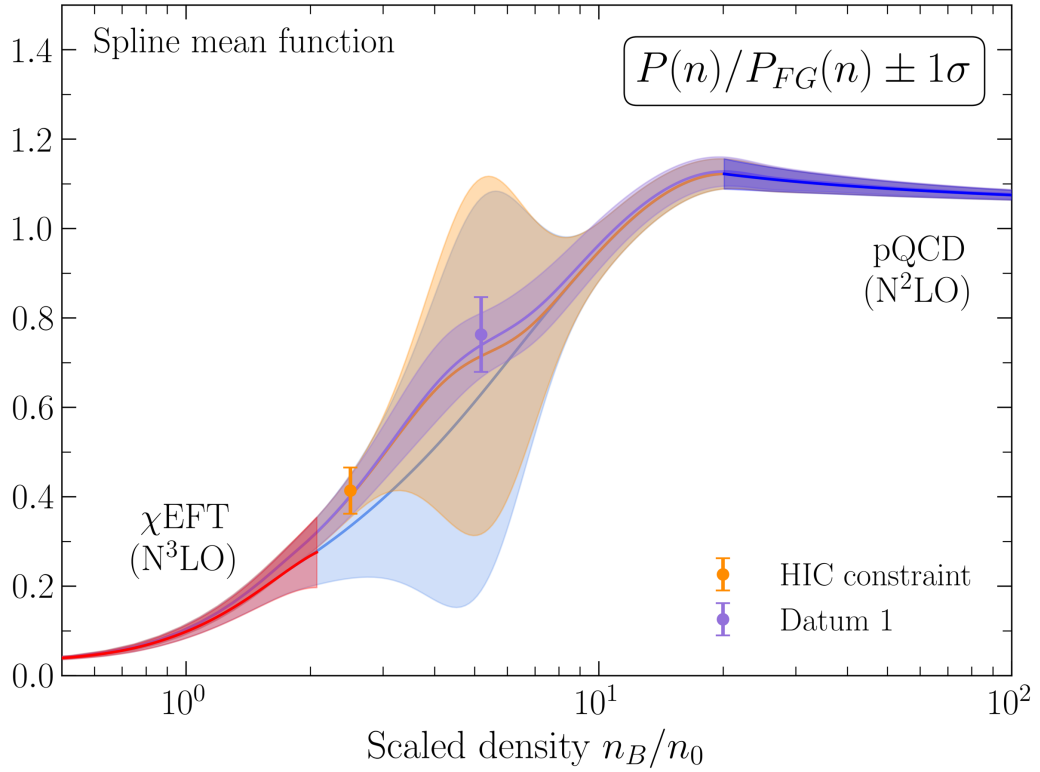
BMM application: the dense matter equation of state

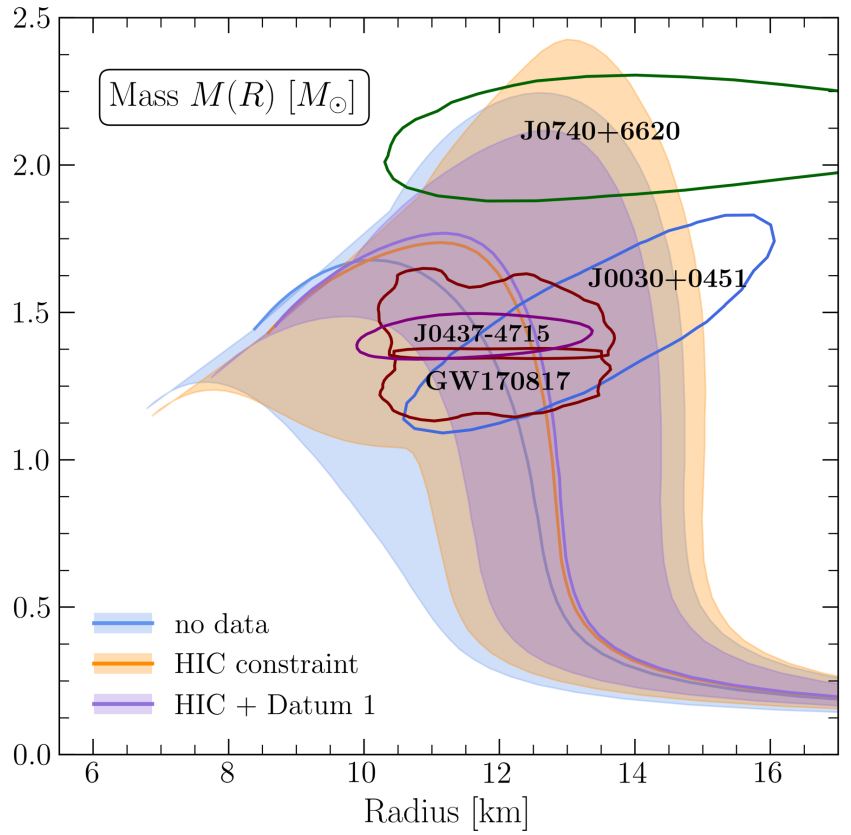
Smoothly transition between two stationary RBF kernels using a switching function

$$\kappa_f(x,x';\ell_1,\ell_2,\bar{c}_1^2,\bar{c}_2^2,\boldsymbol{\xi}) = [1-\alpha(x,\boldsymbol{\xi})] \times \kappa_1(x,x';\bar{c}_1^2,\ell_1)[1-\alpha(x',\boldsymbol{\xi})] + \alpha(x,\boldsymbol{\xi})\kappa_2(x,x';\bar{c}_2^2,\ell_2)\alpha(x',\boldsymbol{\xi})$$

$$\alpha_{\text{sigmoid}}(x;x_0,w) = \left[1+\exp\left(-\frac{(x-x_0)}{w}\right)\right]^{-1}$$

Use a "greedy approach" to sequentially add heavy-ion data and future model predictions to the EOS





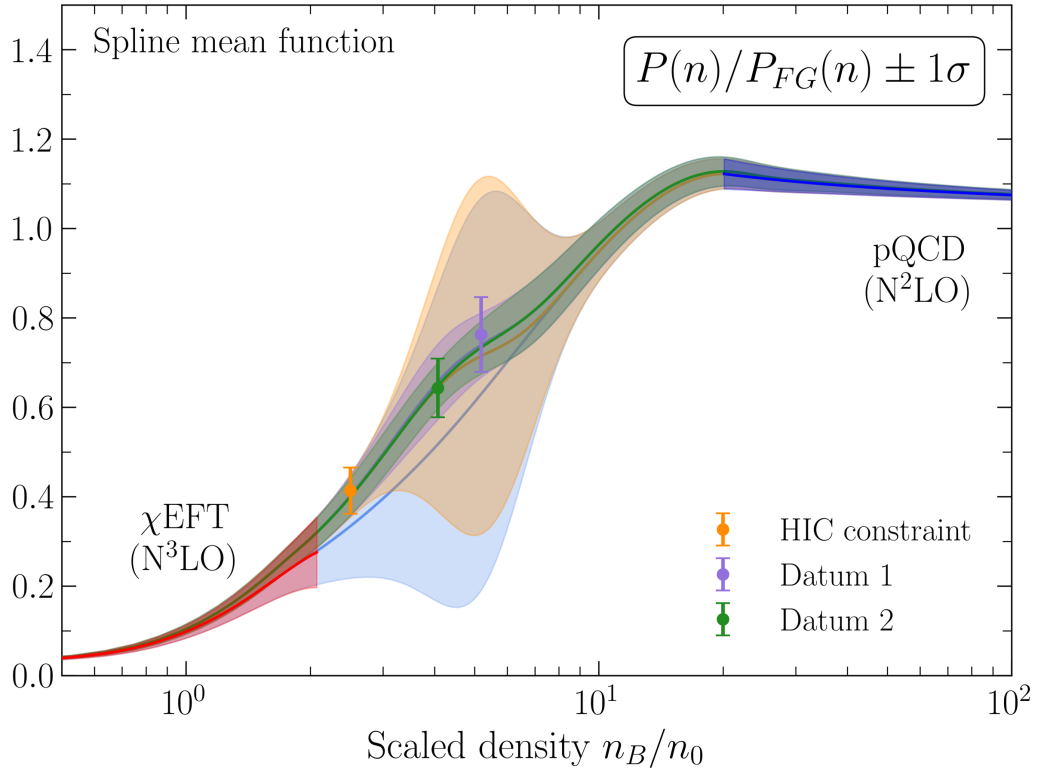
BMM application: the dense matter equation of state

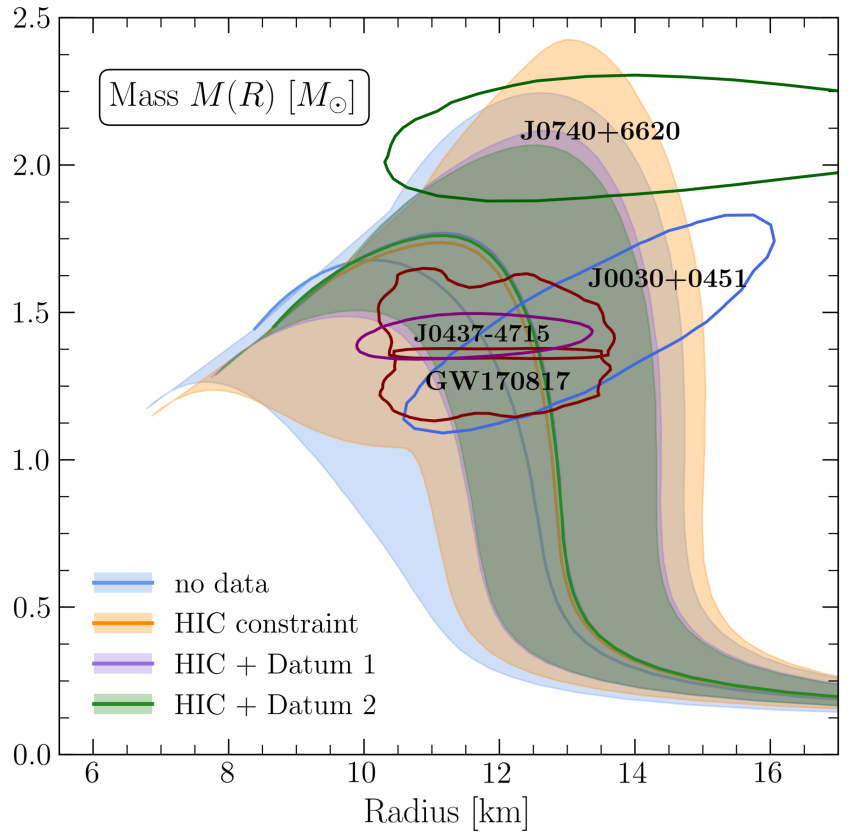
Smoothly transition between two stationary RBF kernels using a switching function

$$\alpha_{\text{sigmoid}}(x; x_0, w) = \left[1 + \exp\left(-\frac{(x - x_0)}{w}\right)\right]^{-1}$$

$$\begin{aligned}\kappa_f(x, x'; \ell_1, \ell_2, \bar{c}_1^2, \bar{c}_2^2, \boldsymbol{\xi}) &= [1 - \alpha(x, \boldsymbol{\xi})] \\ &\times \kappa_1(x, x'; \bar{c}_1^2, \ell_1)[1 - \alpha(x', \boldsymbol{\xi})] \\ &+ \alpha(x, \boldsymbol{\xi})\kappa_2(x, x'; \bar{c}_2^2, \ell_2)\alpha(x', \boldsymbol{\xi})\end{aligned}$$

Use a "greedy approach" to sequentially add heavy-ion data and future model predictions to the EOS





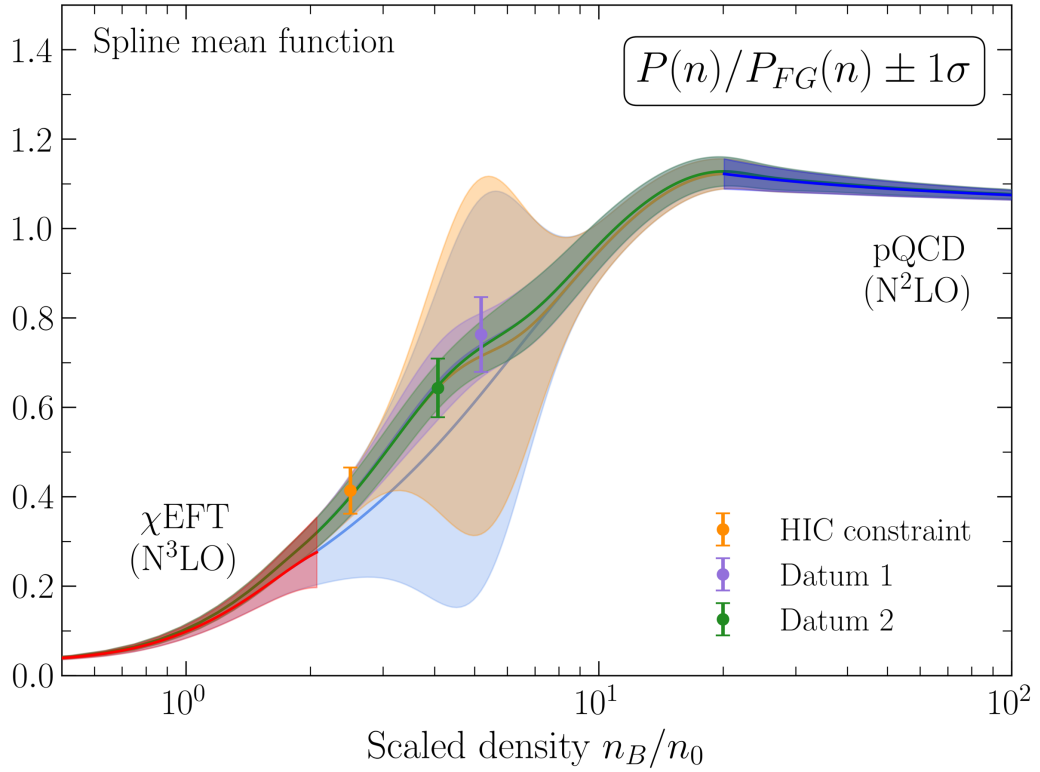
BMM application: the dense matter equation of state

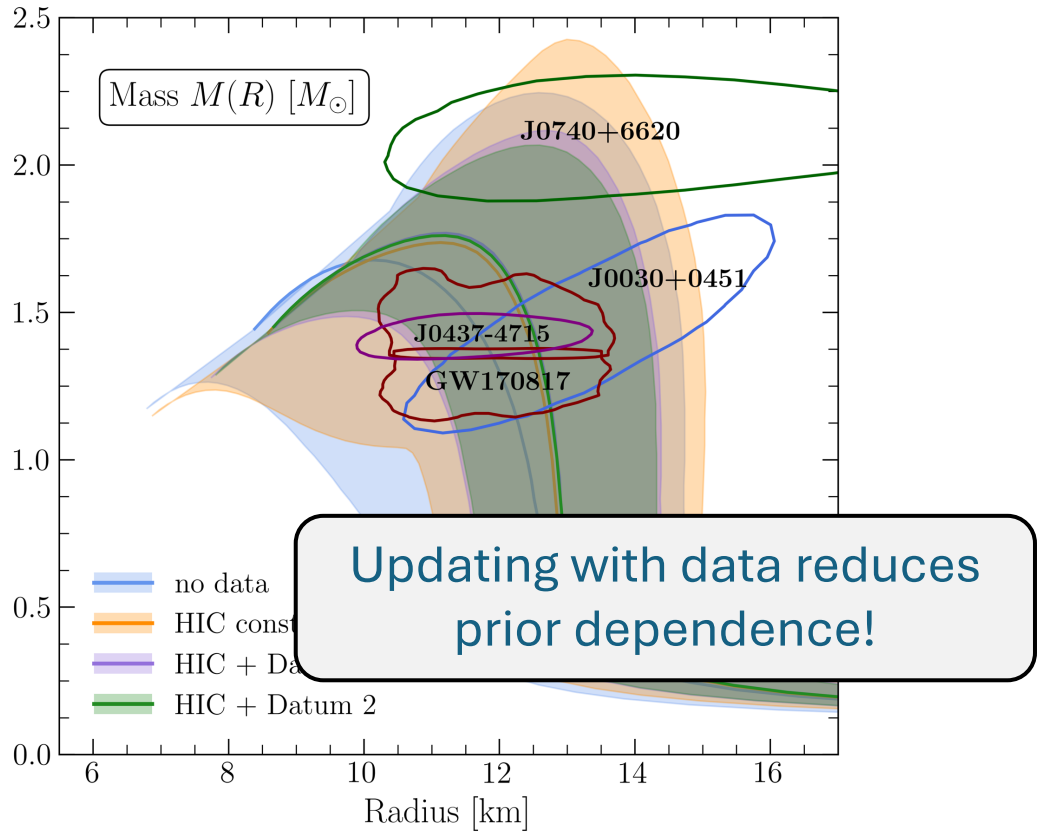
Smoothly transition between two stationary RBF kernels using a switching function

$$\kappa_f(x, x'; \ell_1, \ell_2, \bar{c}_1^2, \bar{c}_2^2, \boldsymbol{\xi}) = [1 - \alpha(x, \boldsymbol{\xi})] \times \kappa_1(x, x'; \bar{c}_1^2, \ell_1)[1 - \alpha(x', \boldsymbol{\xi})] + \alpha(x, \boldsymbol{\xi})\kappa_2(x, x'; \bar{c}_2^2, \ell_2)\alpha(x', \boldsymbol{\xi})$$

$$\alpha_{\text{sigmoid}}(x; x_0, w) = \left[1 + \exp\left(-\frac{(x - x_0)}{w}\right)\right]^{-1}$$

Use a "greedy approach" to sequentially add heavy-ion data and future model predictions to the EOS

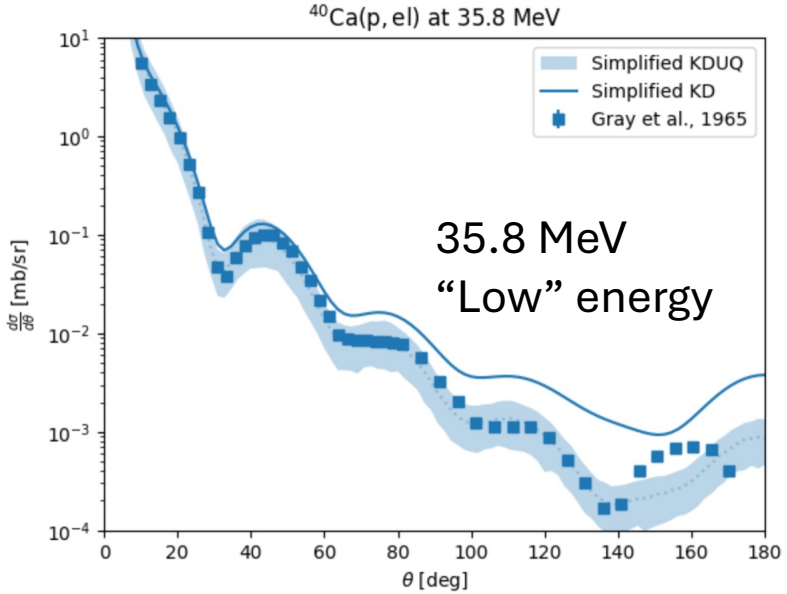


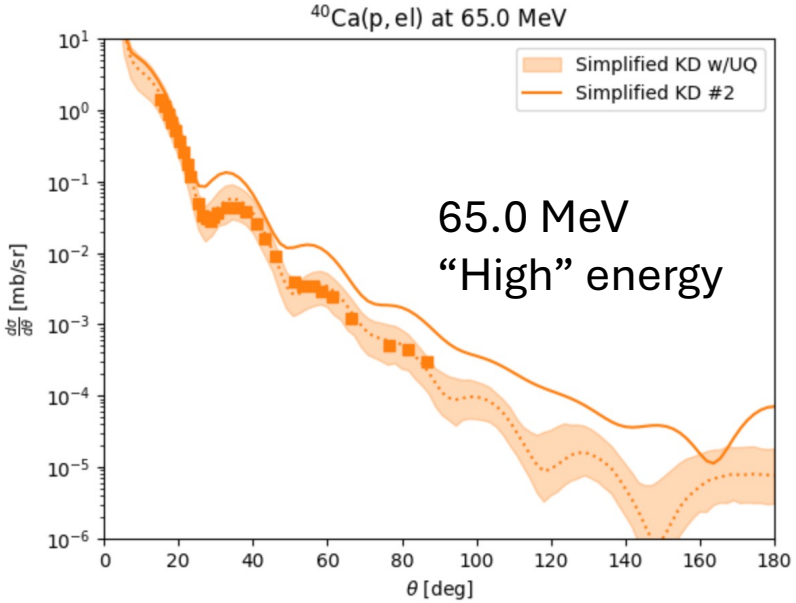


Updating with data reduces prior dependence!

A. C. Semposki et al. PRC **113** 015808 (2026)

Ongoing work: BMM for optical potentials





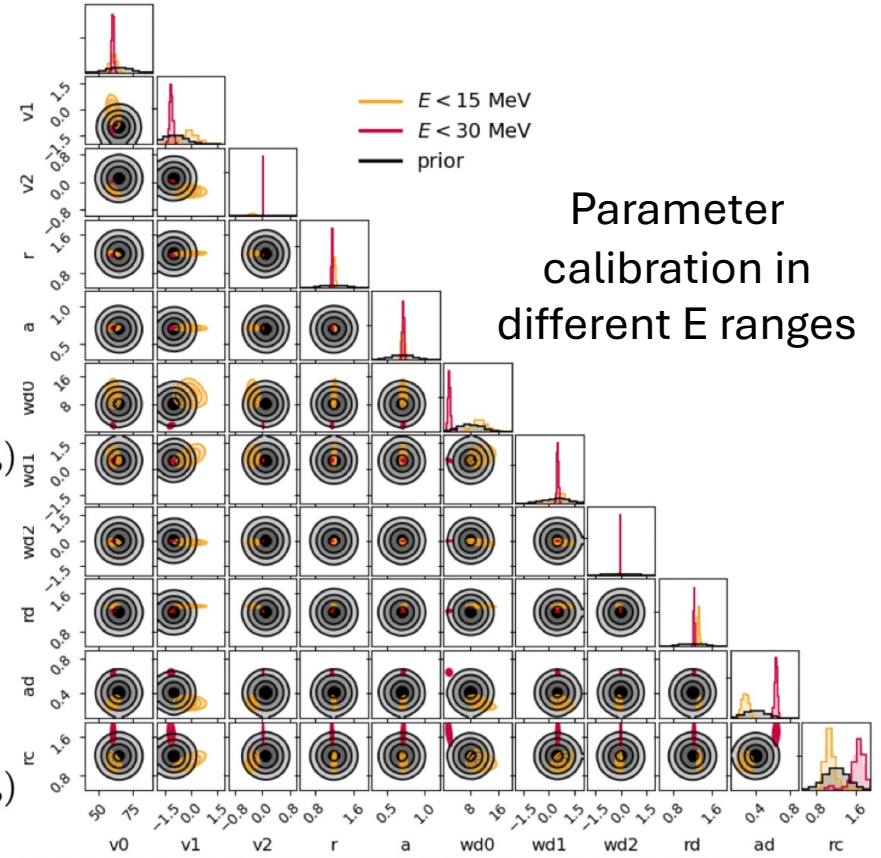
Simplified KDUQ for low and high energy regions

Removed imaginary volume term:

$$U(r, E; \boldsymbol{\theta}, \ell, j) = -V_v(E) f_{WS}(r, R_v, a_v) - i4W_s(E)\frac{d}{dr} f_{WS}(r, R_w, a_w) + 2\ell \cdot sV_{so}\left(\frac{\hbar}{m_\pi c}\right)^2 \frac{1}{r}\frac{d}{dr} f_{WS}(R_{so}, a_{so}) + V_C(r)$$

Removed imaginary surface term:

$$U(r, E; \boldsymbol{\theta}, \ell, j) = -V_v(E) f_{WS}(r, R_v, a_v) - iW_v(E) f_{WS}(r, R_w, a_w) + 2\ell \cdot sV_{so}\left(\frac{\hbar}{m_\pi c}\right)^2 \frac{1}{r}\frac{d}{dr} f_{WS}(R_{so}, a_{so}) + V_C(r)$$



Parameter calibration in different E ranges

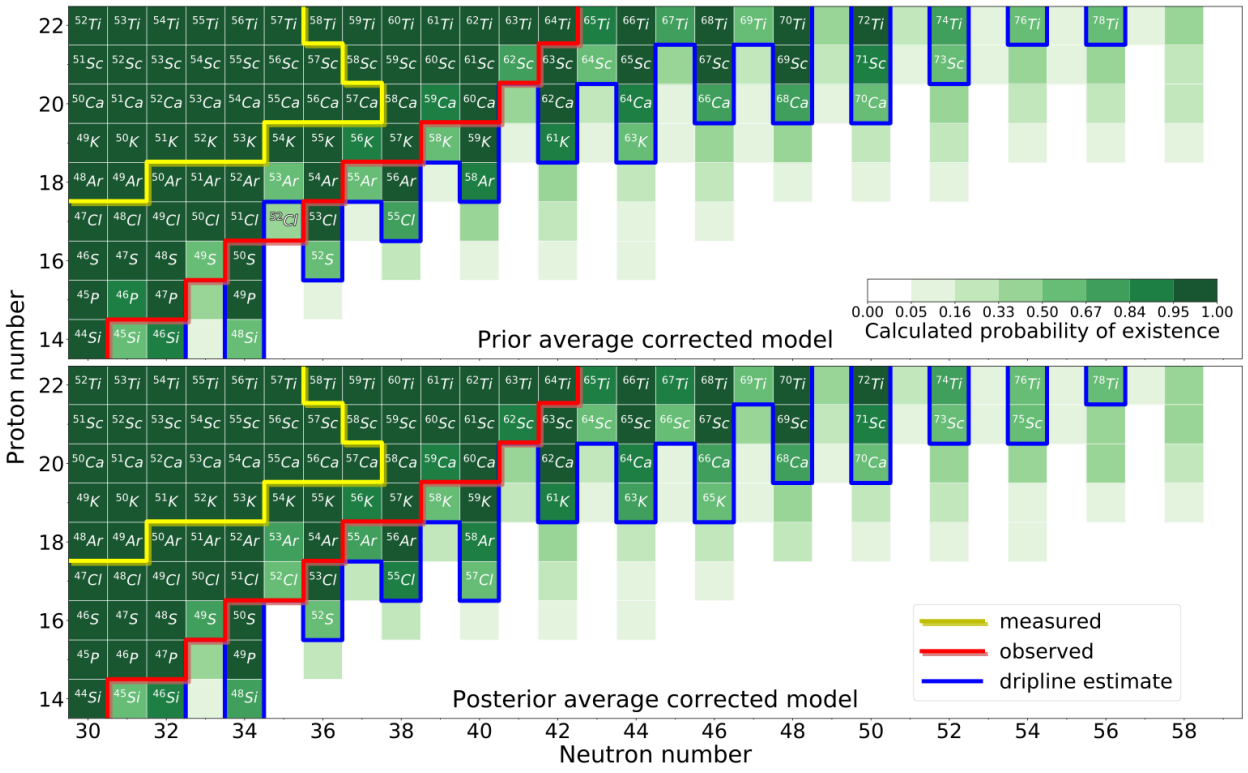
Long-term goals: use model mixing framework to:

- Extrapolate along isotope chains
- Include *ab initio* optical potentials **(see Kyle's talk!)**

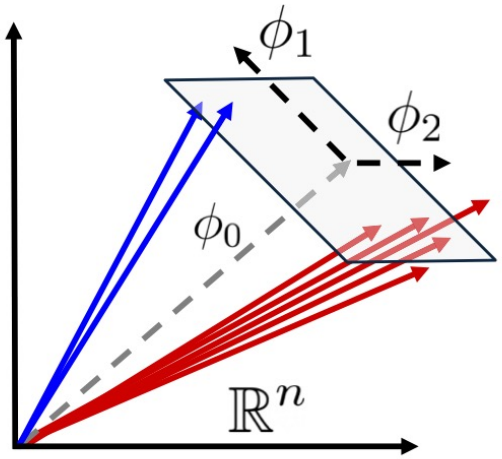
Other studies: Bayesian model combination

Bayesian Model Averaging (BMA) globally weights models to use each in an overall, collective model (see dripline predictions below!)

Principal Component Analysis (PCA) eliminates **redundant information** in models and construct a better mixed model posterior => possible uses with NLDs

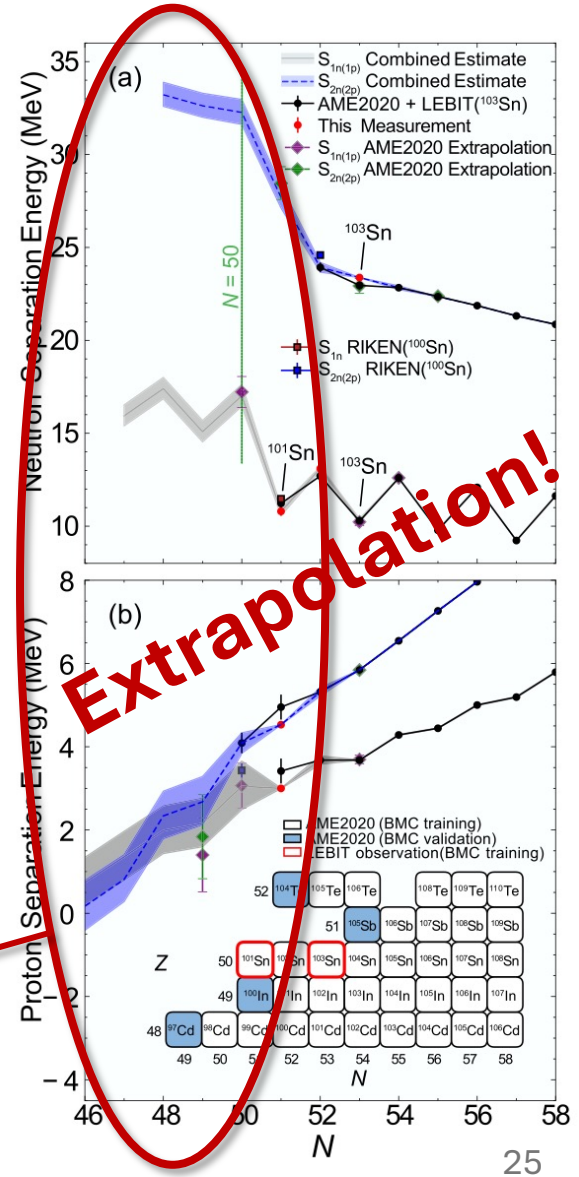


L. Neufcourt, Y. Cao, W. Nazarewicz, E. Olsen, F. Viens, Phys. Rev. Lett. **122**, 062502 (2019)



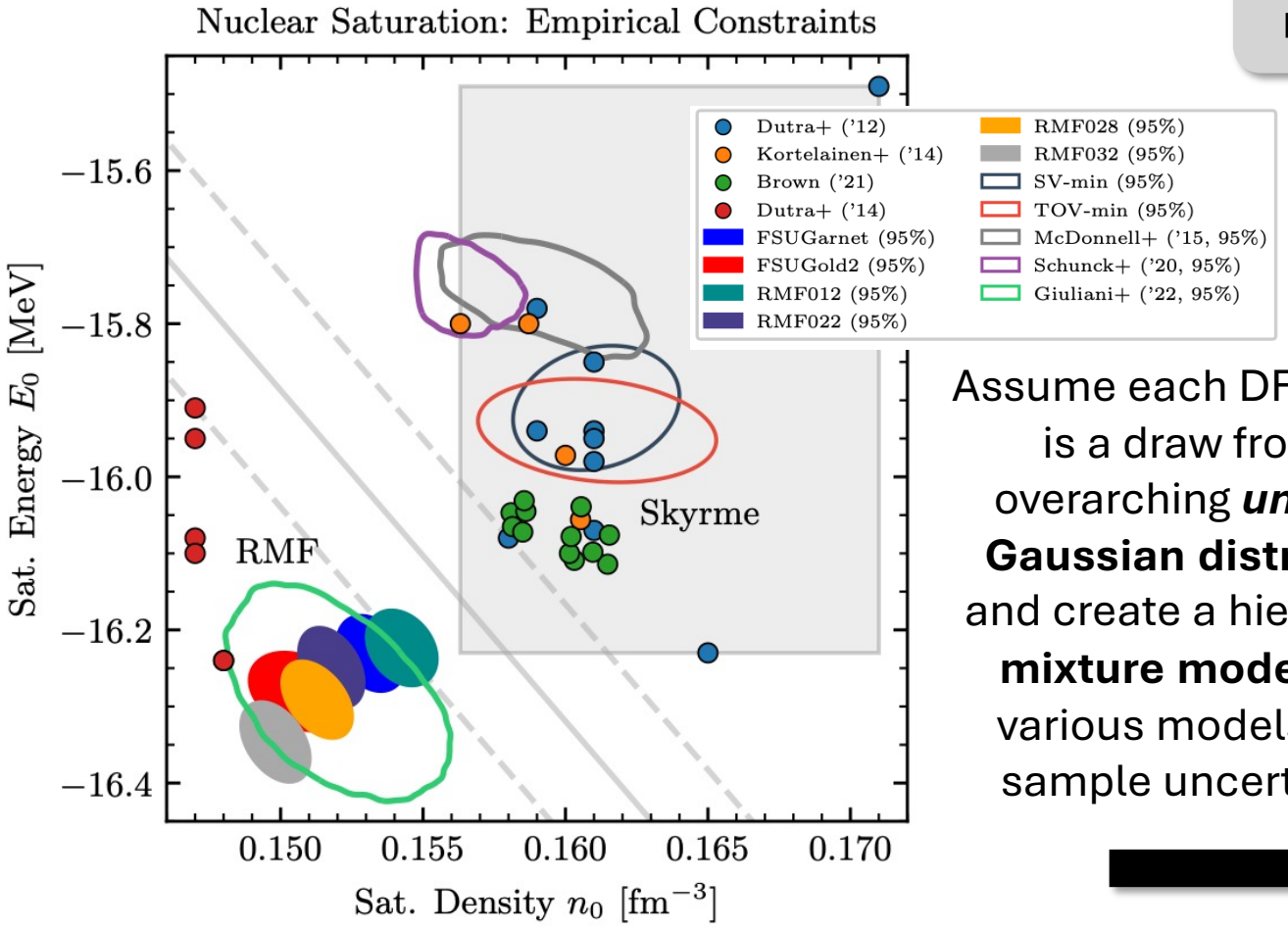
P. G. Giuliani, K. Godbey, V. Kejzlar, W. Nazarewicz, Phys. Rev. Research **6**, 033266 (2024)

See Sudhanva Lalit's talk on the tin isotope chain!



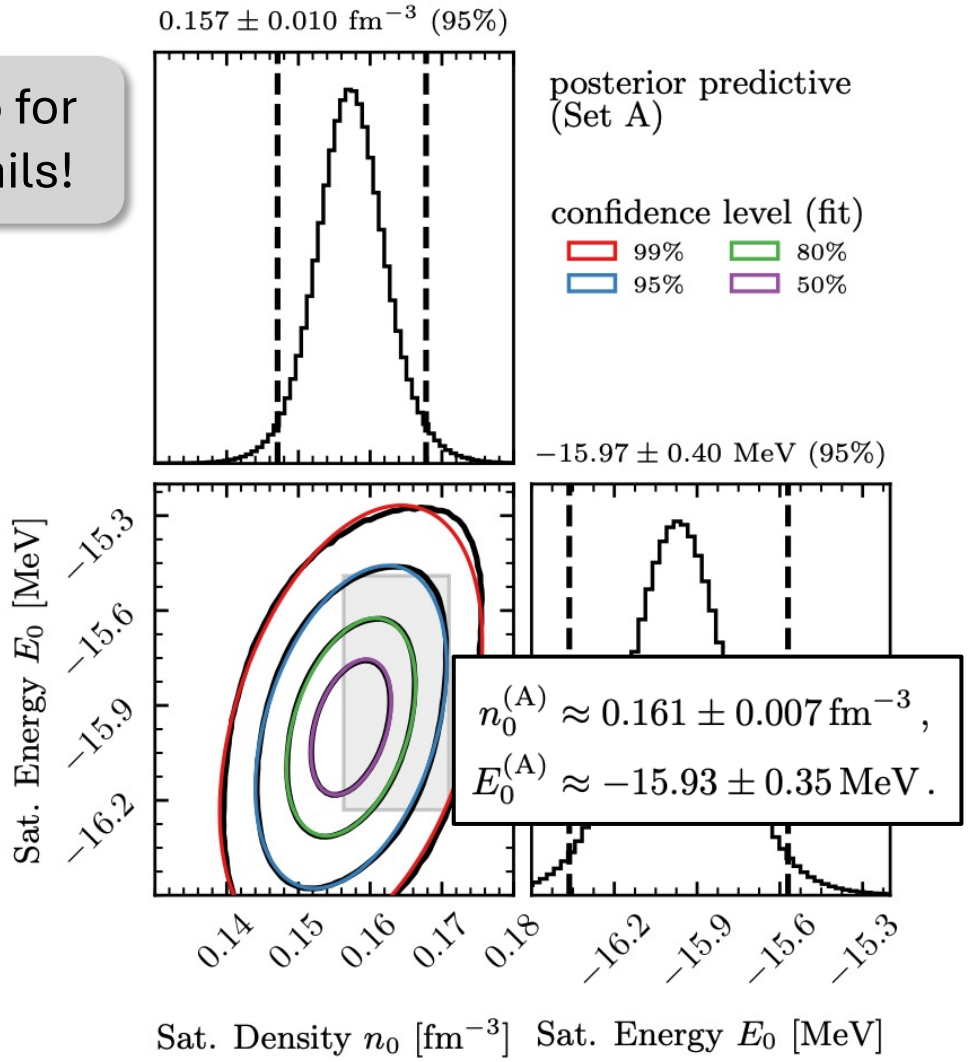
Other studies: nuclear saturation point

RMF and **Skyrme** models predict two different classes of (E_0, n_0) => how to **reconcile** this?



Ask **Pablo** for more details!

Assume each DFT sample is a draw from an overarching ***unknown*** **Gaussian distribution** and create a hierarchical **mixture model** using various models of the sample uncertainties



Paper: C. Drischler, P. Giuliani, S. Bezoui, J. Piekarewicz, F. Viens, PRC **110** 044320 (2024)

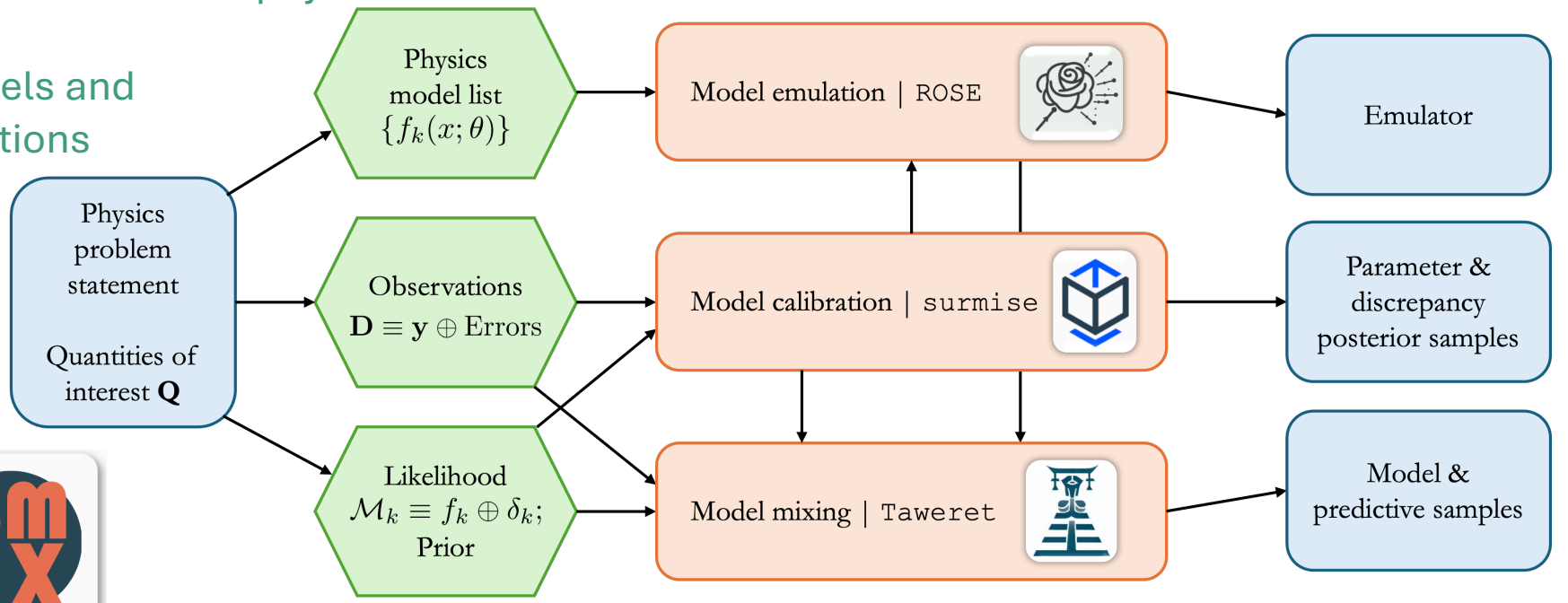
Bayesian resource: the BAND framework

Bayesian Analysis of Nuclear Dynamics

Cyber-infrastructure framework

Developed tools for the nuclear physics community
Tested on toy models and real-world applications

https://github.com/bandframework/bandframework



Our model mixing software!

ParMOO

Taweret

BMEX

surmise

jitR

ROSE

+ BAND applications database

+ examples shown in this talk

PUQ

pybmc

LCGP

+ more!

BMM resource: Taweret Python package

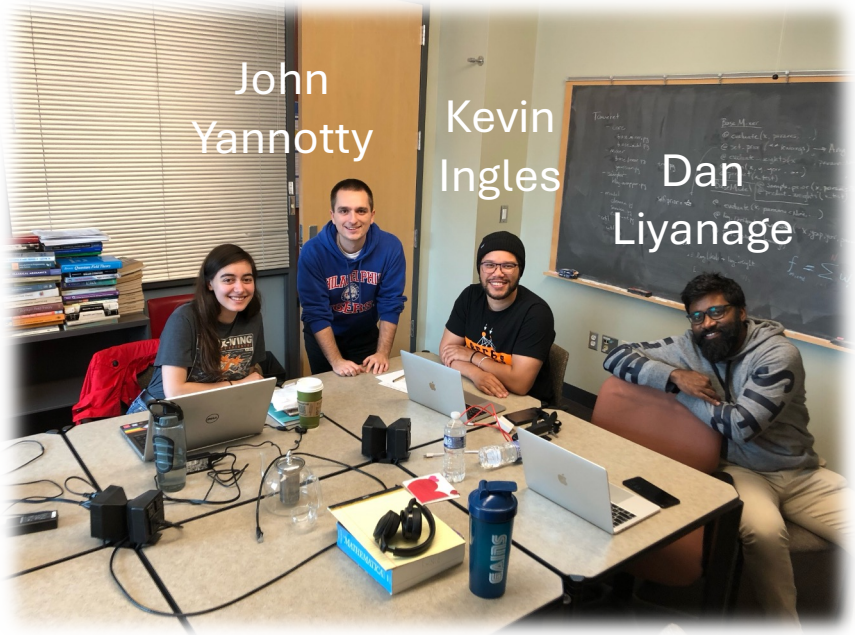
Python interface for users to employ BMM methods on **their own models**

Present: our 3 methods + toy models

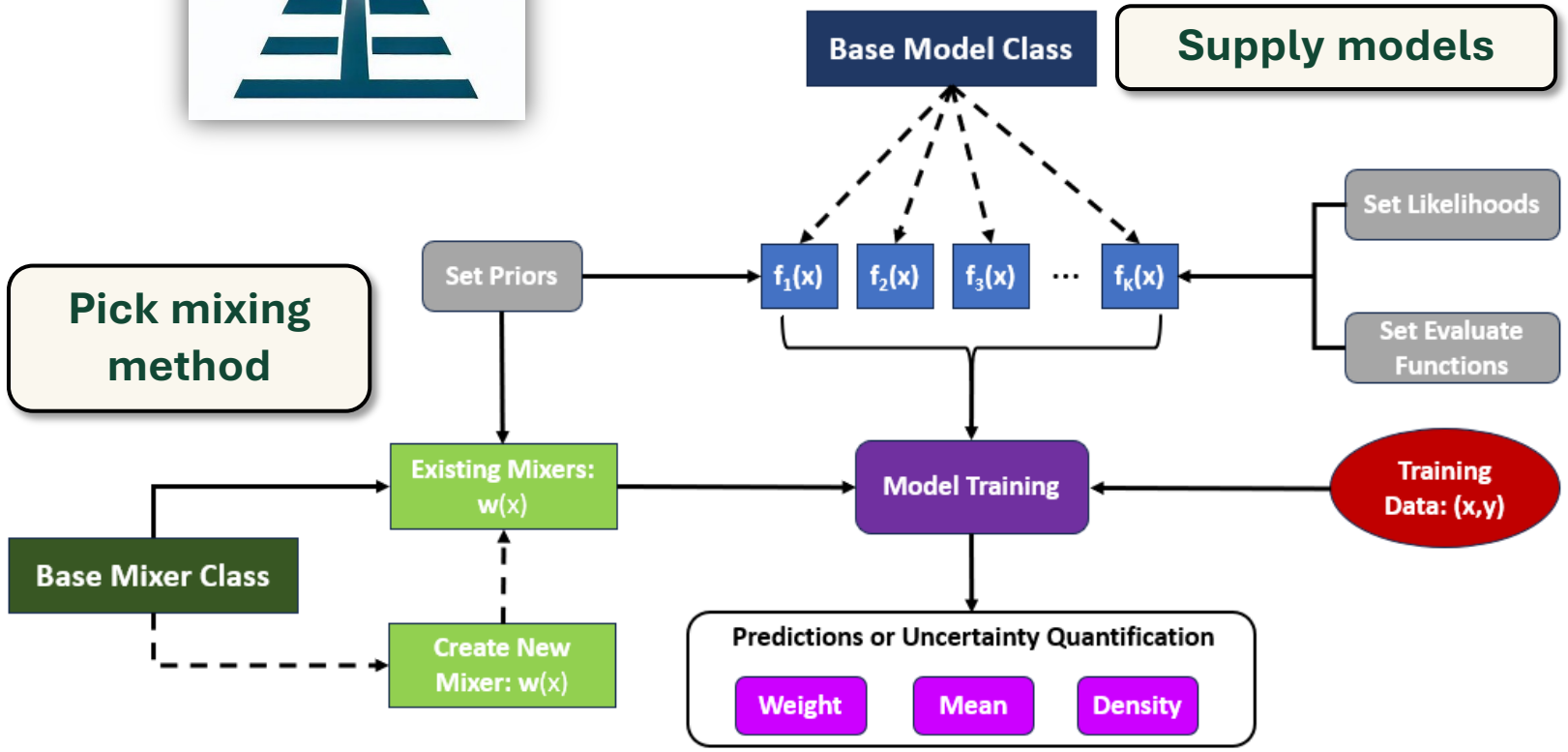
Advantage: *add your own BMM method!*

Open-source repository:
https://github.com/bandframework/Taweret

Published in JOSS The Journal of Open Source Software



Picture credit: Dick Furnstahl



DOI: 10.21105/joss.06175

New GP model mixing feature coming with BAND release in summer!

Summary

Several ways to **quantify errors in models** that may be applicable in statistical reactions

- *UQ via discrepancy modelling*
- *UQ through model calibration (as many are already doing)*

BMM still in its toddler phase, but three powerful techniques **available now**—and improvements are ongoing!

Work in **BAND collaboration** useful for this community, especially model calibration and mixing packages





Thank you!



Daniel Phillips (OU)



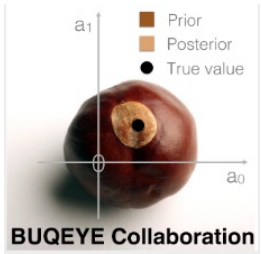
Dick Furnstahl (OSU)



Sudhanva Lalit (FRIB)



Divya Singh (UC Berkeley)





Christian Drischler (OU)



Jordan Melendez



Joshua Maldonado (OU)



Mia Kumamoto (UC Berkeley/OU)



Sanika S. Khadkikar (Penn State)

ACS was supported by: US DOE, contract DE-FG02-93ER-40756, NSF CSSI program, award number OAC-2004601

Backup slides

Multivariate model mixing

ACS, RJ Furnstahl, DR Phillips, PRC (2022)

Toy model to mimic EFTs

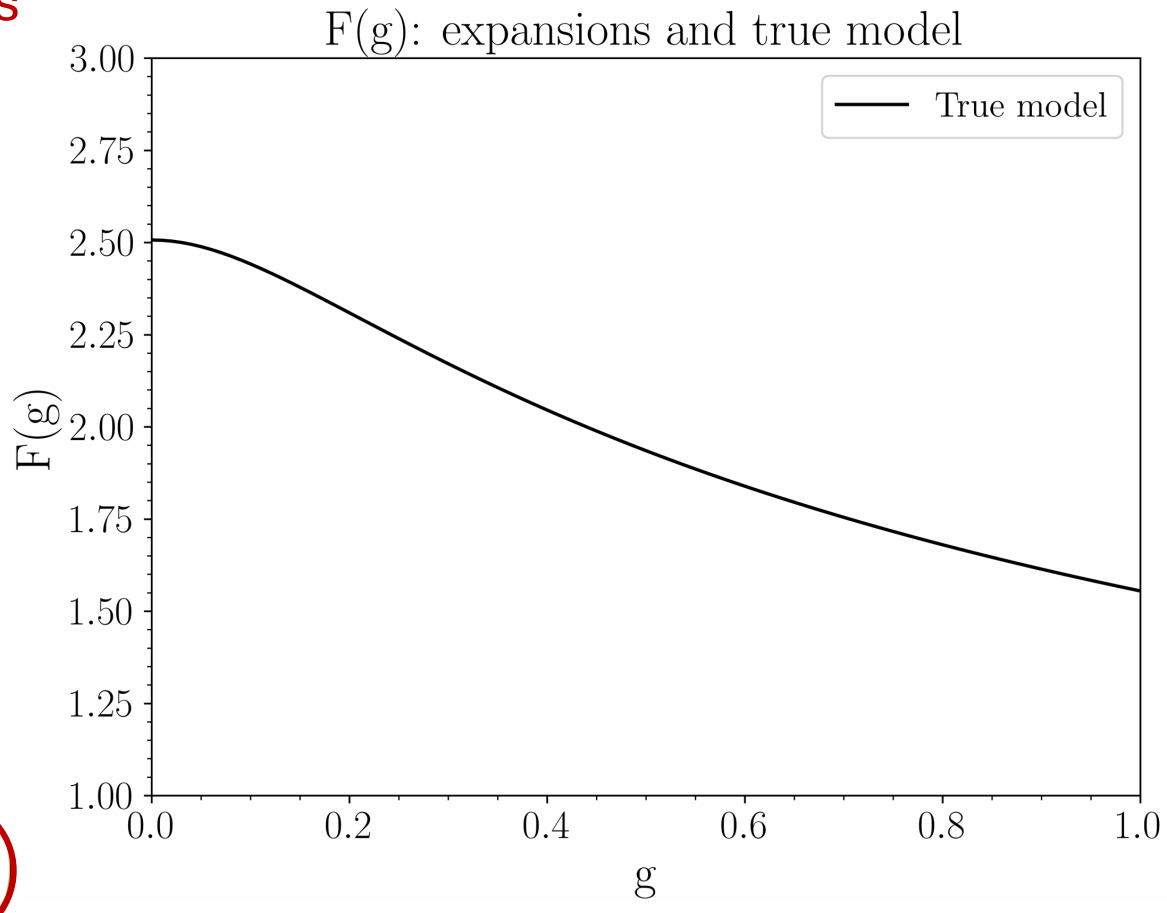
$$F(g) = \int_{-\infty}^{\infty} dx\ e^{-\frac{x^2}{2} - g^2 x^4}$$

$f_s(g) \simeq \sum_{k=0}^{N_s} s_k g^k$ Asymptotic

$f_l(g) \simeq \frac{1}{\sqrt{g}} \sum_{k=0}^{N_l} l_k g^{-k}$ Convergent

Use a Gaussian form for the calibrated models

$$\mathcal{M}_{N_s,l} = \mathcal{N}\left(f_{s,l}(g), \sigma^2_{N_s,l}\right)$$



Uncertainty Quantification

Use estimation of the uncertainties from higher order terms

$$\sigma_s^{2(N_s)} = \bar{s}^2 \left(\frac{N_s}{2}\right)!^2 (4g)^{2N_s+4}$$

$$\sigma_s^{2(N_s)} = \bar{s}^2 \left(\frac{N_s-1}{2}\right)!^2 (4g)^{2N_s+2}$$

$$\sigma_l^{2(N_l)} = \bar{l}^2 \frac{1}{\Gamma\left(\frac{N_l+3}{2}\right)^2} \frac{1}{(4g)^{2N_l+3}}$$

Using orders we know to approximate those we don't...

Multivariate model mixing

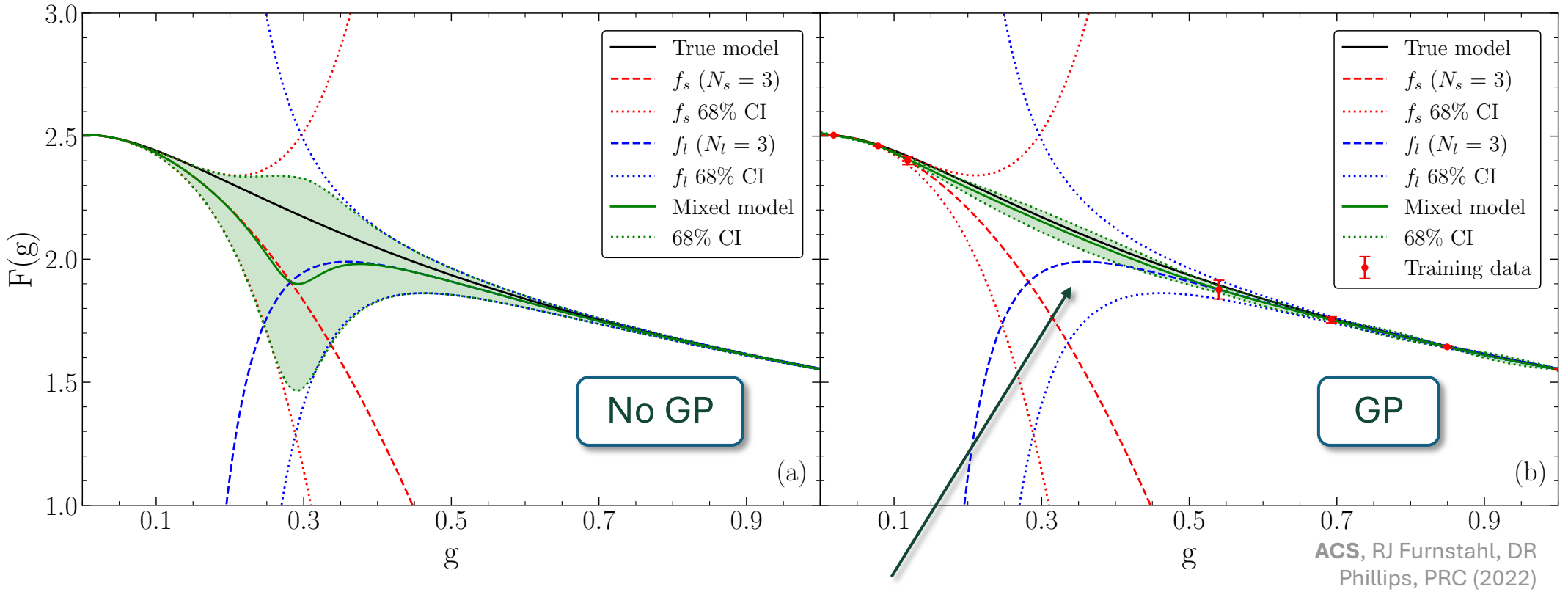
Combine models...

$$\mathcal{N}(f_s, \sigma_s^2)\ \mathcal{N}(f_l, \sigma_l^2) \ldots$$

$$f_\dagger \propto \mathcal{N}\left(\frac{\sigma_s^2 f_l + \sigma_l^2 f_s}{\sigma_s^2 + \sigma_l^2}, \frac{\sigma_s^2 \sigma_l^2}{\sigma_s^2 + \sigma_l^2}\right)$$

$$f_\dagger \sim \mathcal{N}(Z_P^{-1} \sum_k \frac{1}{\sigma_k^2} f_k, Z_P^{-1}) \qquad Z_P \equiv \sum_{k=1}^{K} \frac{1}{\sigma_k^2}$$

Weights considered the inverse of the variances for each model



GP does a good job of predicting in the unknown region

ACS, RJ Furnstahl, DR Phillips, PRC (2022)

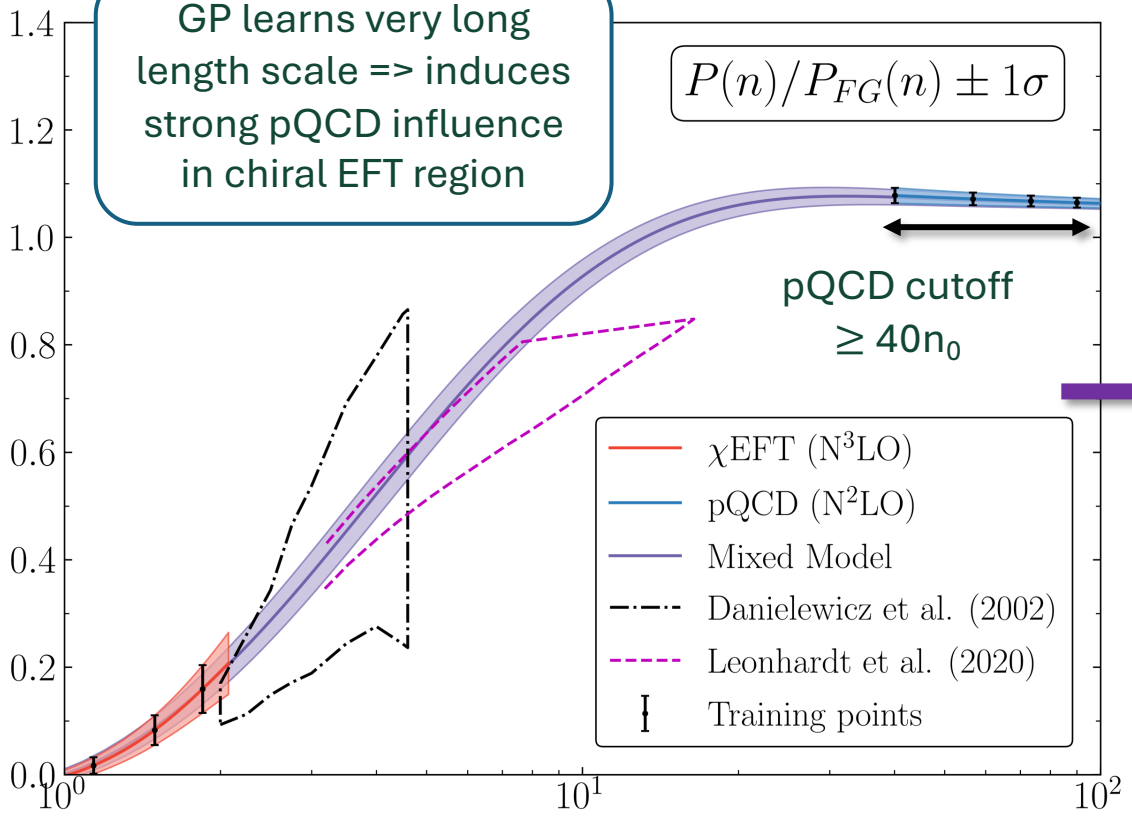
Promising for real applications!

Multivariate model mixing

$$k_{\mathrm{RBF}}(x, x'; \ell) = \exp\left(\frac{-(x - x')^2}{2\ell^2}\right)$$

Unconstrained prior

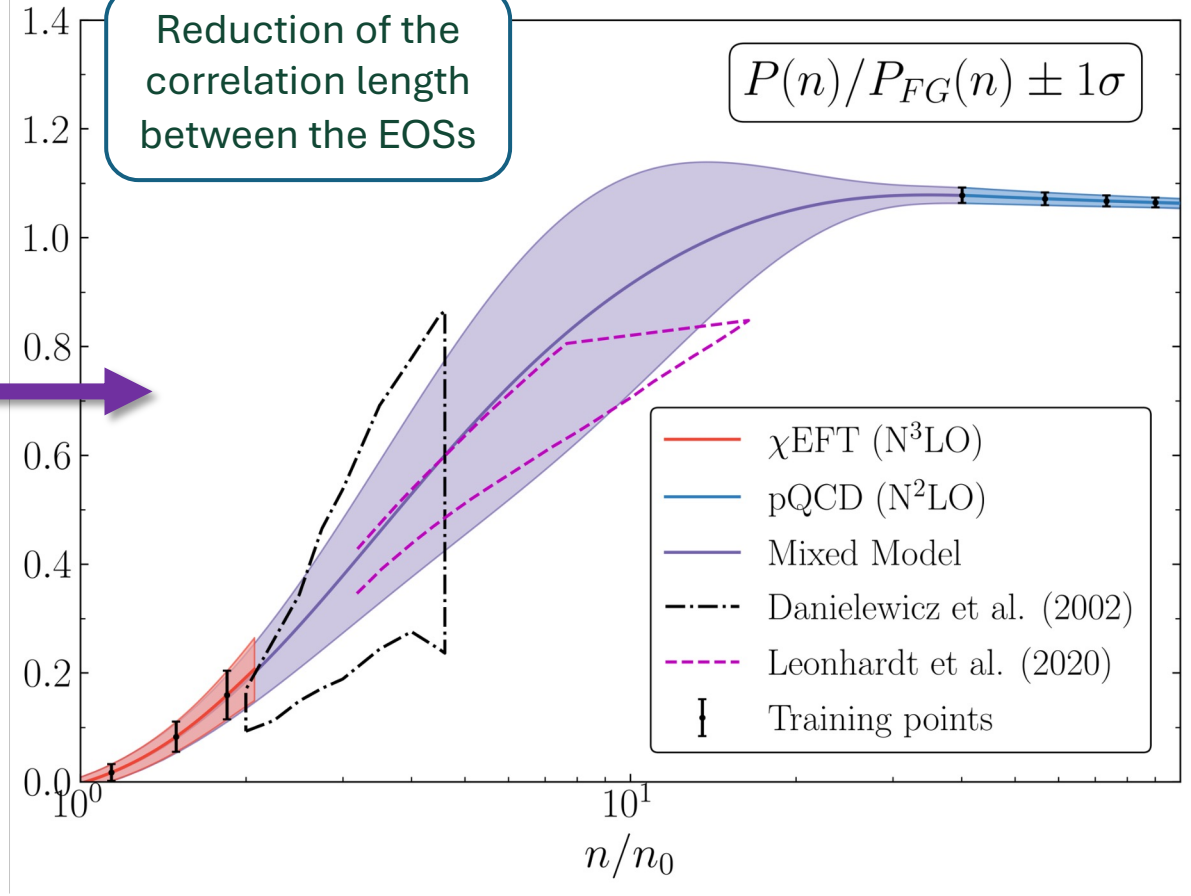
GP learns very long length scale => induces strong pQCD influence in chiral EFT region



A. C. Semposki et al. PRC **111** 035804 (2025)

Constrained prior on lengthscale

Reduction of the correlation length between the EOSs

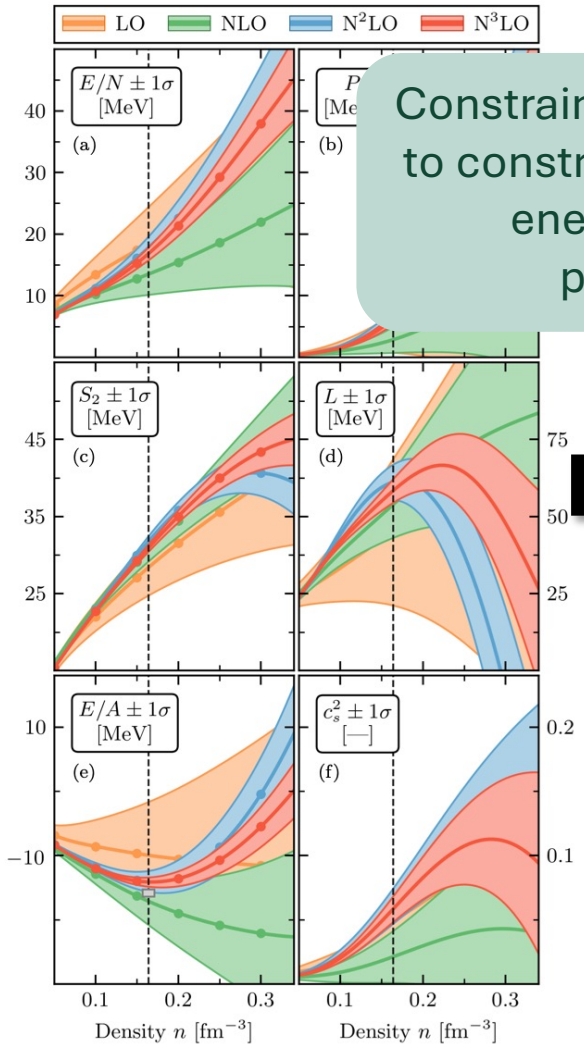


Physics-informed priors on the mixed model GP have a very large influence on the uncertainties obtained

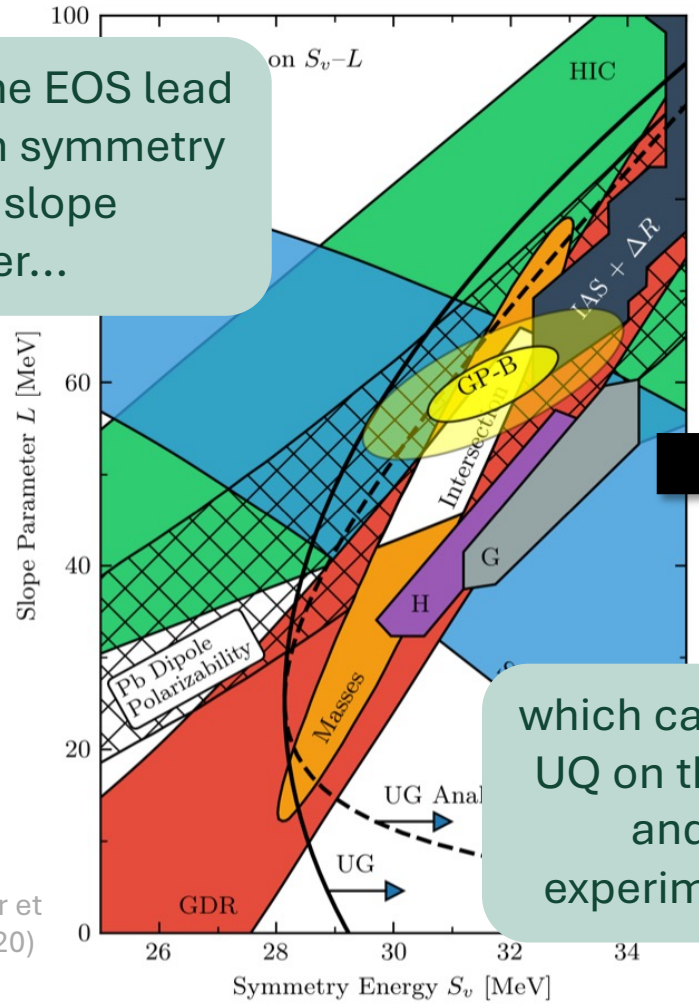
BUQEYE + the nuclear matter EOS

Recent progress using GPs has led to **tight theoretical constraints** on the EOS

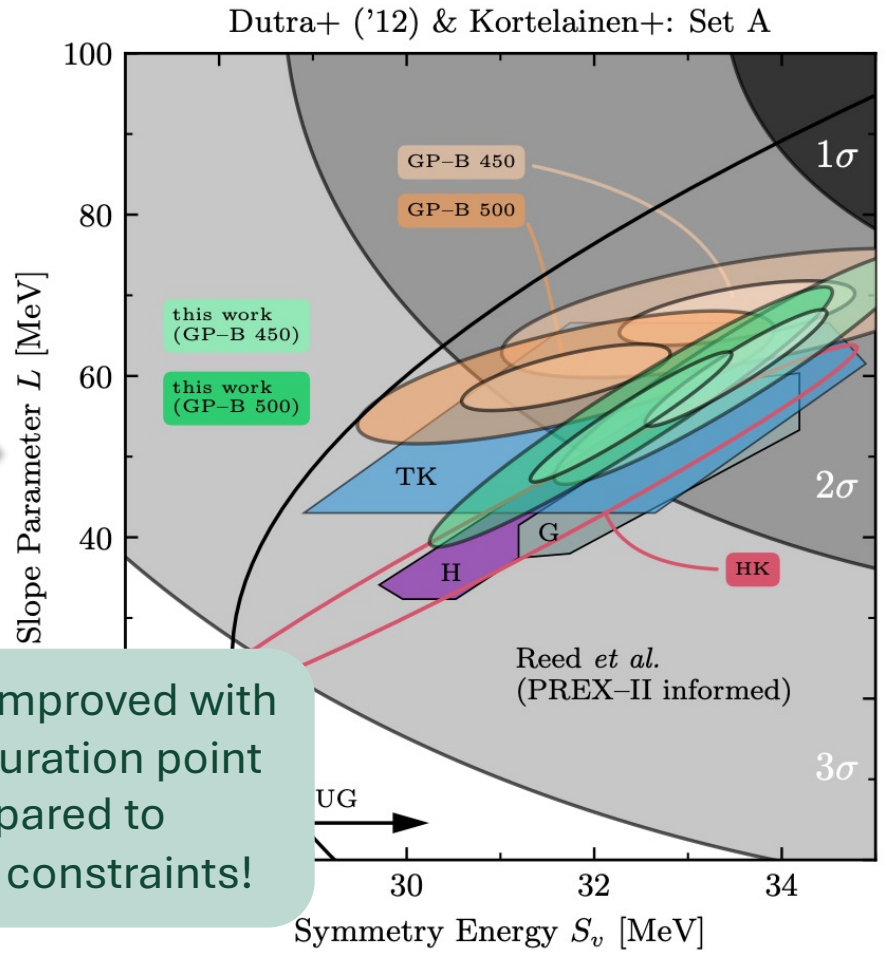
Constraints on the EOS lead to constraints on symmetry energy and slope parameter...



C. Drischler et al. PRL (2020)



which can be improved with UQ on the saturation point and compared to experimental constraints!



C. Drischler et al. PRC (2025)