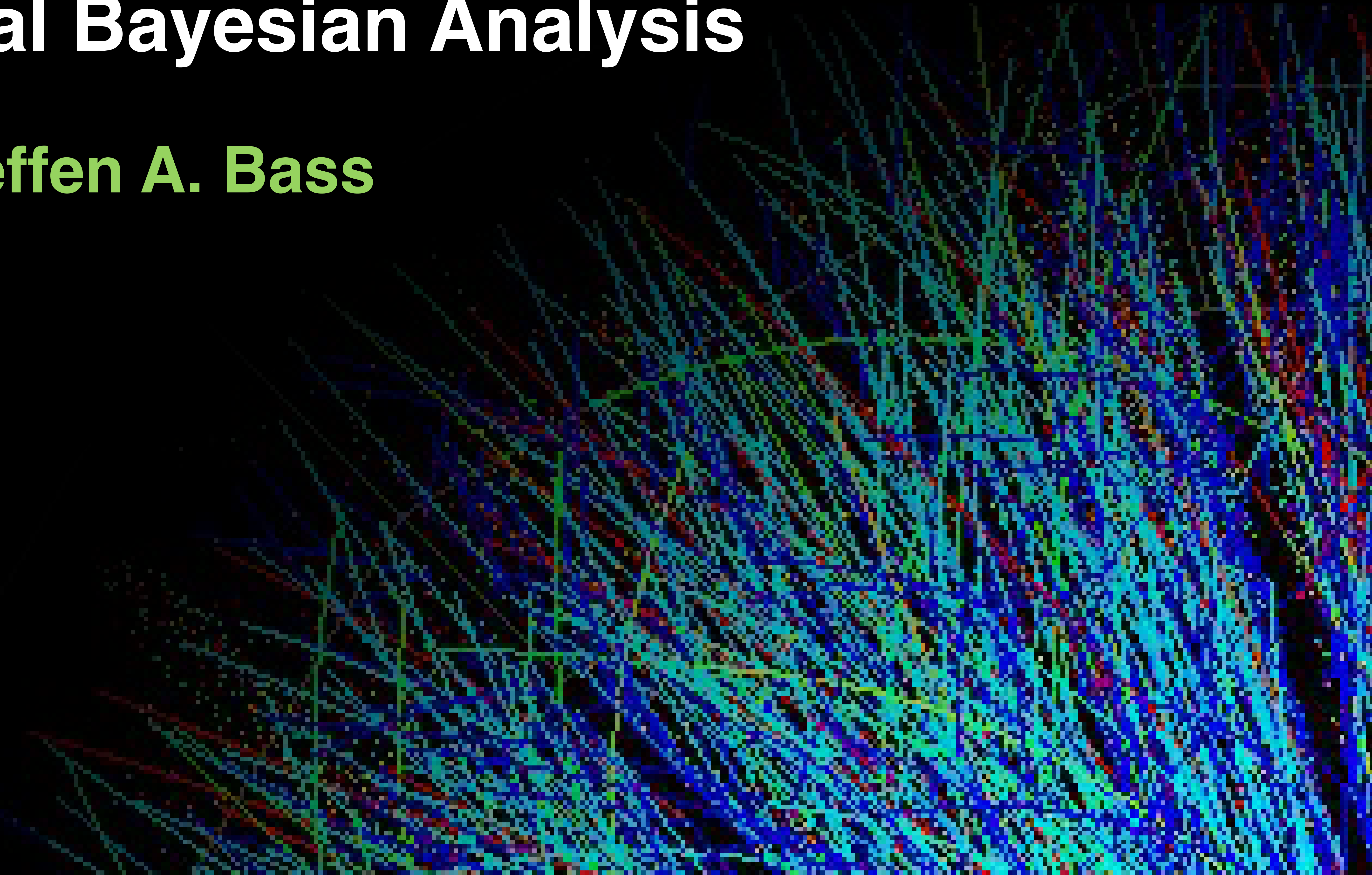
Determination of QGP Parameters from a Global Bayesian Analysis

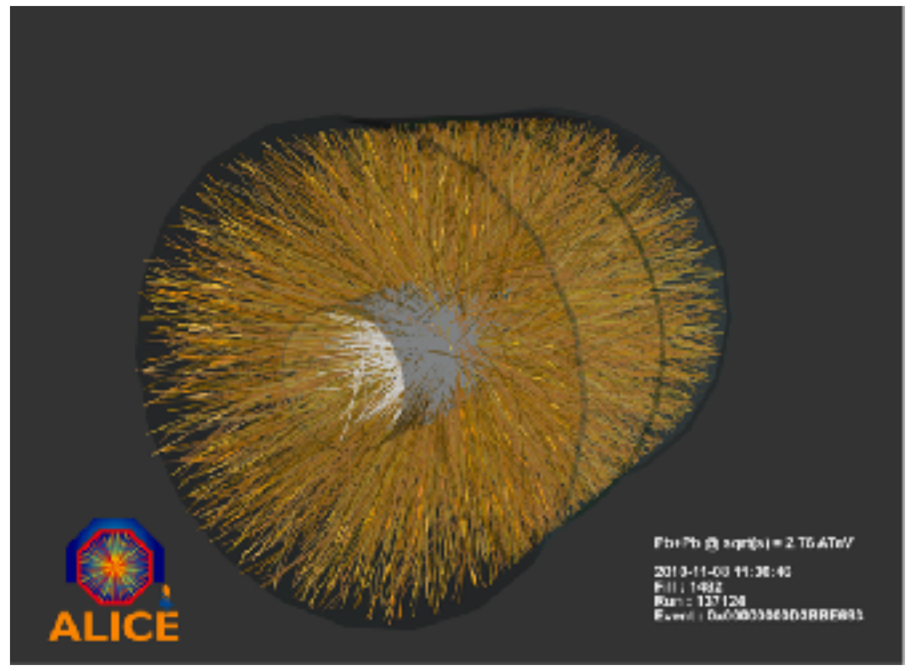
Steffen A. Bass

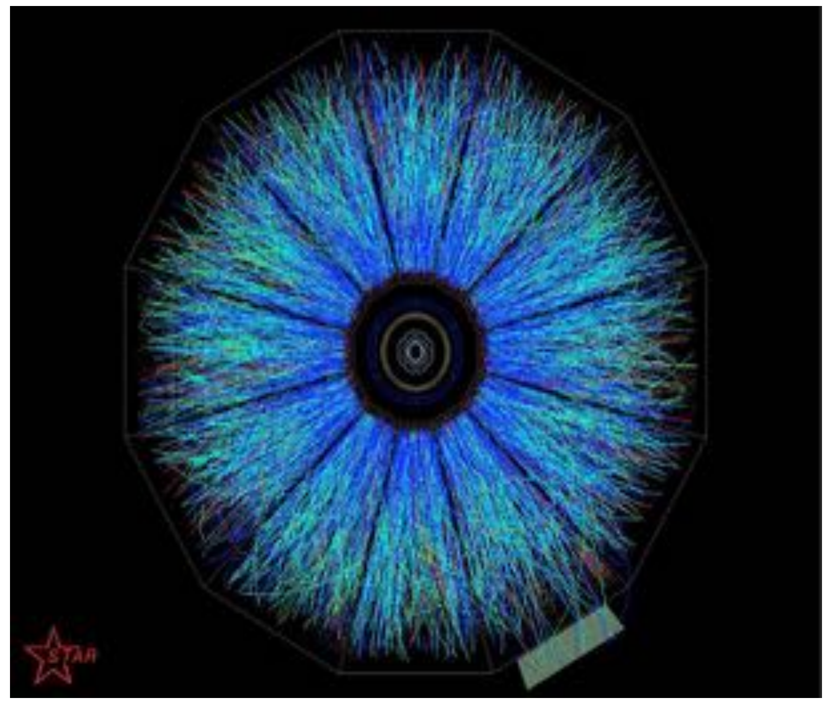


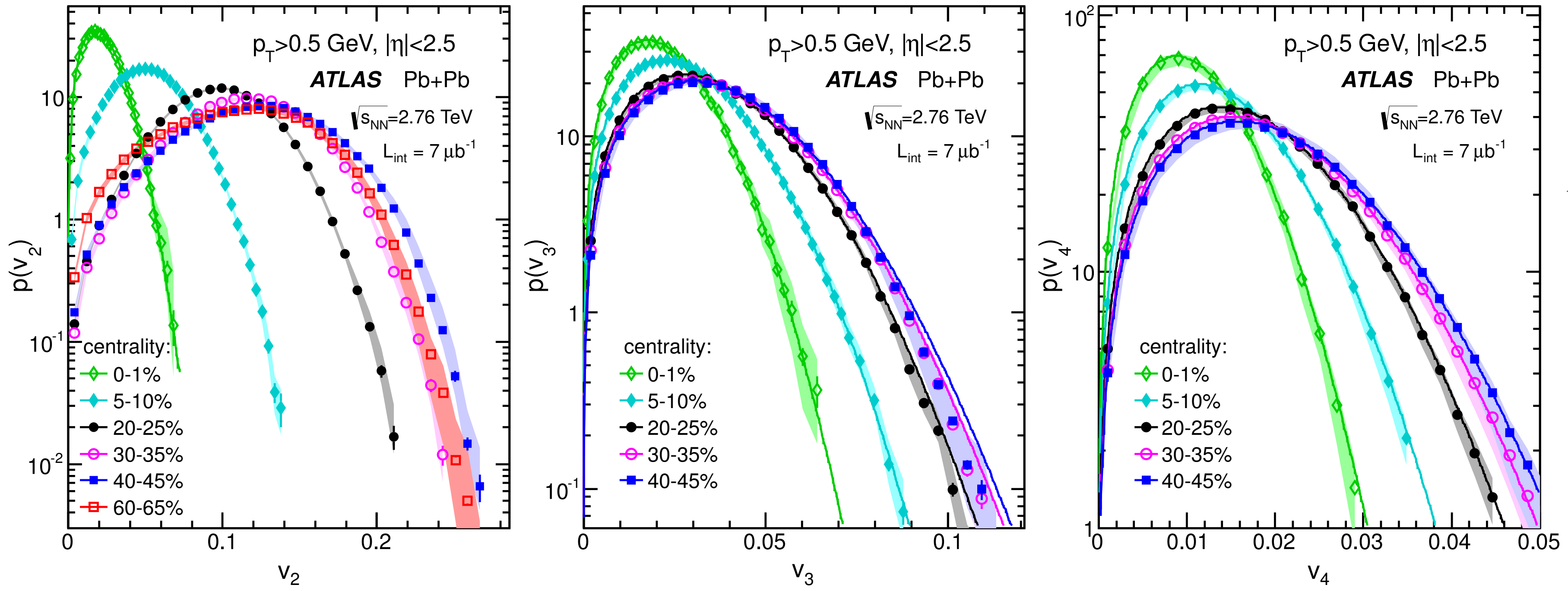
Knowledge Extraction from Relativistic Heavy-Ion Collisions

Probing QCD in Heavy-Ion Collisions

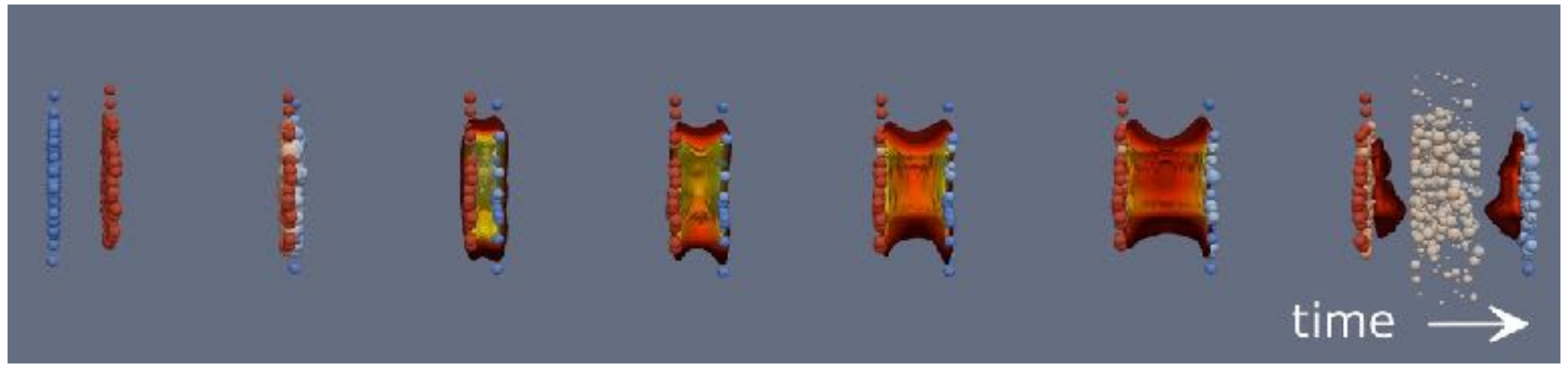
Data:

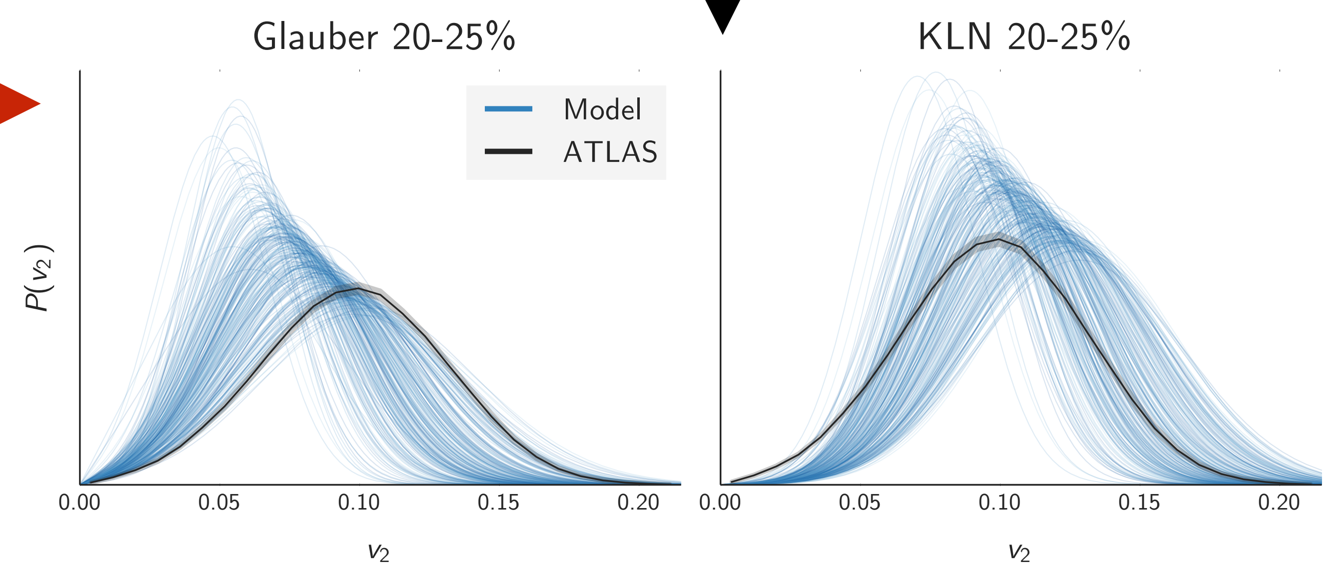






Model:
initial conditions, τ_0, η/s, ζ/s, ….





extracted QGP properties: η/s, …

Principal Challenges of Probing the QGP with Heavy-Ion Collisions:

- confinement: quarks & gluons form bound states, experiments don't observe them directly
- observables are limited to final state hadrons - we want to learn about the dynamics of the collision
- computational models are expensive: about 1 cpu-hour per simulated event

Computational Modeling of Relativistic Heavy-Ion Collisions

nuclei at 99.99% speed of light

Quark-Gluon-Plasma

hadronic final state interactions

measurable (stable) particles in detector

1x 10^{-23} s

10 x 10^{-23} s

30 x 10^{-23} s

non-equilibrium early time dynamics → viscous fluid dynamics → hadronic transport

Physics Model:

- multi-scale dynamical model comprised of three well-established codes:
  - TRENTO initial condition model
  - VISHNU 2+1D relativistic viscous hydro
  - URQMD: hadronic Boltzmann transport

Model-to-data comparison challenges:

- large number of interconnected parameters w/ non-factorizable data dependencies
- require novel optimization techniques: Bayesian Statistics and MCMC methods
- transport models require too much CPU: need new techniques based on emulators
- general problem, not restricted to heavy-ion physics

→collaboration with Statistical Sciences

Bayesian Analysis

Each computational model relies on a set of physics parameters to describe the dynamics and properties of the system. These physics parameters act as a representation of the information we wish to extract from RHIC & LHC.

Model Parameters - System Properties
- initial state
- temperature-dependent viscosities
- hydro to micro switching temperature

estimate or calculate parameters

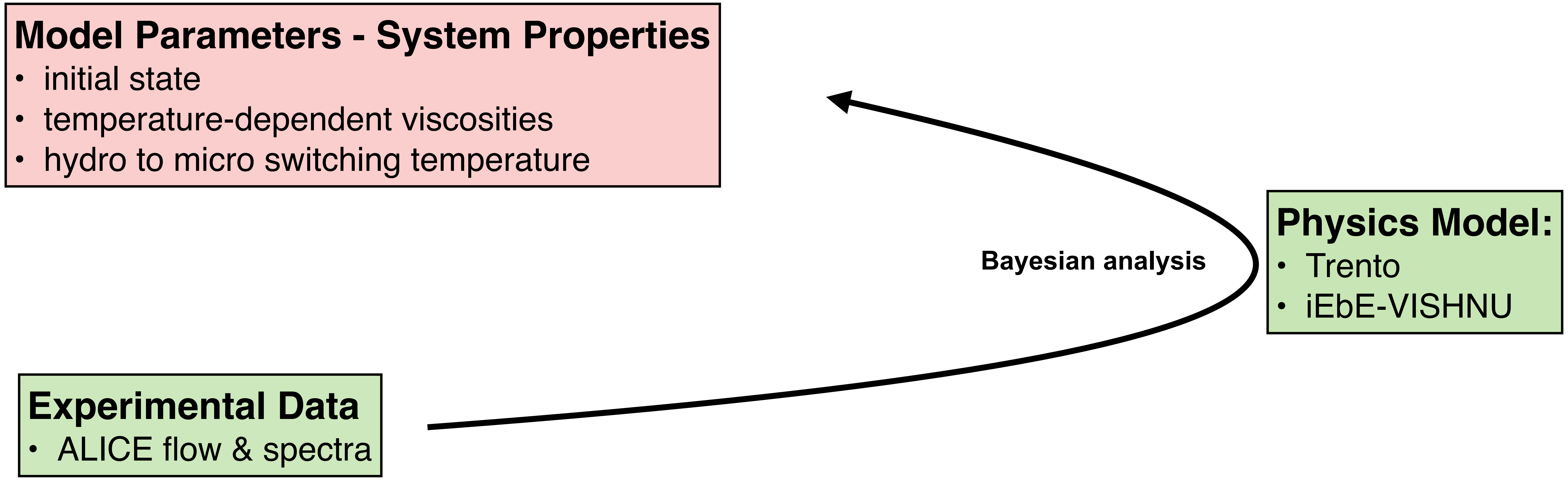
Physics Model:
- Trento
- iEbE-VISHNU

calculate observables & compare to data

Experimental Data
- ALICE flow & spectra

Bayesian Analysis

Each computational model relies on a set of physics parameters to describe the dynamics and properties of the system. These physics parameters act as a representation of the information we wish to extract from RHIC & LHC.



Bayesian Analysis

Each computational model relies on a set of physics parameters to describe the dynamics and properties of the system. These physics parameters act as a representation of the information we wish to extract from RHIC & LHC.

Model Parameters - System Properties
- initial state
- temperature-dependent viscosities
- hydro to micro switching temperature

Bayesian analysis

Physics Model:
- Trento
- iEbE-VISHNU

Experimental Data
- ALICE flow & spectra

- Bayesian analysis allows us to simultaneously calibrate all model parameters via a model-to-data comparison
- determine parameter values such that the model best describes experimental observables
- extract the probability distributions of all parameters

Setup of a Bayesian Statistical Analysis

Model Parameters - System Properties
- initial state
- temperature-dependent viscosities
- hydro to micro switching temperature

calculate events on Latin hypercube

Physics Model:
- Trento
- iEbE-VISHNU

Experimental Data
- ALICE flow & spectra

Gaussian Process Emulator
- non-parametric interpolation
- fast surrogate to full Physics Model

MCMC
(Markov-Chain Monte-Carlo)
- random walk through parameter space weighted by posterior probability

after many steps, MCMC equilibrates to

Posterior Distribution
- **diagonals**: probability distribution of each parameter, integrating out all others
- **off-diagonals**: pairwise distributions showing dependence between parameters

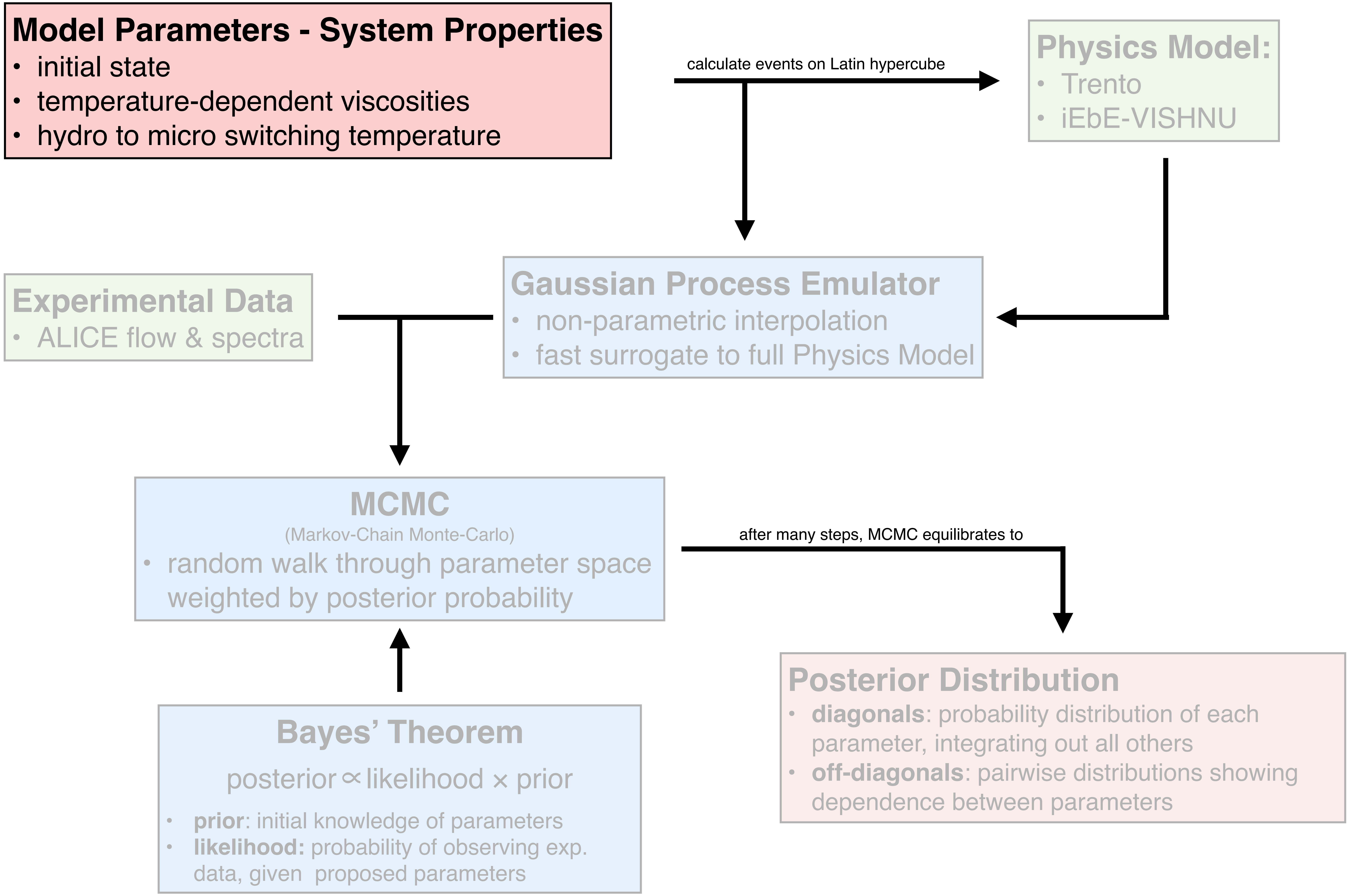
Bayes' Theorem

posterior $\propto$ likelihood $\times$ prior

- **prior**: initial knowledge of parameters
- **likelihood:** probability of observing exp. data, given proposed parameters

Components of the Bayesian Analysis

Methodology



Calibration Parameters

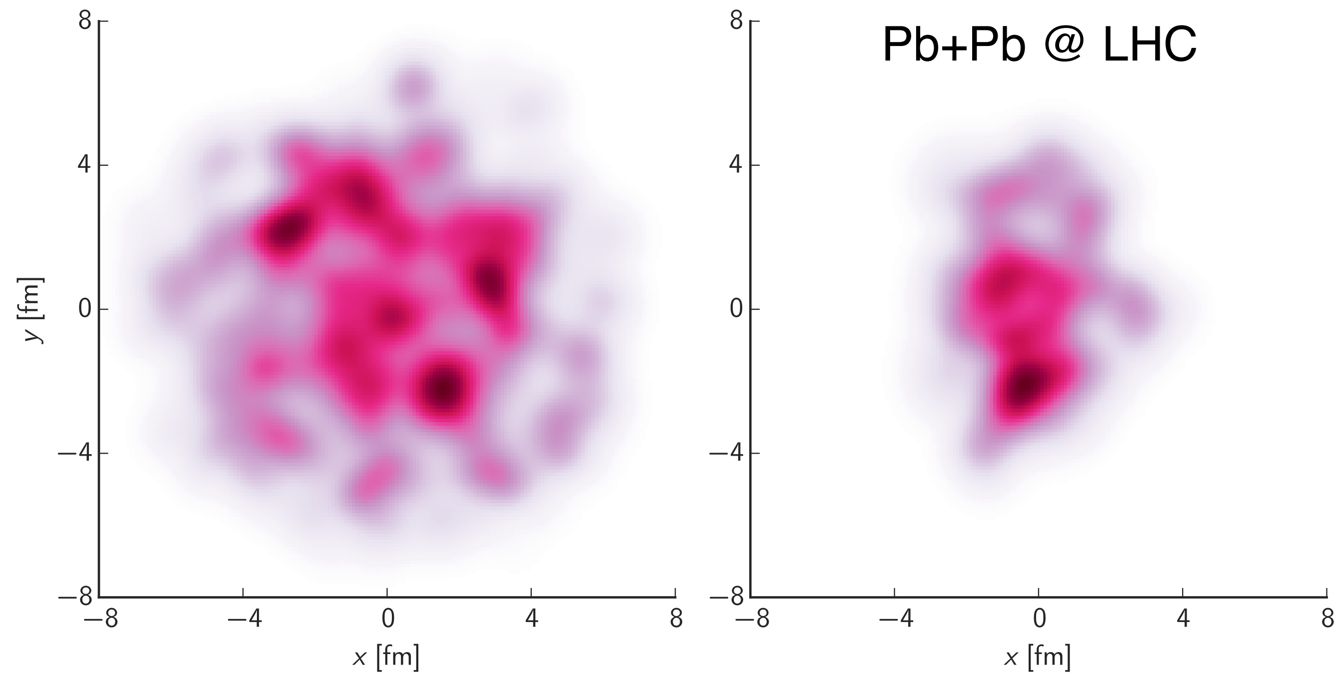
- the calibration parameters are the model parameters that codify the physical properties of the system that we wish to characterize with the analysis

Calibration Parameters

- the calibration parameters are the model parameters that codify the physical properties of the system that we wish to characterize with the analysis

Trento initial condition:

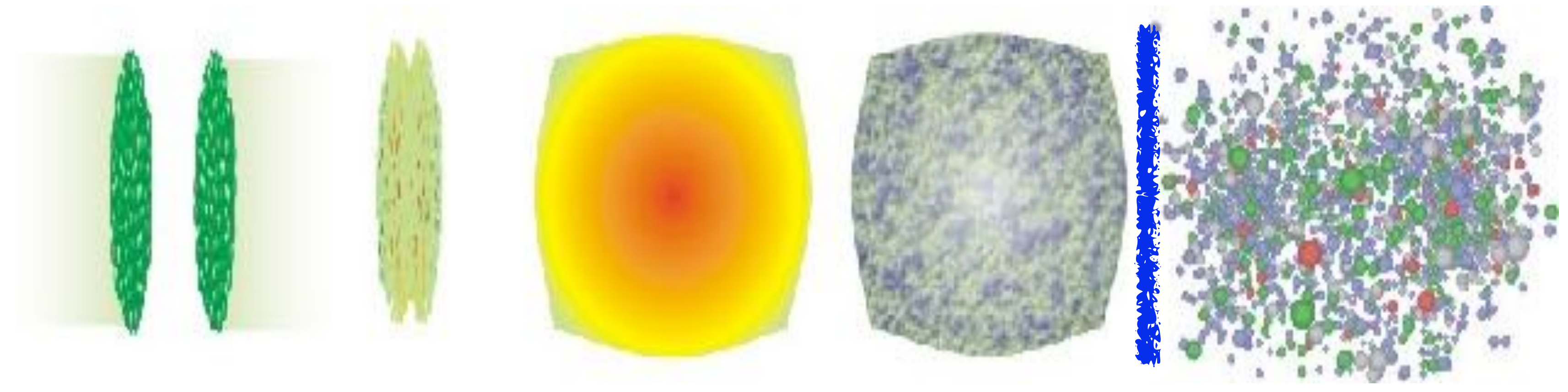
- p: attenuation parameter - entropy deposition
- k: governs fluctuation in nuclear thickness
- w: Gaussian nucleon width



Calibration Parameters

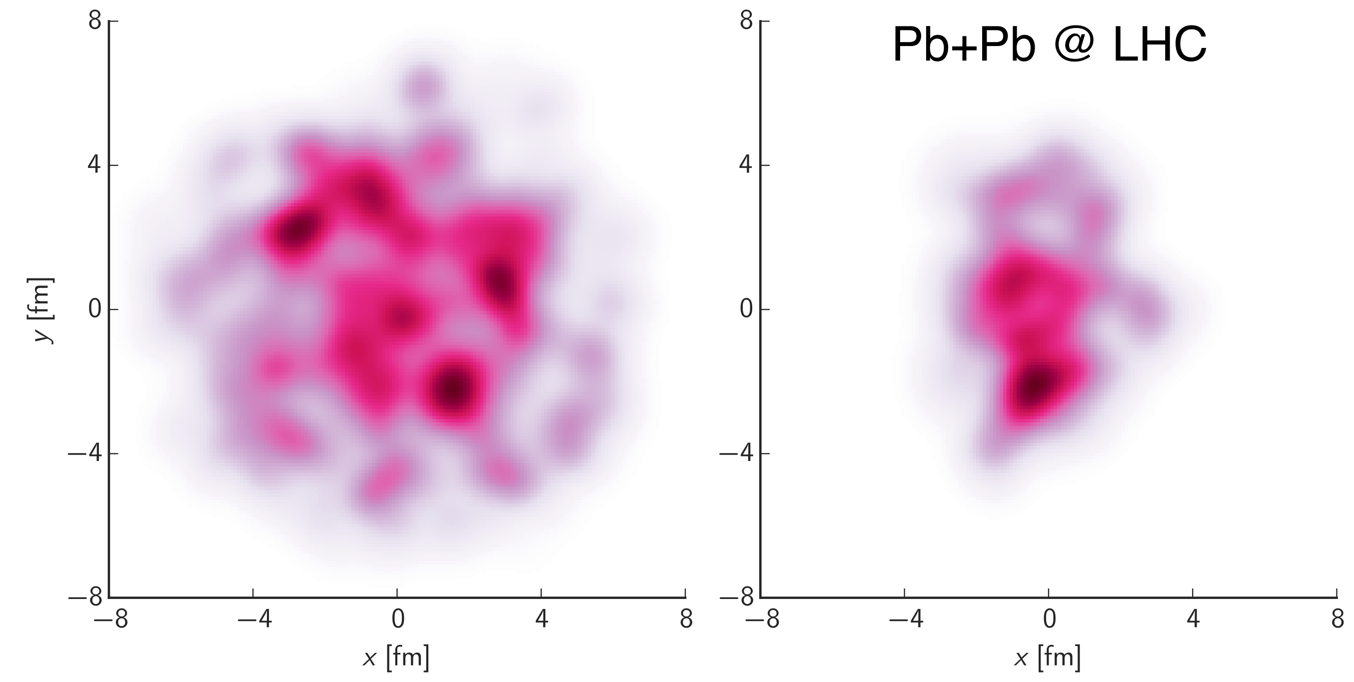
- the calibration parameters are the model parameters that codify the physical properties of the system that we wish to characterize with the analysis

- hydro to micro switching temperature T_{sw}



Trento initial condition:

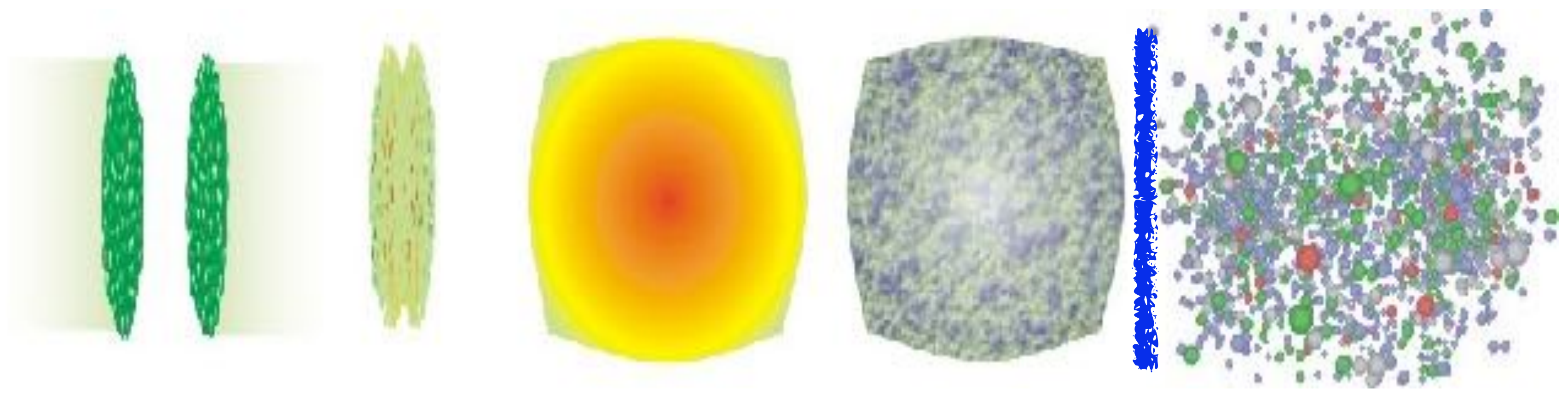
- p: attenuation parameter - entropy deposition
- k: governs fluctuation in nuclear thickness
- w: Gaussian nucleon width



Calibration Parameters

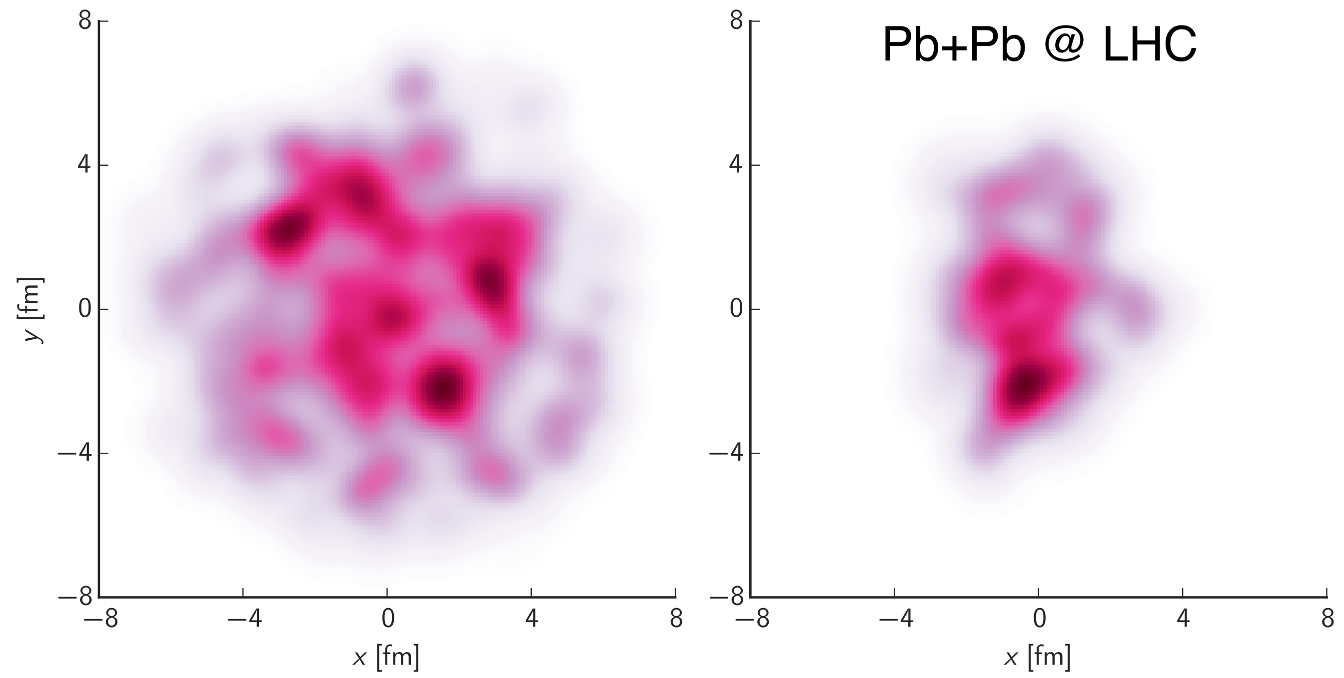
- the calibration parameters are the model parameters that codify the physical properties of the system that we wish to characterize with the analysis

- hydro to micro switching temperature T_{sw}



Trento initial condition:

- p: attenuation parameter - entropy deposition
- k: governs fluctuation in nuclear thickness
- w: Gaussian nucleon width

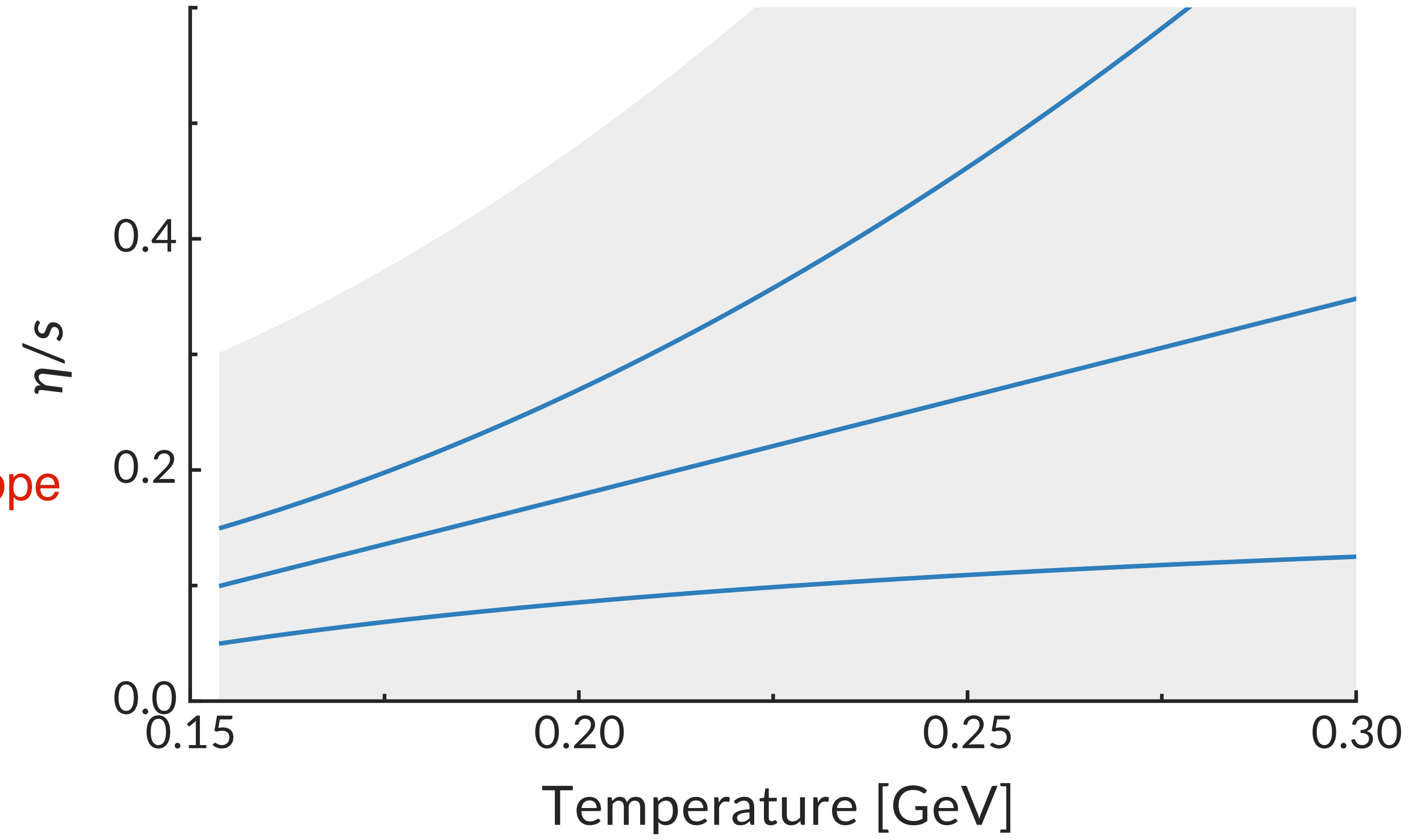


temperature dependent shear viscosity:

$$\eta/s(T) = (\eta/s)_{min} + (\eta/s)_{slope} \times (T-T_C)\times(T/T_C)^{\beta}$$

parameters:

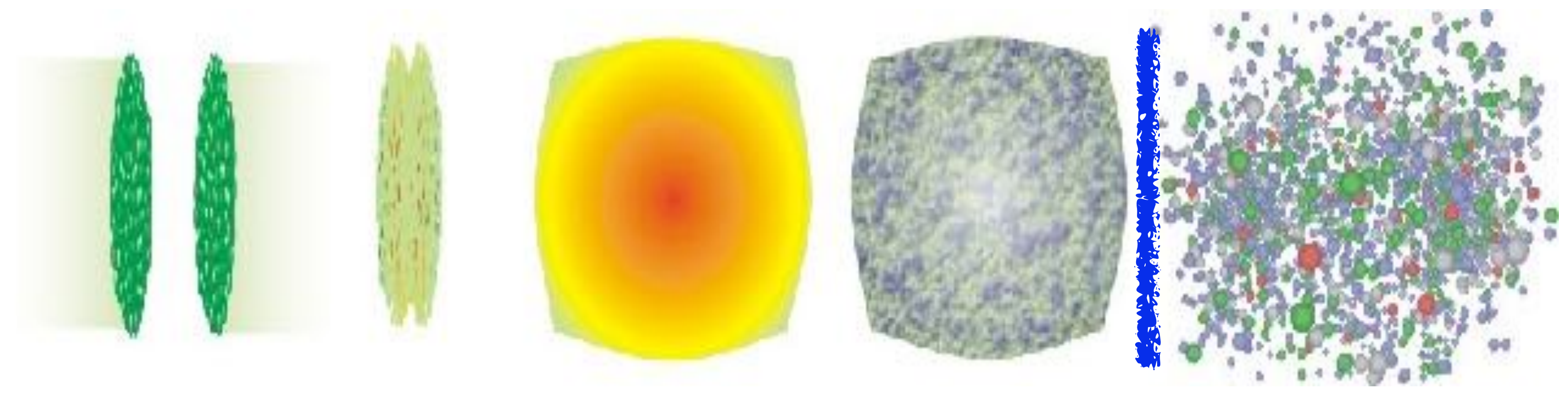
- intercept: $(\eta/s)_{min}$ at T_C
- slope: $(\eta/s)_{slope}$
- curvature: β



Calibration Parameters

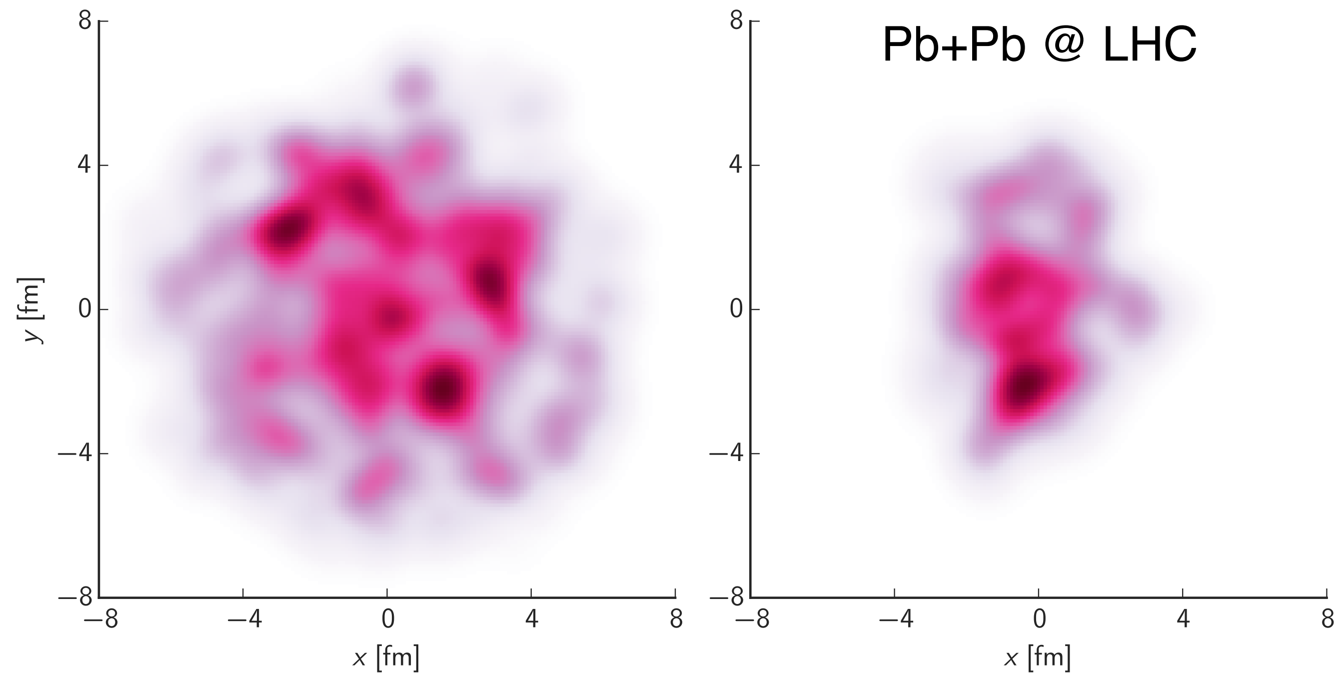
- the calibration parameters are the model parameters that codify the physical properties of the system that we wish to characterize with the analysis

- hydro to micro switching temperature T_{sw}



Trento initial condition:

- p: attenuation parameter - entropy deposition
- k: governs fluctuation in nuclear thickness
- w: Gaussian nucleon width

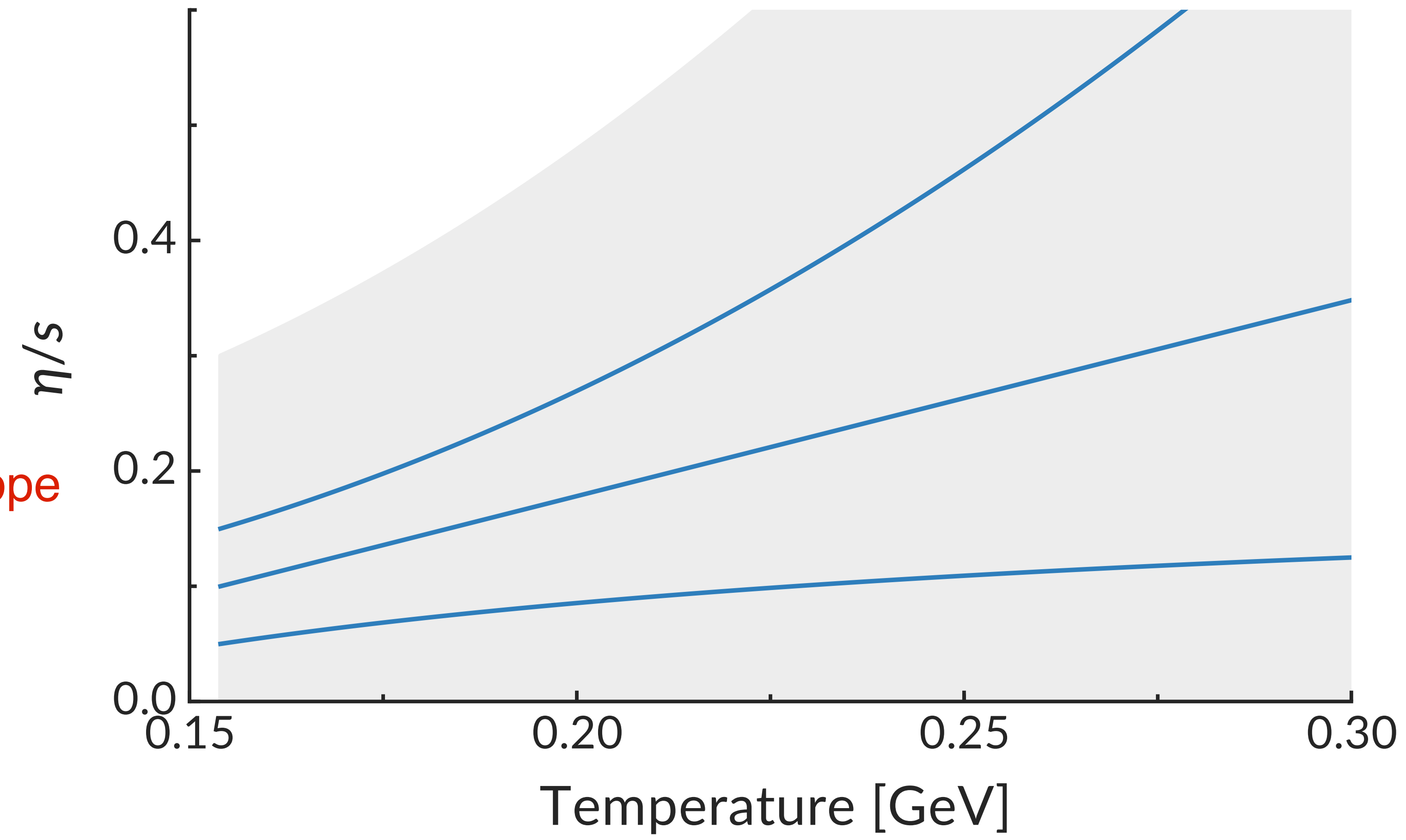


temperature dependent shear viscosity:

$$\eta/s(T) = (\eta/s)_{min} + (\eta/s)_{slope} \times (T-T_C)\times(T/T_C)^{\beta}$$

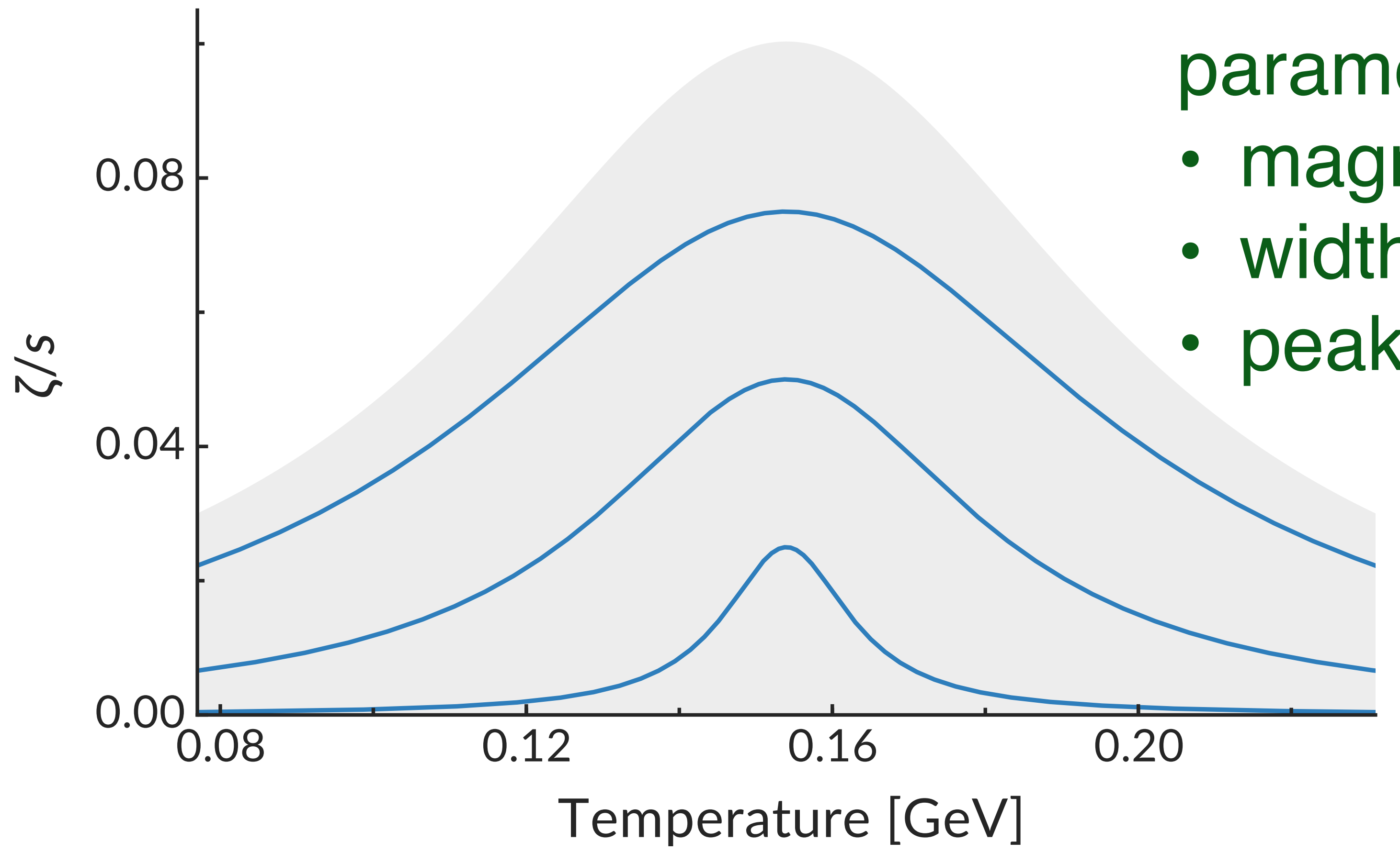
parameters:

- intercept: $(\eta/s)_{min}$ at T_C
- slope: $(\eta/s)_{slope}$
- curvature: β



temperature dependent bulk viscosity:

$$\zeta/s(T)=(\zeta/s)_{max} / [1+(T-(\zeta/s)_{peak})^2/\Gamma^2]$$



parameters:

- magnitude $(\zeta/s)_{max}$
- width: Γ
- peak position: $(\zeta/s)_{peak}$

Methodology

Model Parameters - System Properties
- initial state
- temperature-dependent viscosities
- hydro to micro switching temperature

calculate events on Latin hypercube

Physics Model:
- Trento
- iEbE-VISHNU

Experimental Data
- ALICE flow & spectra

Gaussian Process Emulator
- non-parametric interpolation
- fast surrogate to full Physics Model

MCMC
(Markov-Chain Monte-Carlo)
- random walk through parameter space weighted by posterior probability

after many steps, MCMC equilibrates to

Bayes' Theorem

posterior ∝ likelihood × prior

- **prior**: initial knowledge of parameters
- **likelihood:** probability of observing exp. data, given proposed parameters

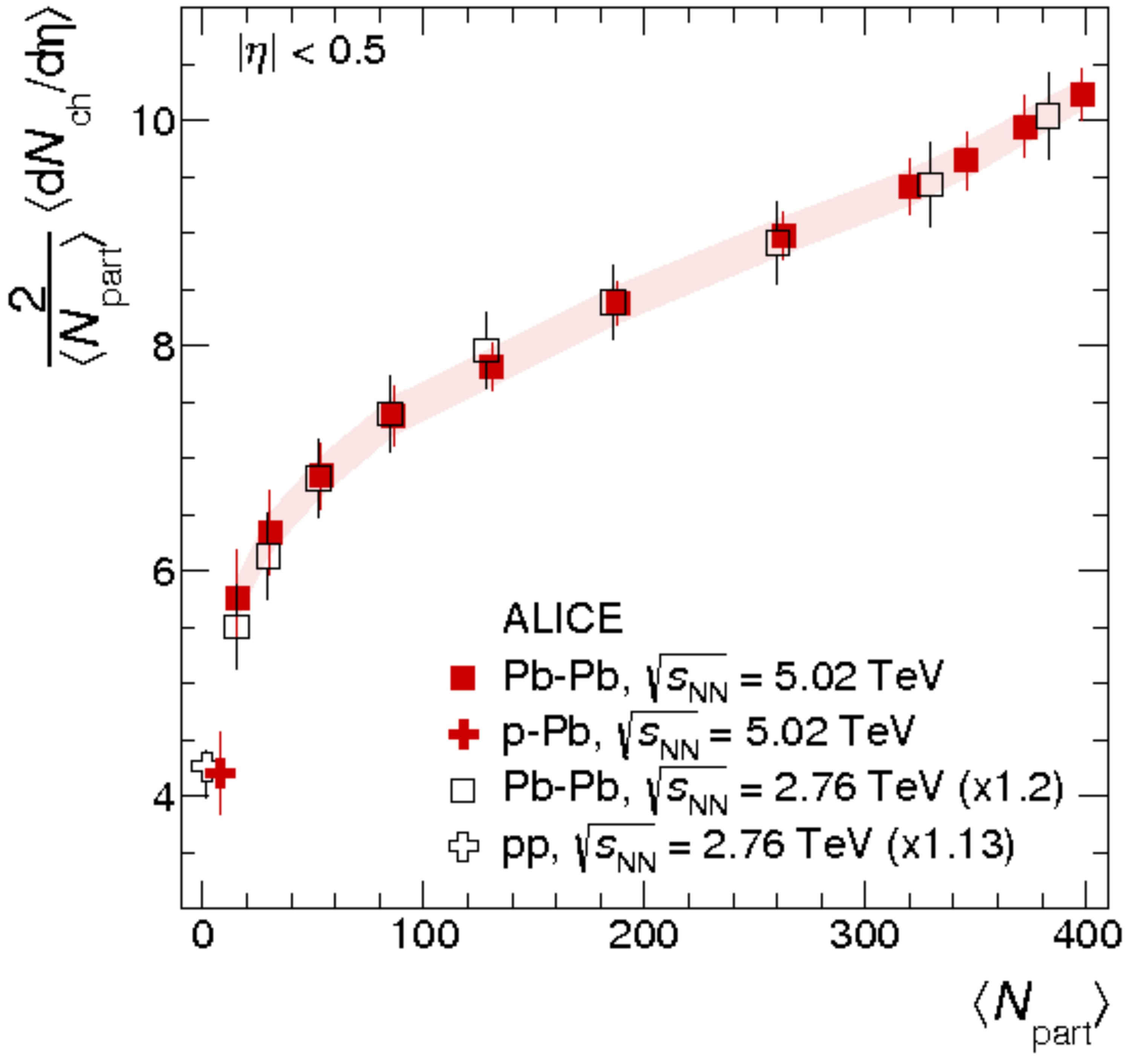
Posterior Distribution
- **diagonals**: probability distribution of each parameter, integrating out all others
- **off-diagonals**: pairwise distributions showing dependence between parameters

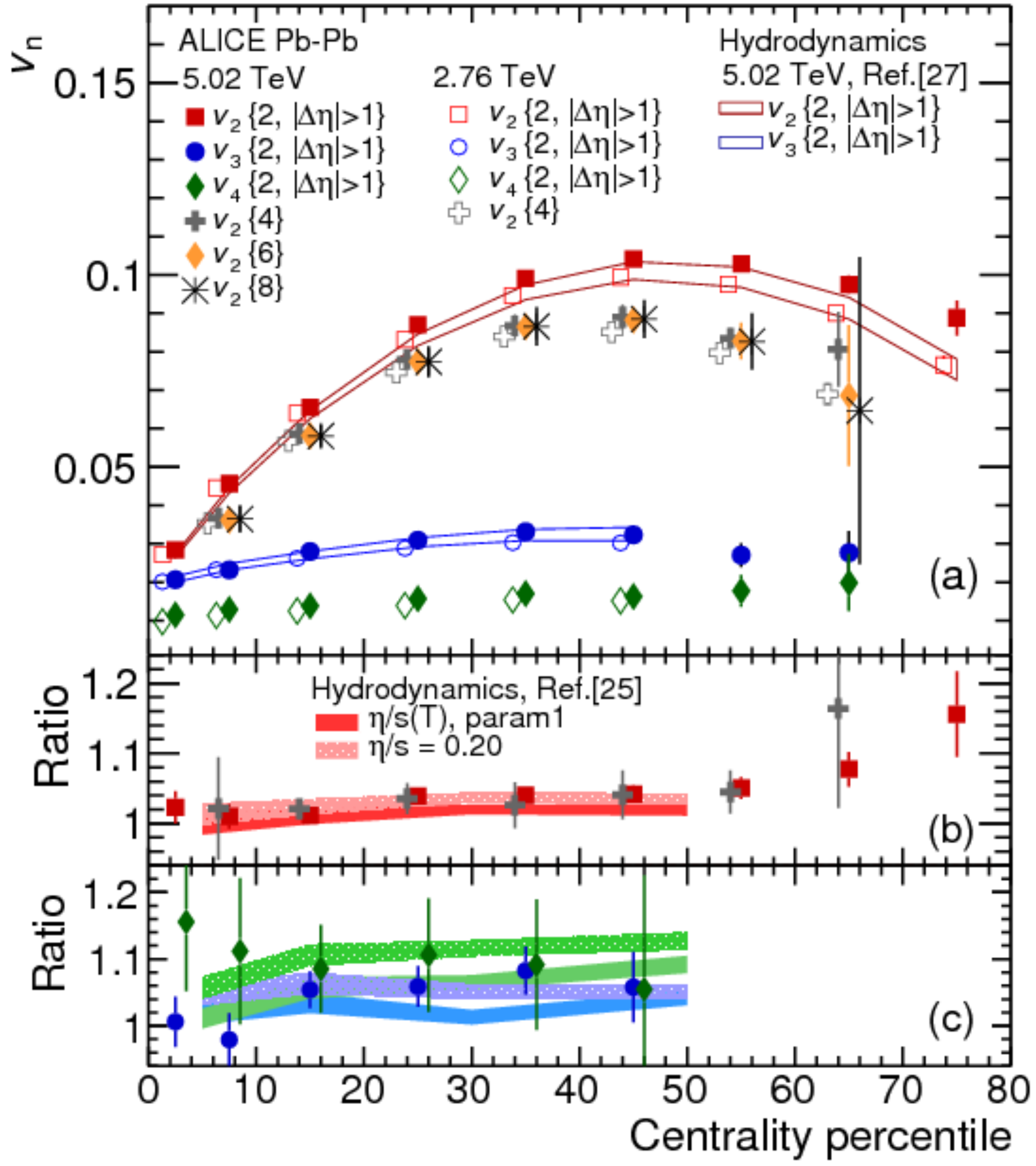
Training Data

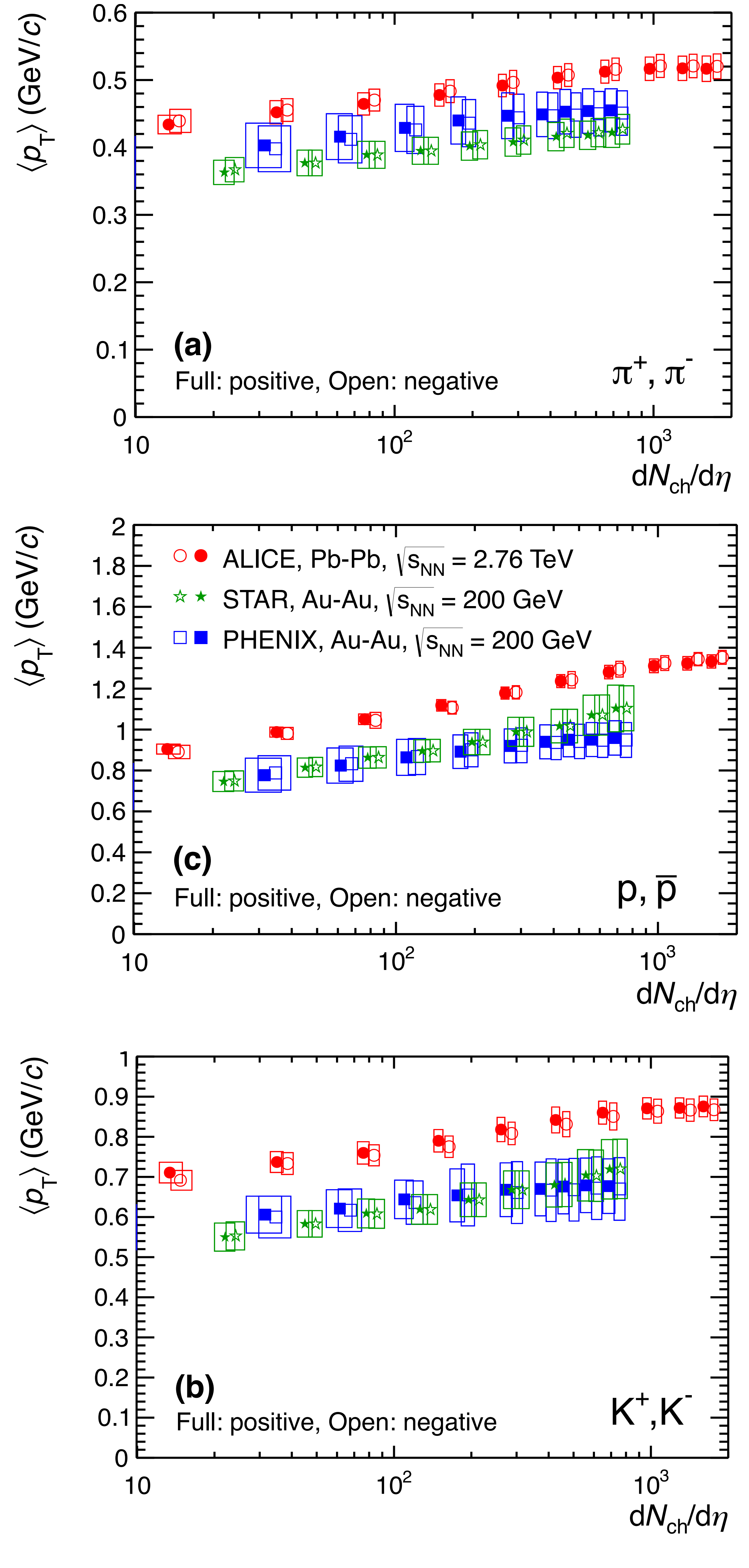
Data:

- ALICE v_2, v_3 & v_4 flow cumulants
- identified & charged particle yields
- identified particle mean p_T
- 2 beam energies:
  2.76 & 5.02 TeV

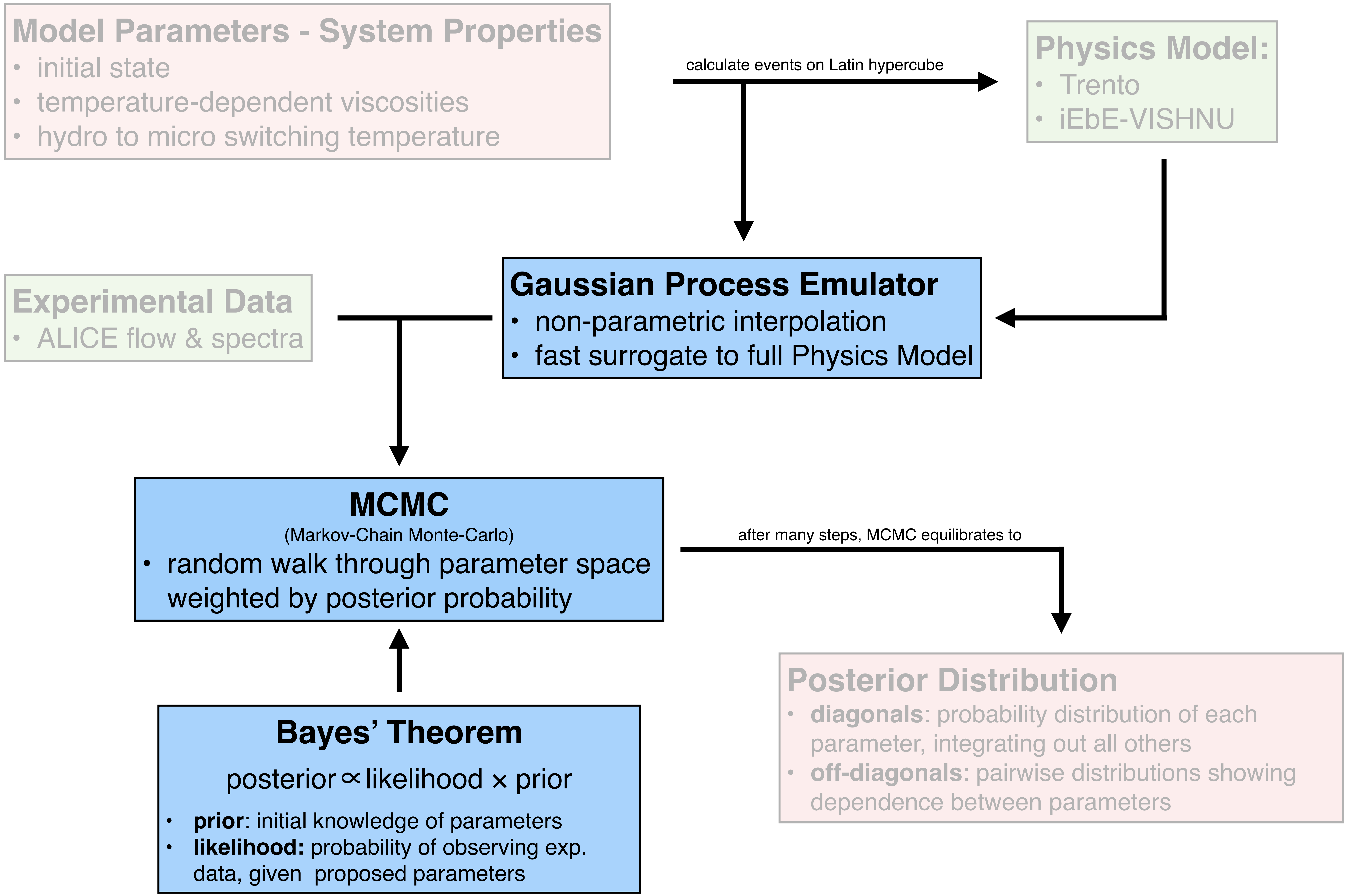
the entire success of the analysis depends on the quality of the exp. data!







Methodology



Exploring the Model Parameter-Space

Exploring the Model Parameter-Space

brute force analysis:

- 14 model parameters
- 9 centrality bins
- 20 bins per parameter
- need to evaluate model at 9×20^{14} points
- fluctuating initial conditions: $\mathcal{O}(10^4)$ events per point $\rightarrow 10^{18}$ events
- assume 1 cpu hour per event: 10^{18} cpu-hours!
- **2 billion years 100% use of TITAN @ ORNL (Cray XK7 w/ 560,640 cores)**
- then start MCMC to find point that optimally describes data…

Exploring the Model Parameter-Space

brute force analysis:

- 14 model parameters
- 9 centrality bins
- 20 bins per parameter
- need to evaluate model at 9×20^{14} points
- fluctuating initial conditions: $\mathcal{O}(10^4)$ events per point $\rightarrow 10^{18}$ events
- assume 1 cpu hour per event: 10^{18} cpu-hours!
- **2 billion years 100% use of TITAN @ ORNL (Cray XK7 w/ 560,640 cores)**
- then start MCMC to find point that optimally describes data…

Need to find techniques that cut down the cpu needed by at least a factor of 10^{10}: **Gaussian Process Emulators**

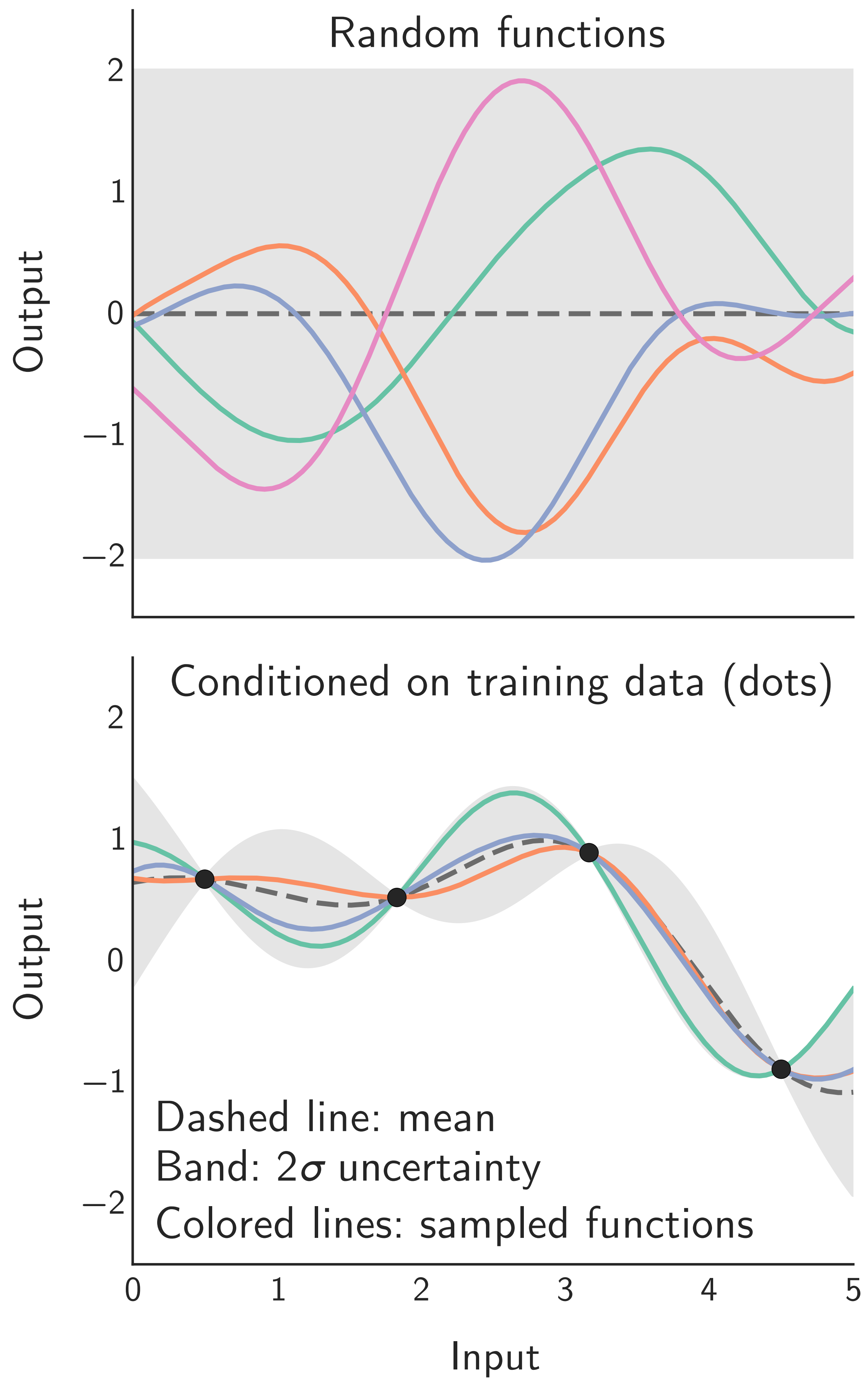
Exploring the Model Parameter-Space

Gaussian process:

- stochastic function: maps inputs to normally distributed outputs
- specified by mean and covariance functions

GP as a model emulator:

- non-parametric interpolation of physics model
- predicts probability distributions for model output at any given input value
  - narrow near training points, wide in gaps
- needs to be conditioned on training data (Latin hypercube points)
- fast *surrogate* to actual model



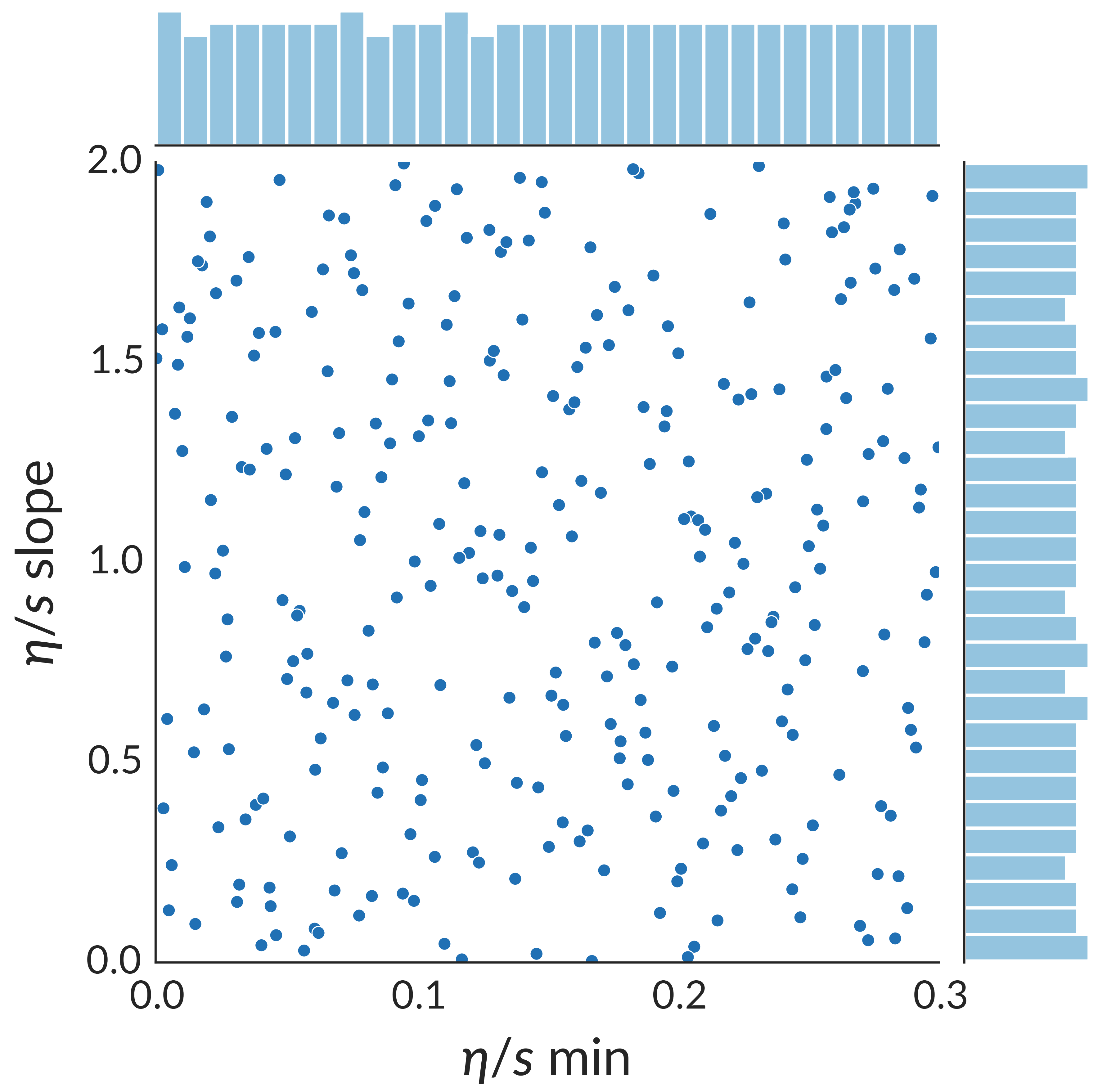
Computer Experiment Design

Latin hypercube:

- algorithm for generating semi-randomized, space-filling points (here: maximin Latin hypercube)
- avoids large gaps and tight clusters
- all parameters varied simultaneously
- needs only *m≥10n* points, with n: number of model parameters

Example:

- Latin-hypercube projection for η/s parameters



Computer Experiment Design

Latin hypercube:

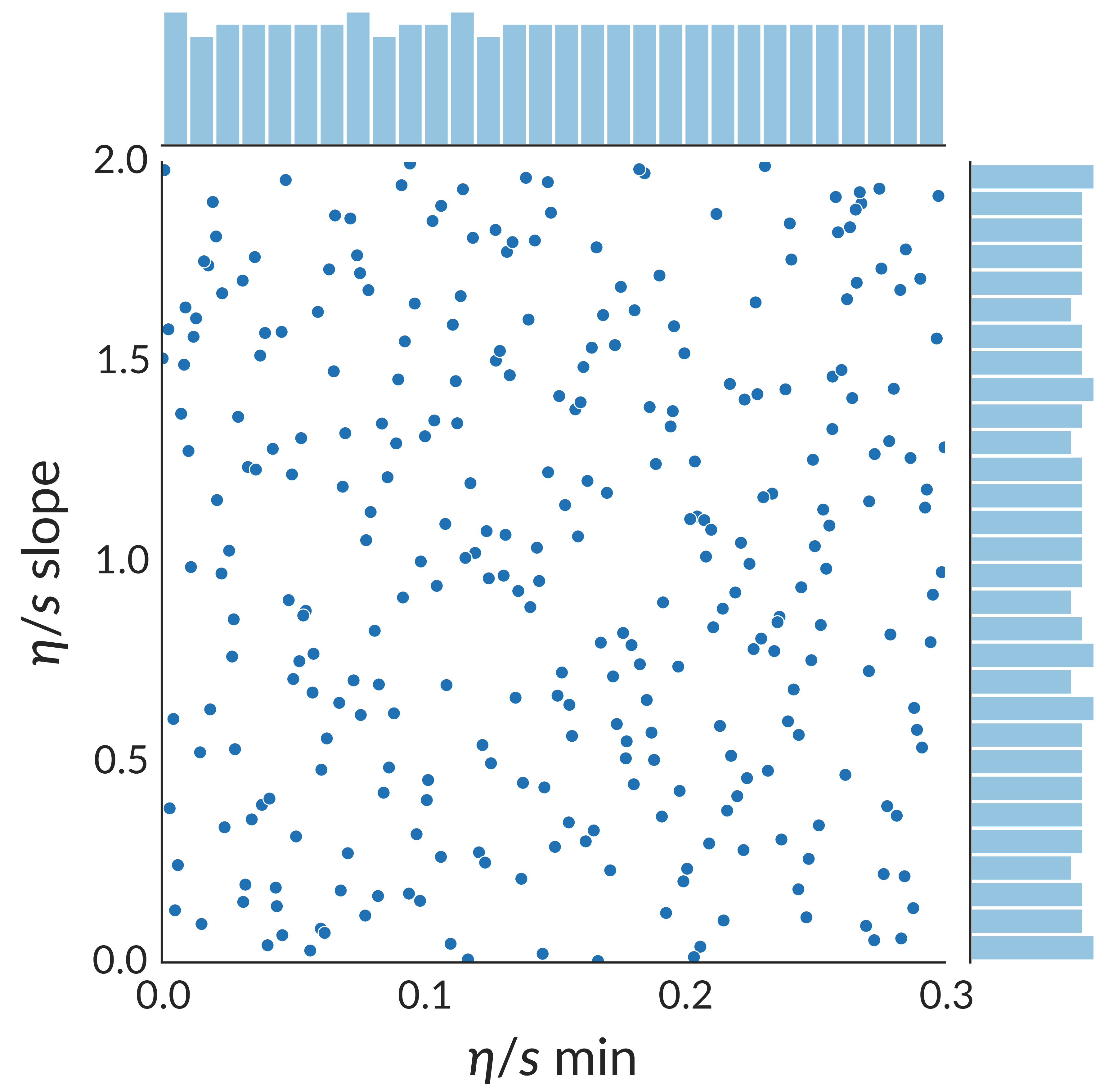
- algorithm for generating semi-randomized, space-filling points (here: maximin Latin hypercube)
- avoids large gaps and tight clusters
- all parameters varied simultaneously
- needs only *m≥10n* points, with n: number of model parameters

this design:

- n=15 model parameters
- 9 centrality bins, 2 energies
- Latin hypercube with m=500 points
- $\mathcal{O}(10^4)$ events per point, for a total of approx. 35,000,000 events
- use Gaussian Process Emulators to interpolate between points

Example:

- Latin-hypercube projection for η/s parameters



Computer Experiment Execution

Computer Experiment Execution

Edison @ NERSC:

- Cray XC30: 5586 nodes w/ 24 cores each
- 2 hyperthreads per core
- 2.57 Petaflops/s

Duke QCD workflow:

- 1000 nodes per job: running on 48K cores simultaneously
- entire model design with 30M events can be computed in 1 day

Computer Experiment Execution

Edison @ NERSC:
- Cray XC30: 5586 nodes w/ 24 cores each
- 2 hyperthreads per core
- 2.57 Petaflops/s

Duke QCD workflow:
- 1000 nodes per job: running on 48K cores simultaneously
- entire model design with 30M events can be computed in 1 day

NOW COMPUTING

A small sample of massively parallel scientific computing jobs running right now at NERSC.

PROJECT	MACHINE	NODES	NERSC HOURS USED
NERSC Staff Accounts PI: Sudip S. Dosanjh, Lawrence Berkeley National Lab - NERSC	Cori KNL	1,008	115,874.8
NERSC Staff Accounts PI: Sudip S. Dosanjh, Lawrence Berkeley National Lab - NERSC	Cori KNL	1,008	77,866.5
Extraction of QCD transport coefficients from ultra-relativistic heavy-ion collisions through a Bayesian model to data analysis PI: Steffen A. Bass, Duke University	Edison	1,000	443,890.9
Extraction of QCD transport coefficients from ultra-relativistic heavy-ion collisions through a Bayesian model to data analysis PI: Steffen A. Bass, Duke University	Edison	1,000	399,224.3
Extraction of QCD transport coefficients from ultra-relativistic heavy-ion collisions through a Bayesian model to data analysis PI: Steffen A. Bass, Duke University	Edison	750	229,928.2
NERSC Staff Accounts PI: Sudip S. Dosanjh, Lawrence Berkeley National Lab - NERSC	Cori KNL	512	282,594.2

Calibration

Vector of input parameters: $\mathbf{x}=[p,k,w,(\eta/s)_{min},(\eta/s)_{slope},(\zeta/s)_{norm},T_{sw},\ldots]$

- assume true parameters $\mathbf{x}_\star$ exist $\Rightarrow$ find probability distribution for $\mathbf{x}_\star$

- X: training data design points
- Y: model output on X

Bayes' Theorem: $P(\mathbf{x}_\star | X,Y,\mathbf{y}_{exp}) \propto P(X,Y,\mathbf{y}_{exp} | \mathbf{x}_\star)P(\mathbf{x}_\star)$

- $P(\mathbf{x}_\star | X,Y,\mathbf{y}_{exp})$ = posterior
  $\Rightarrow$ probability of $\mathbf{x}_\star$ given observations $(X,Y,\mathbf{y}_{exp})$

- $P(X,Y,\mathbf{y}_{exp} | \mathbf{x}_\star)$ = likelihood
  $\Rightarrow$ probability of observing $(X,Y,\mathbf{y}_{exp})$ given proposed $\mathbf{x}_\star$

- $P(\mathbf{x}_\star)$ = prior
  $\Rightarrow$ initial knowledge of $\mathbf{x}_\star$

Markov-Chain Monte-Carlo:

- random walk through parameter space weighted by posterior
- large number of samples
  $\Rightarrow$ chain equilibrates to posterior distribution
- flat prior within design range, zero outside
- posterior ~ likelihood within design range, zero outside

Likelihood and Uncertainty Quantification:

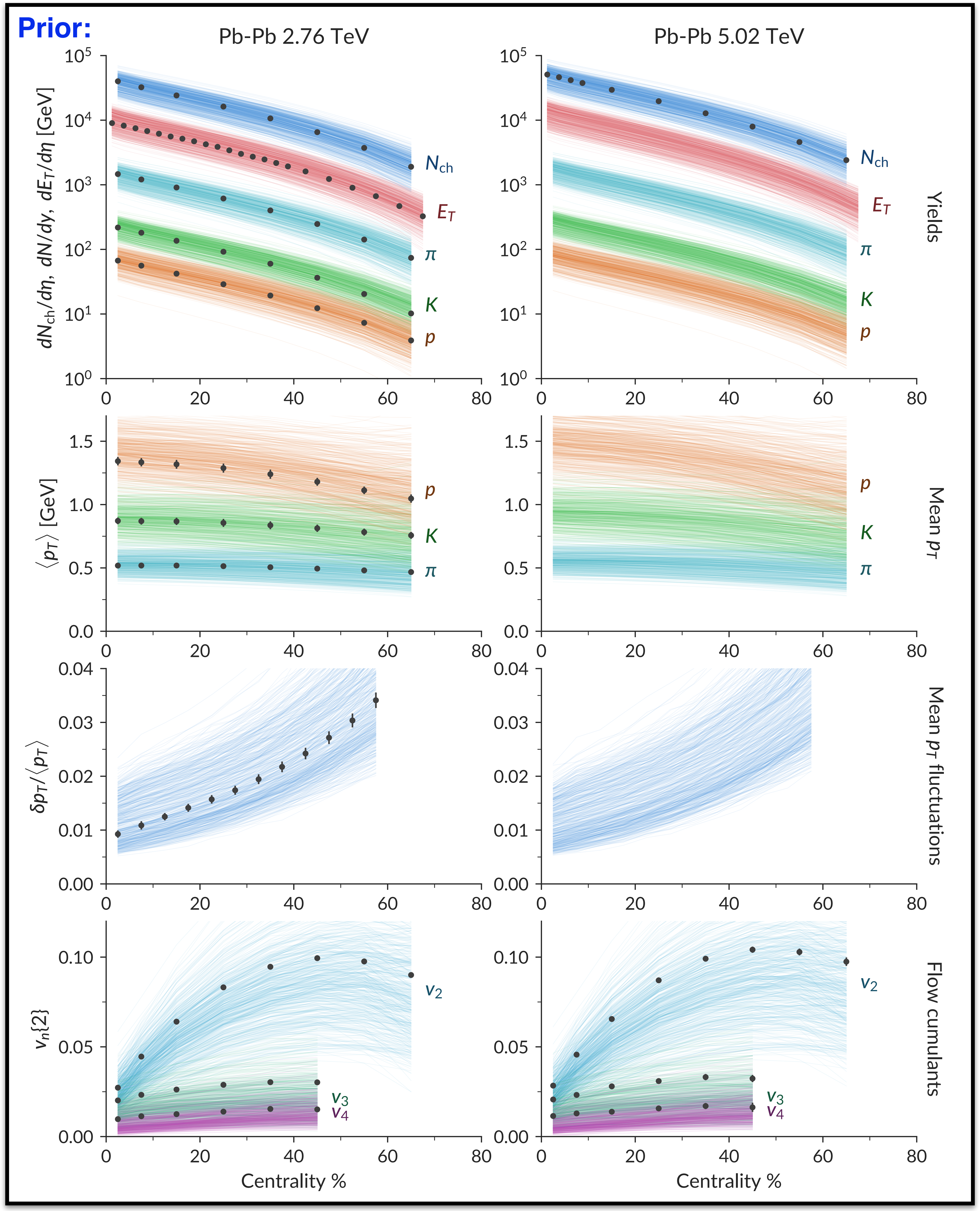
Likelihood $\propto \exp[-1/2\,(\mathbf{y}-\mathbf{y}_{exp})^T \boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{y}_{exp})]$

- covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{experiment} + \boldsymbol{\Sigma}_{model}$
- $\boldsymbol{\Sigma}_{experiment}$=stat(diagonal) + sys(non-diagonal)
- $\boldsymbol{\Sigma}_{model}$ conservatively estimated as 5%

Prior vs. Posterior

Prior:

- each line represents model evaluation at on point in latin hypercube
- covers full range of observable space



Prior vs. Posterior

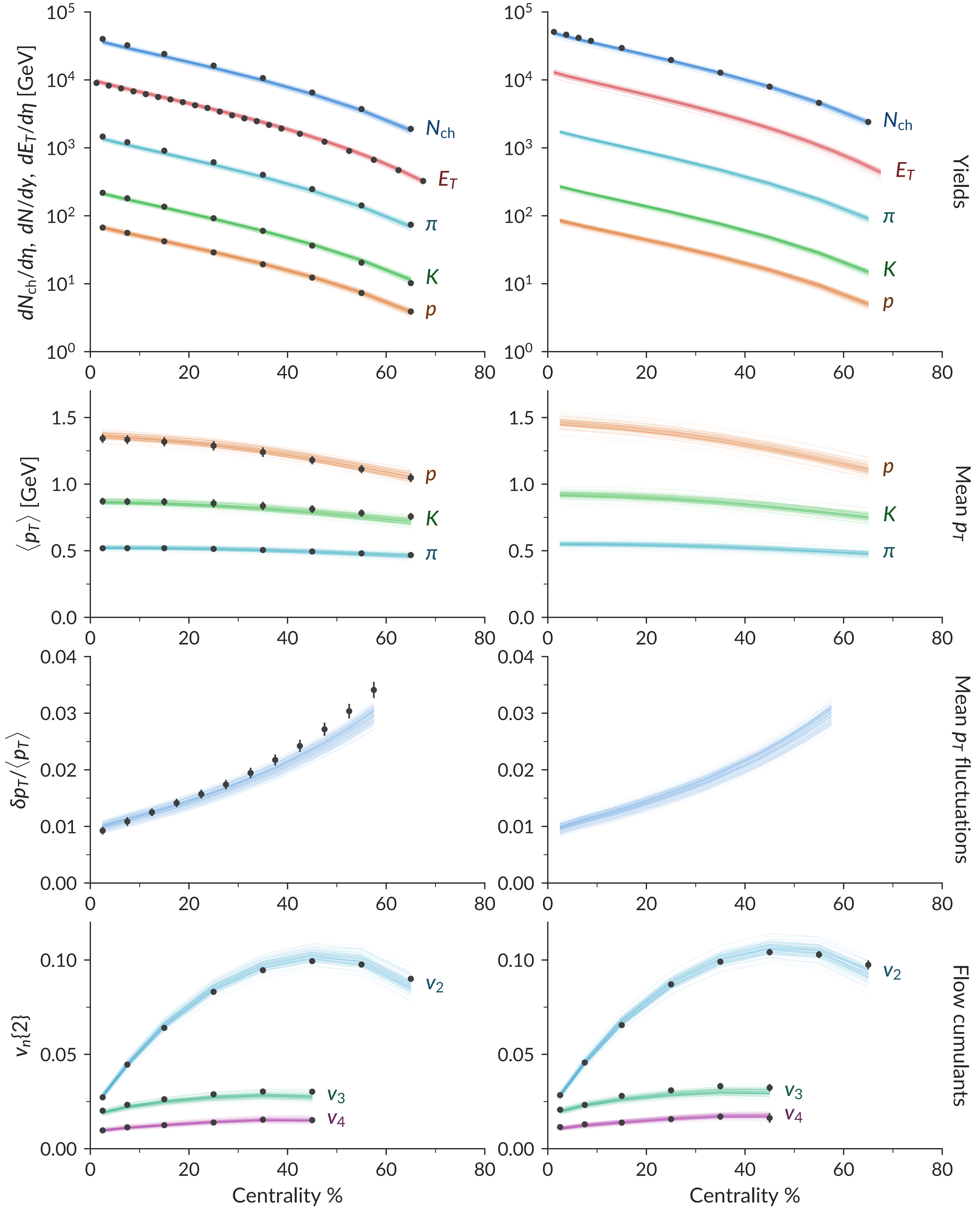
Prior:

- each line represents model evaluation at on point in latin hypercube
- covers full range of observable space

Posterior:

- each line represents model evaluation at points drawn from posterior probability distribution of parameters
- excellent agreement with data!

Posterior:



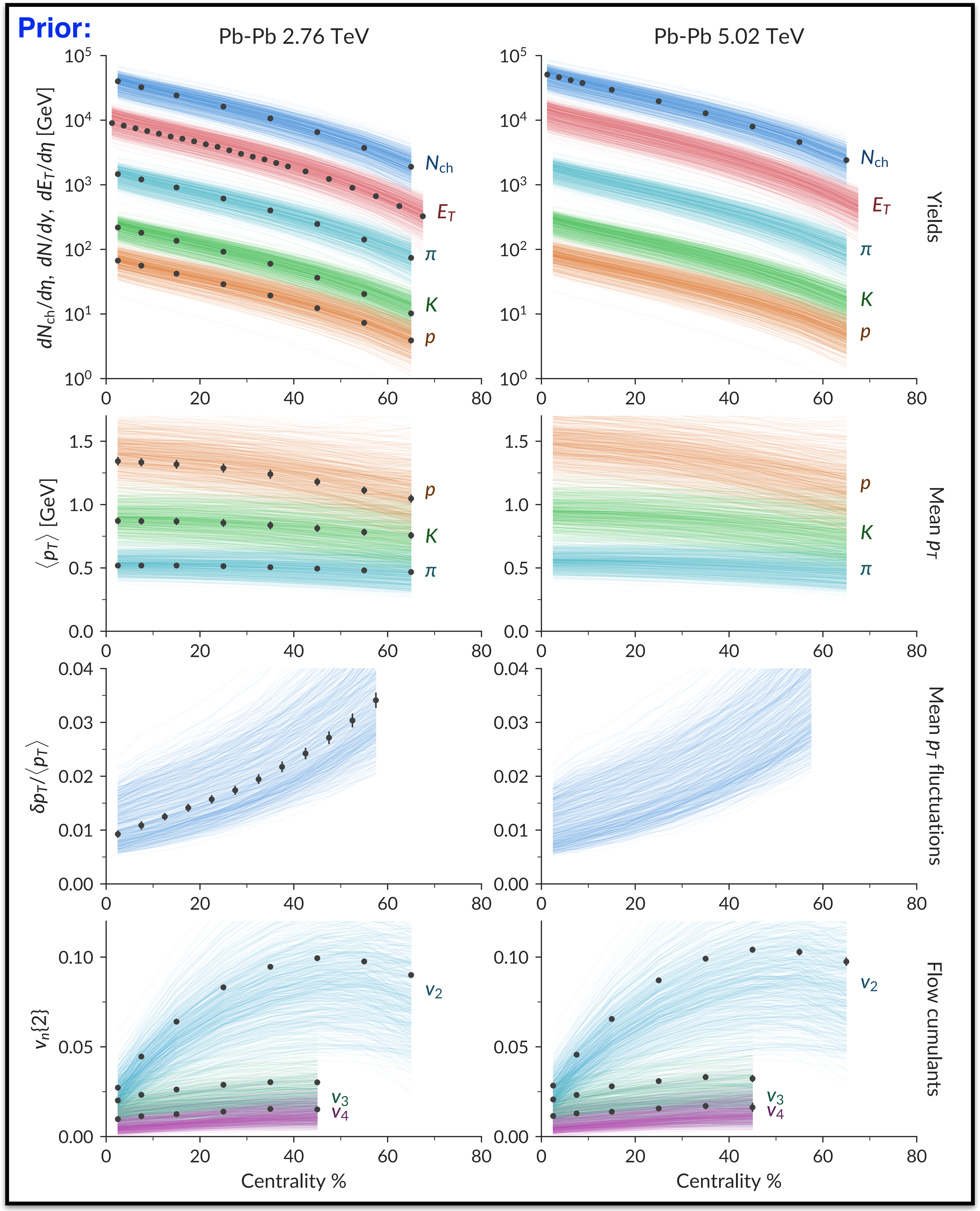
Prior vs. Posterior

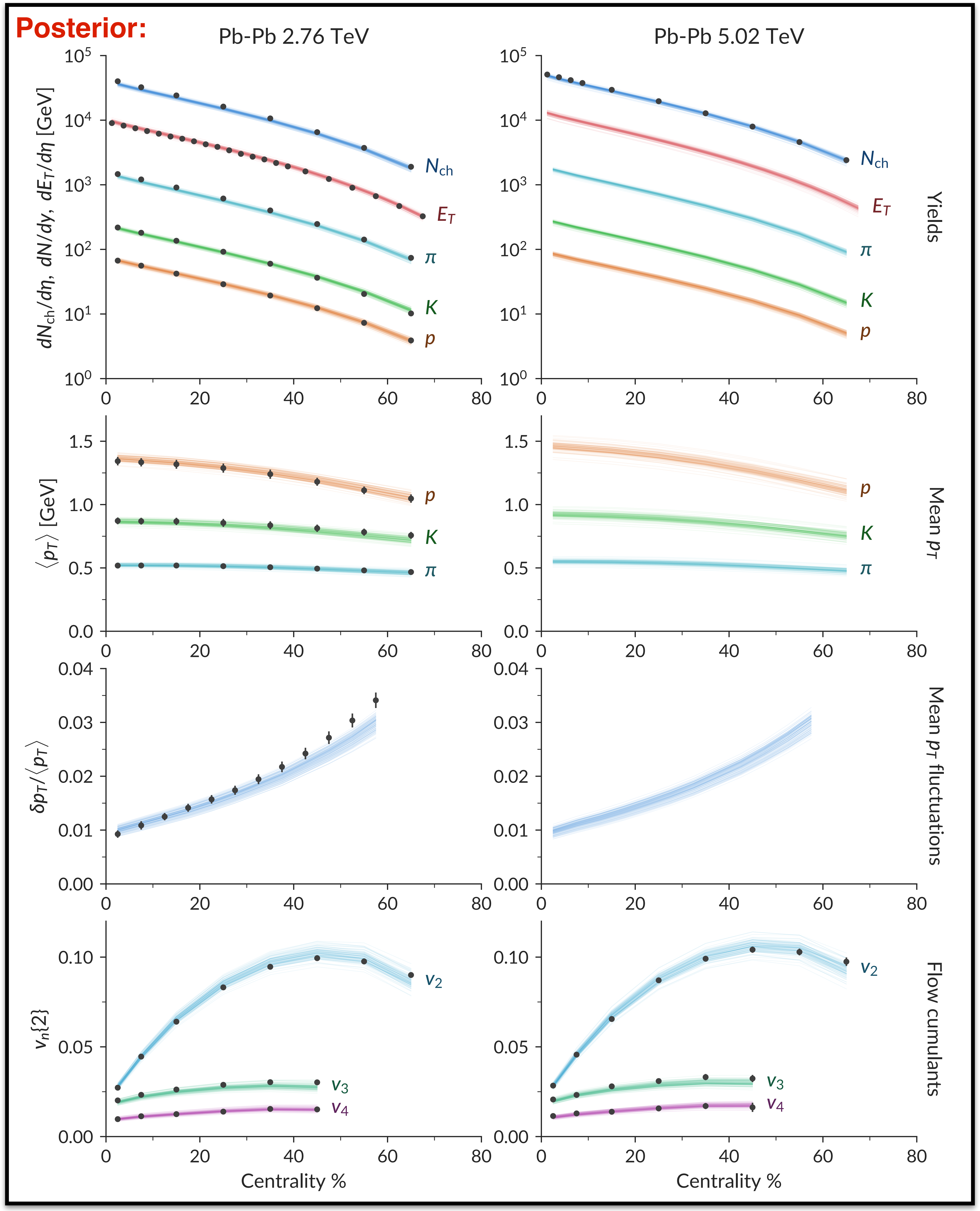
Prior:

- each line represents model evaluation at on point in latin hypercube
- covers full range of observable space

Posterior:

- each line represents model evaluation at points drawn from posterior probability distribution of parameters
- excellent agreement with data!





Methodology

Model Parameters - System Properties
- initial state
- temperature-dependent viscosities
- hydro to micro switching temperature

calculate events on Latin hypercube

Physics Model:
- Trento
- iEbE-VISHNU

Experimental Data
- ALICE flow & spectra

Gaussian Process Emulator
- non-parametric interpolation
- fast surrogate to full Physics Model

MCMC
(Markov-Chain Monte-Carlo)
- random walk through parameter space weighted by posterior probability

after many steps, MCMC equilibrates to

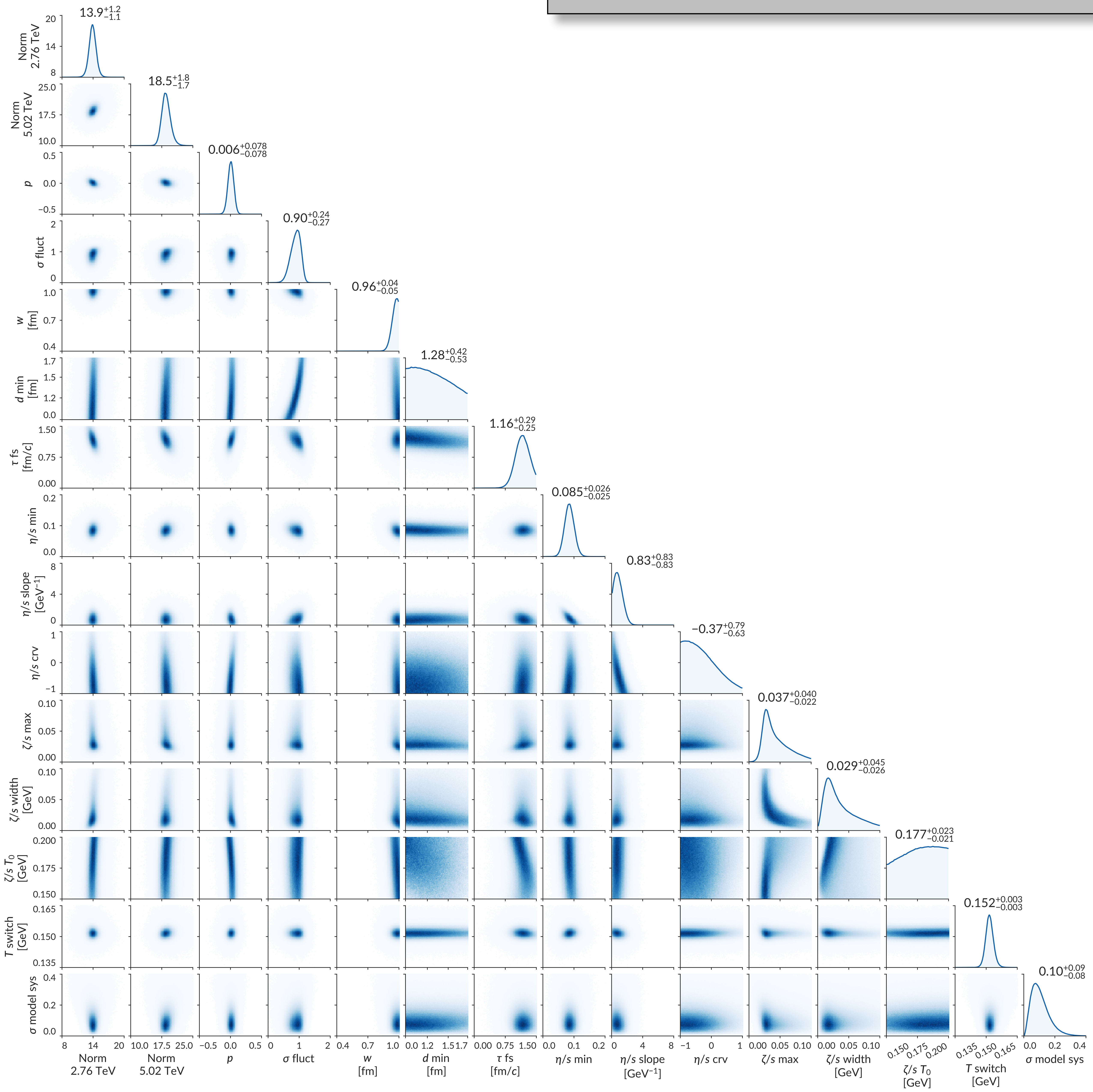
Bayes' Theorem

posterior $\propto$ likelihood $\times$ prior

- **prior**: initial knowledge of parameters
- **likelihood:** probability of observing exp. data, given proposed parameters

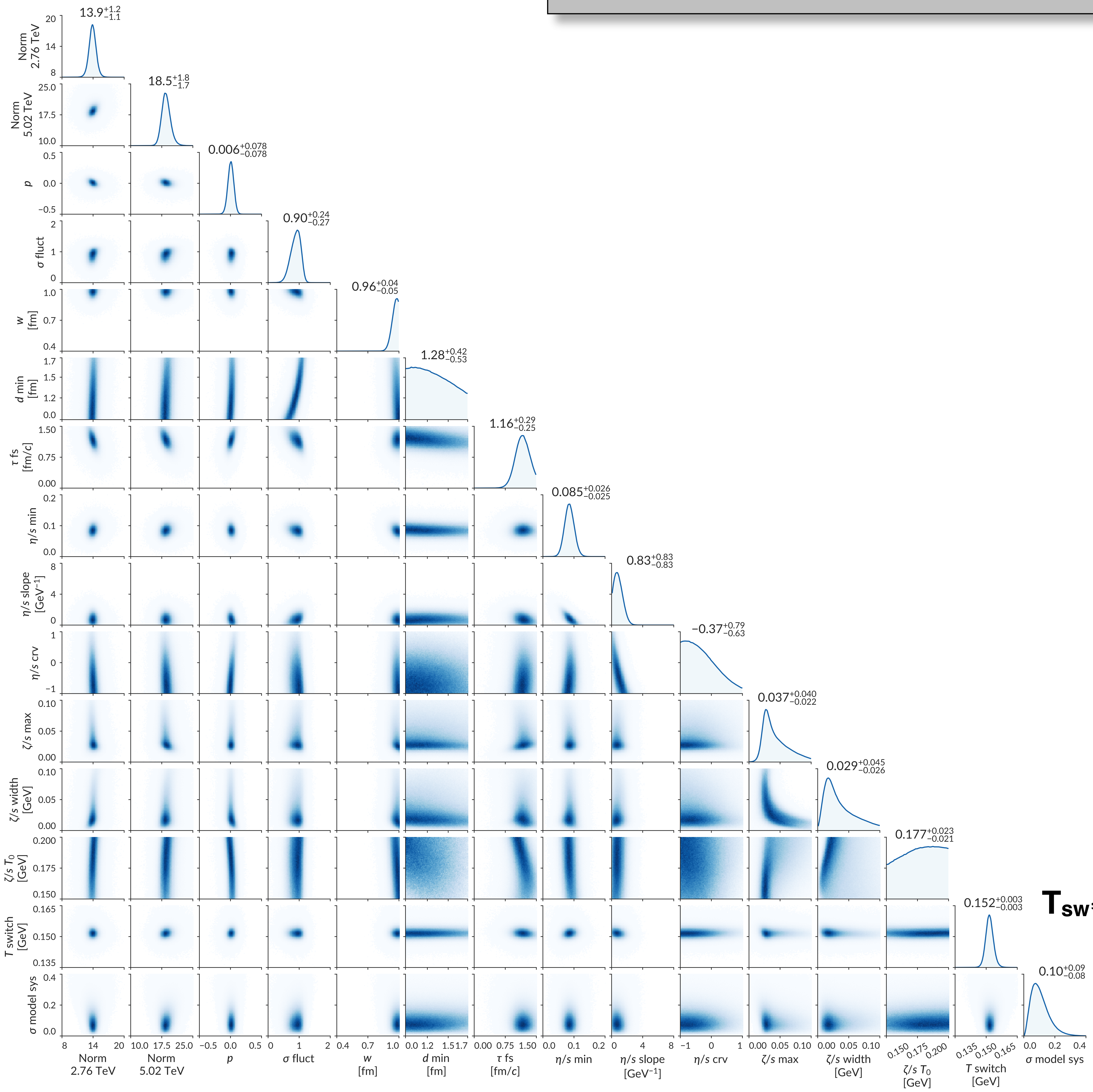
Posterior Distribution
- **diagonals**: probability distribution of each parameter, integrating out all others
- **off-diagonals**: pairwise distributions showing dependence between parameters

Calibrated Posterior Distribution



- **diagonals**: probability distribution of each parameter, integrating out all others
- **off-diagonals**: pairwise distributions showing dependence between parameters

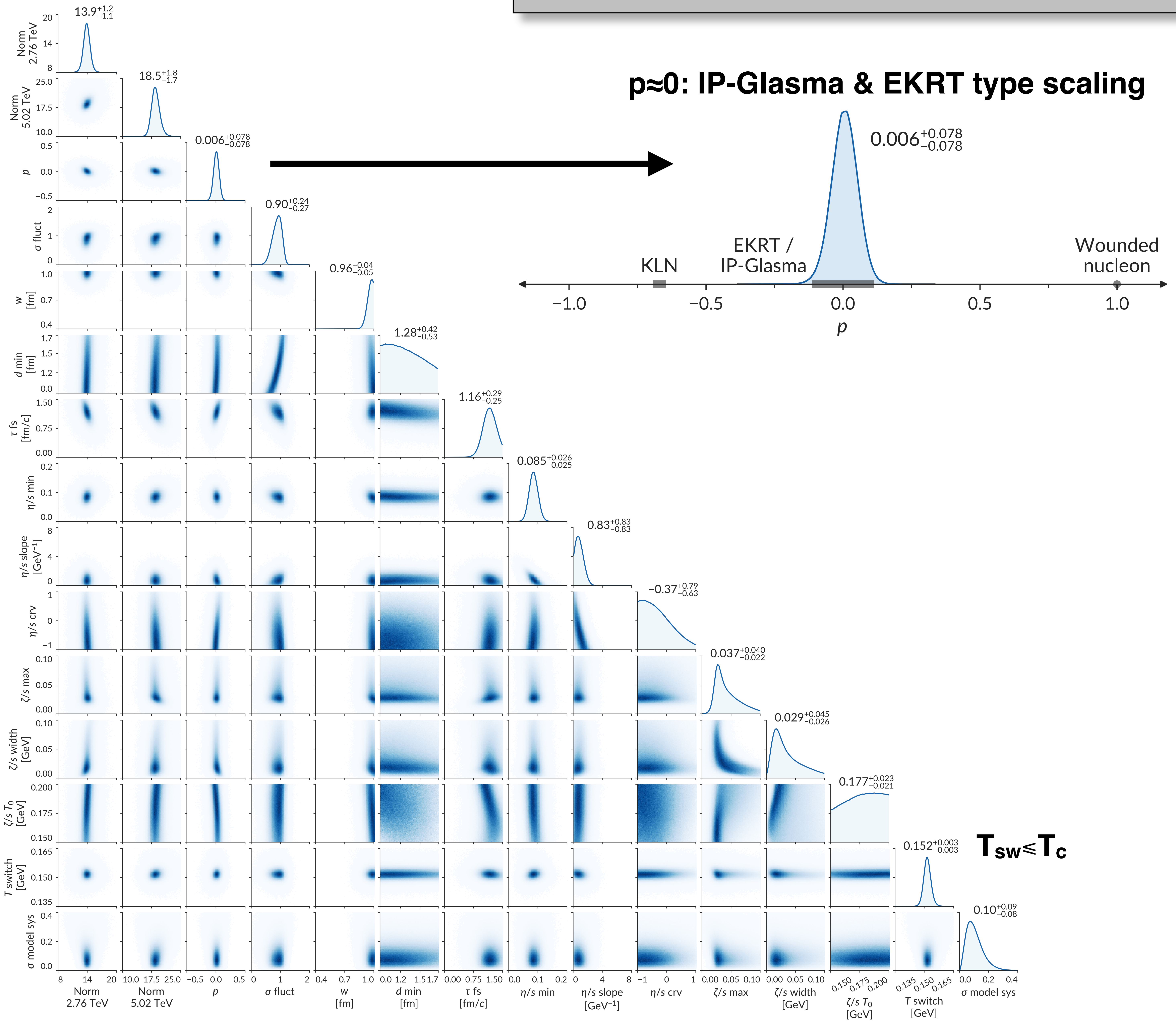
Calibrated Posterior Distribution



$T_{sw} \lesssim T_c$

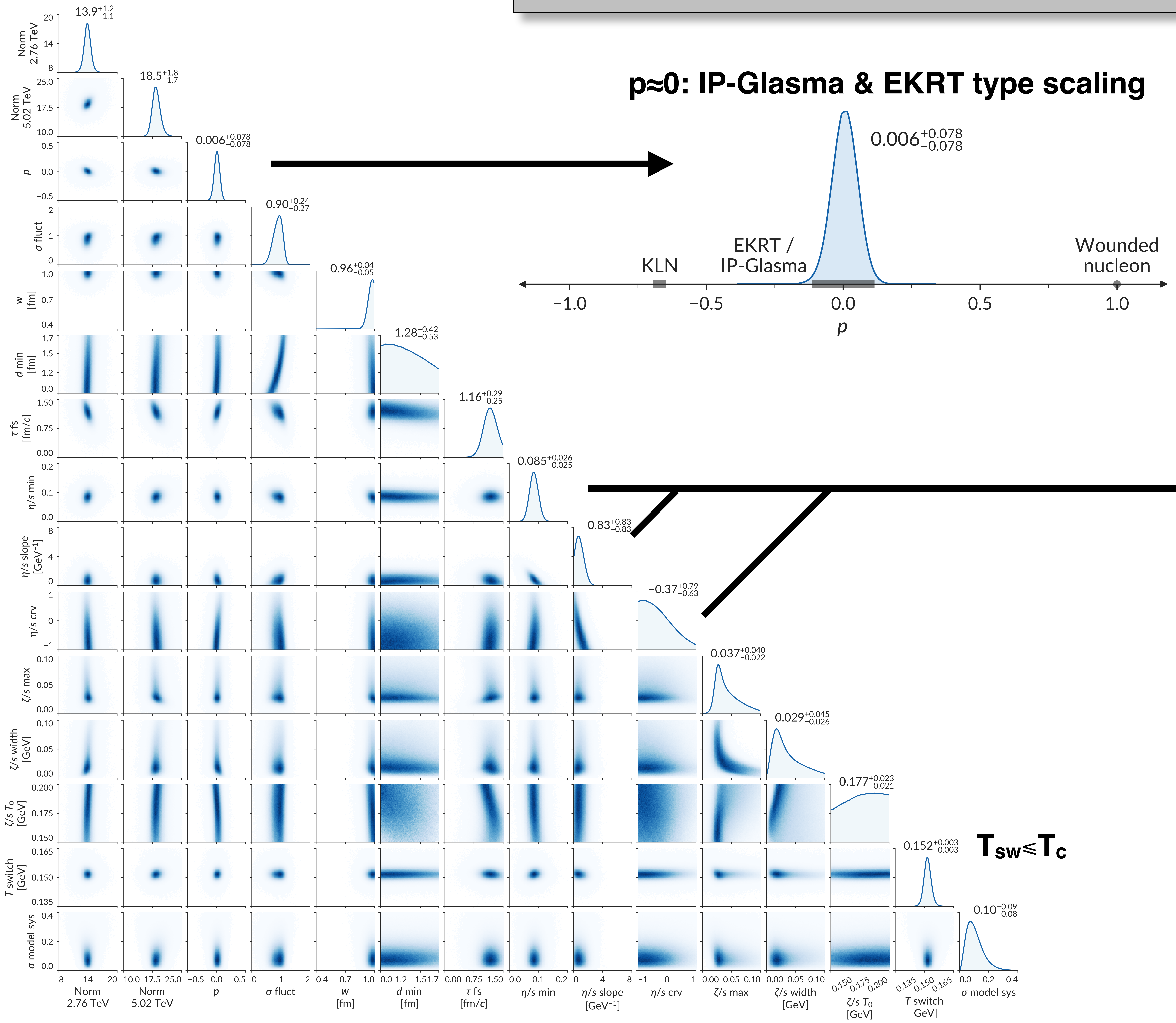
- **diagonals**: probability distribution of each parameter, integrating out all others
- **off-diagonals**: pairwise distributions showing dependence between parameters

Calibrated Posterior Distribution



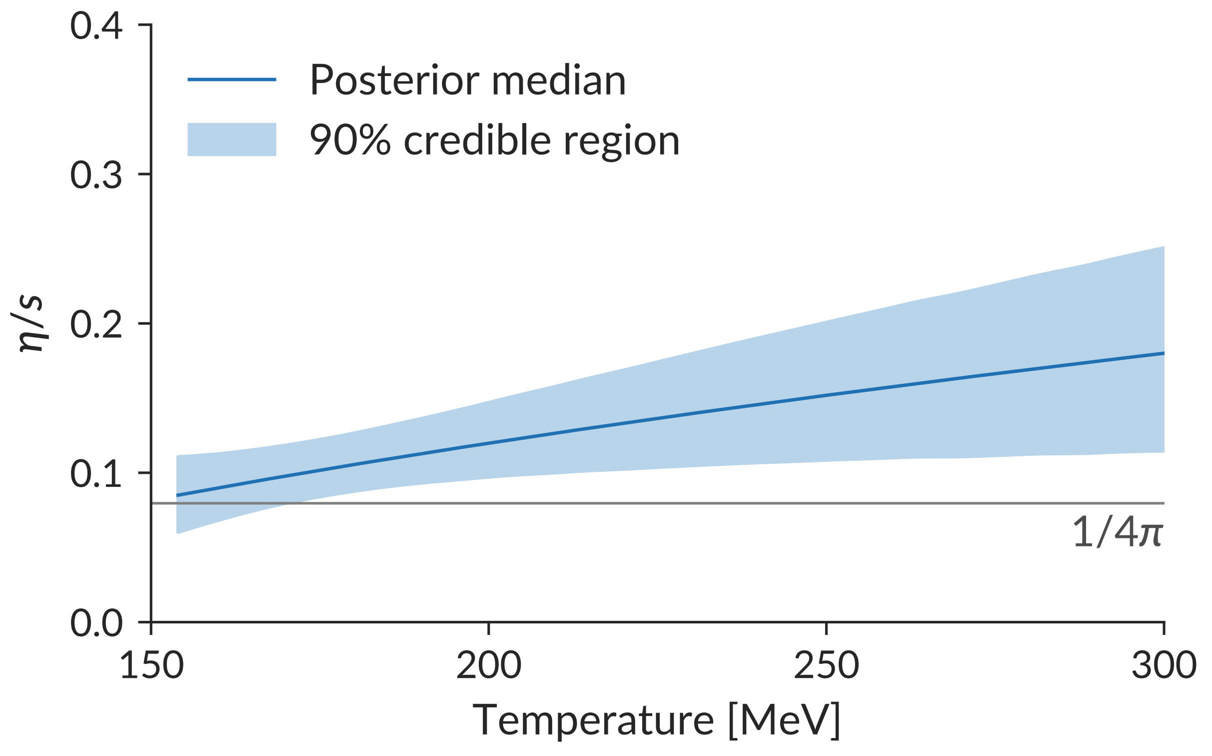
- **diagonals**: probability distribution of each parameter, integrating out all others
- **off-diagonals**: pairwise distributions showing dependence between parameters

Calibrated Posterior Distribution

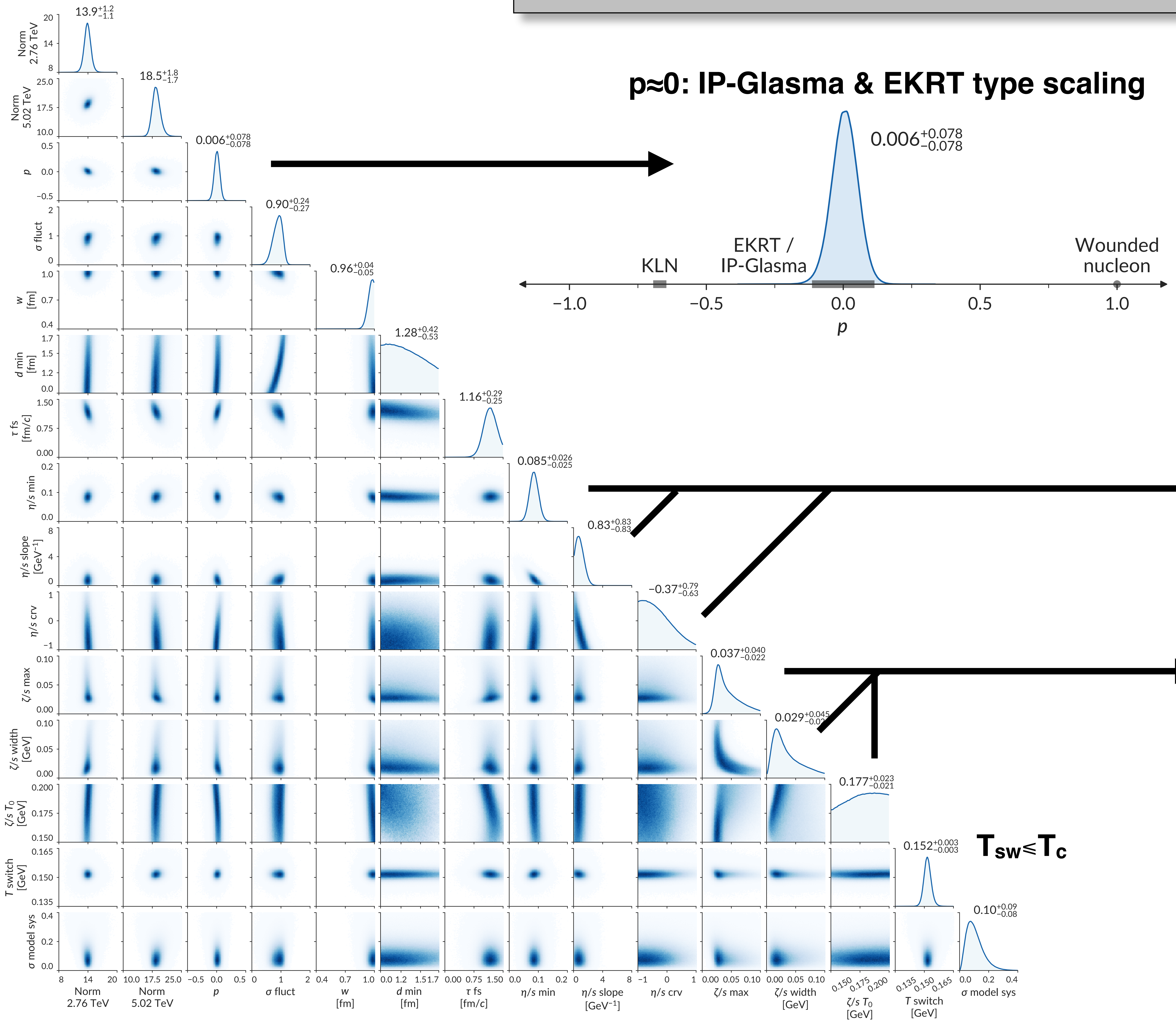


- **diagonals**: probability distribution of each parameter, integrating out all others
- **off-diagonals**: pairwise distributions showing dependence between parameters

temperature-dependent viscosities:

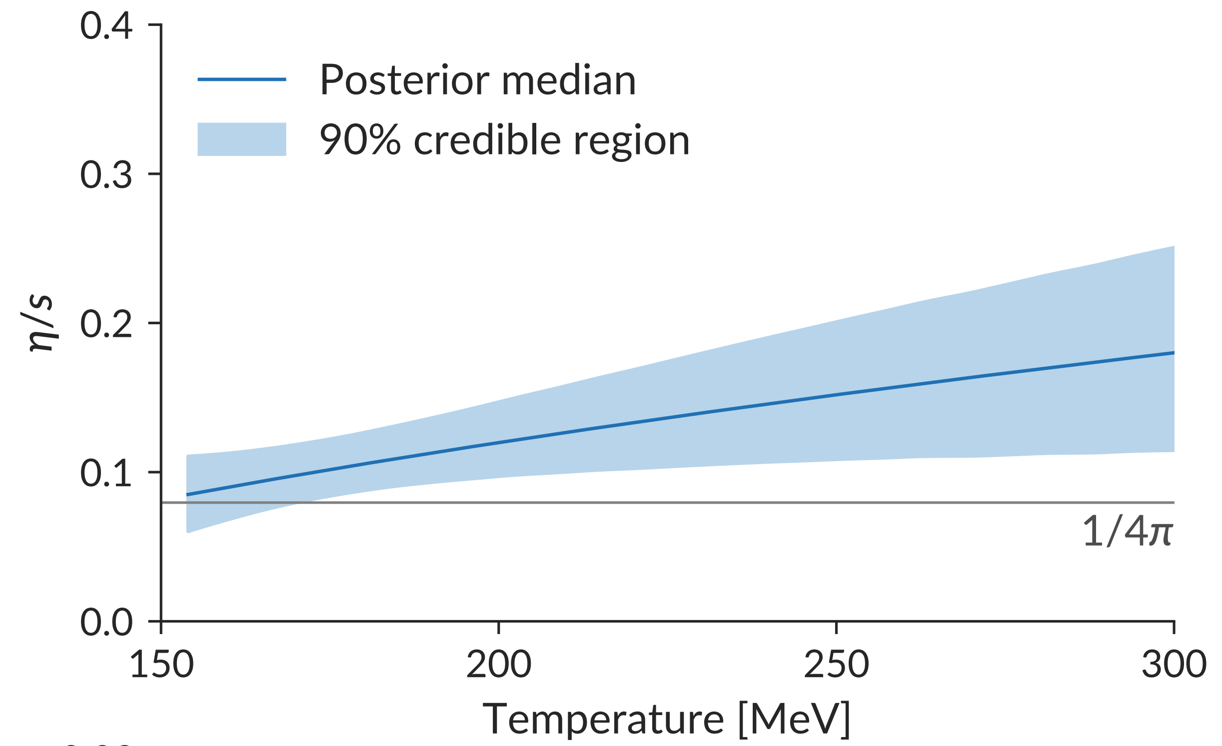


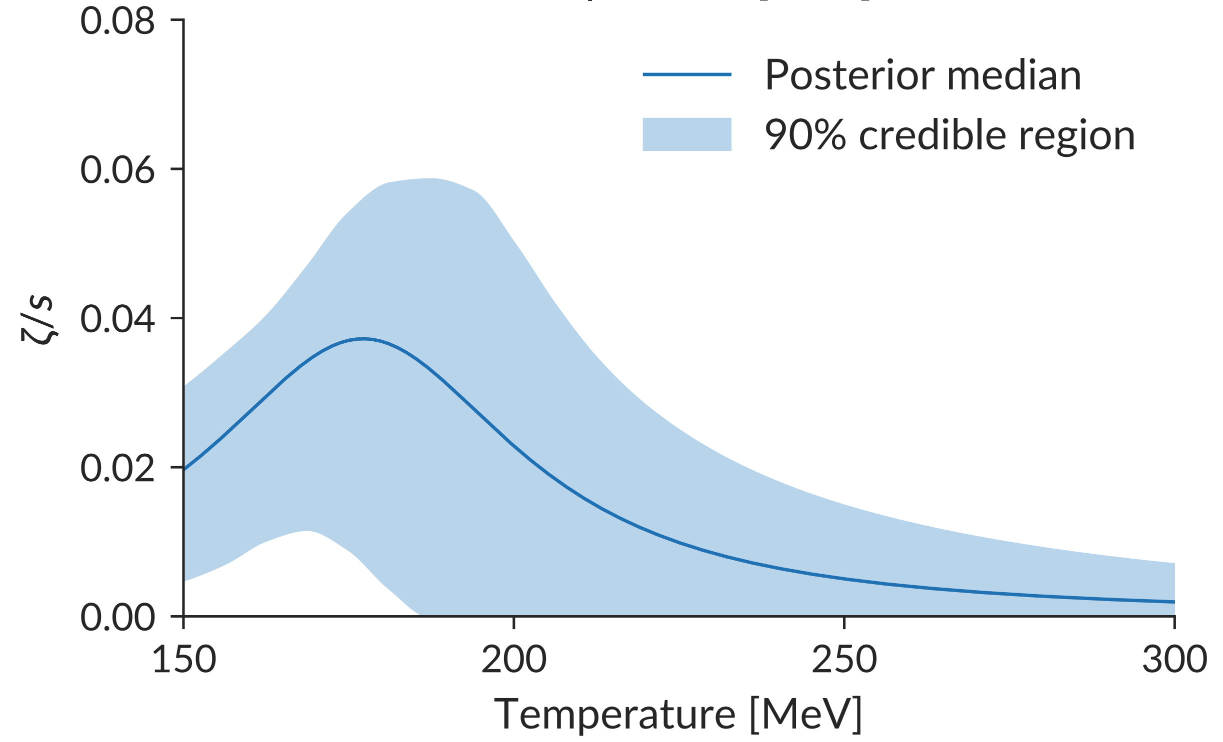
Calibrated Posterior Distribution



- **diagonals**: probability distribution of each parameter, integrating out all others
- **off-diagonals**: pairwise distributions showing dependence between parameters

temperature-dependent viscosities:

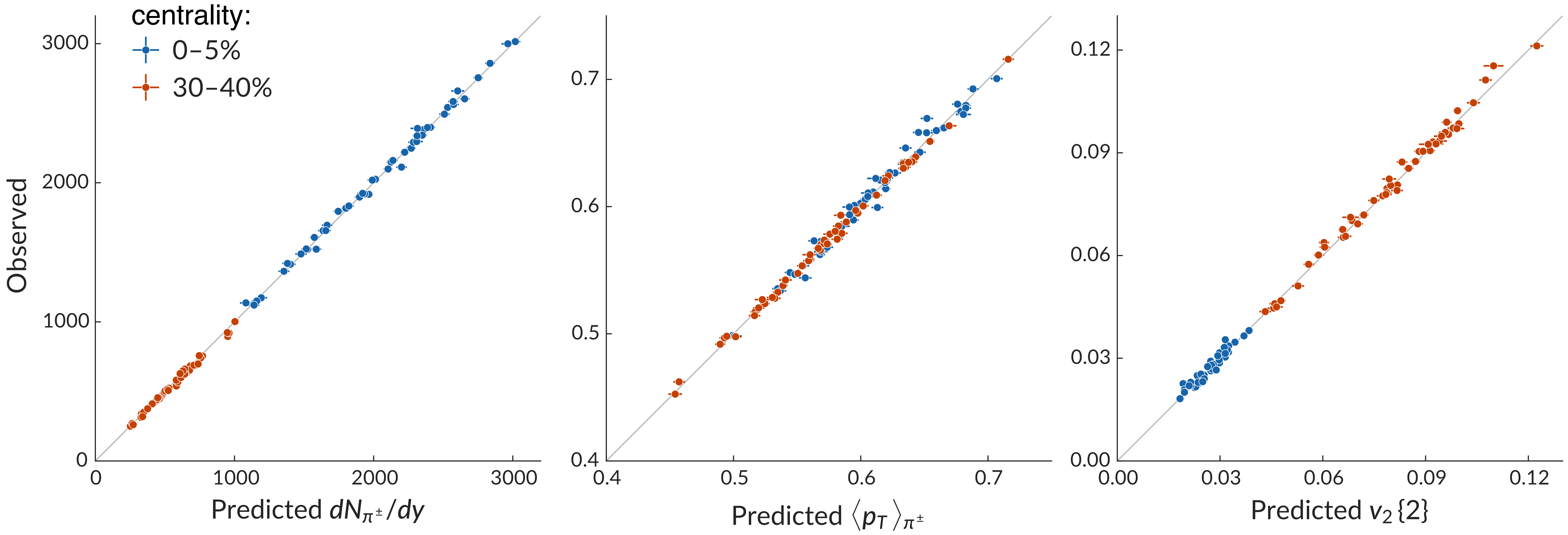




Precision Science
or
“Smoke & Mirrors”?

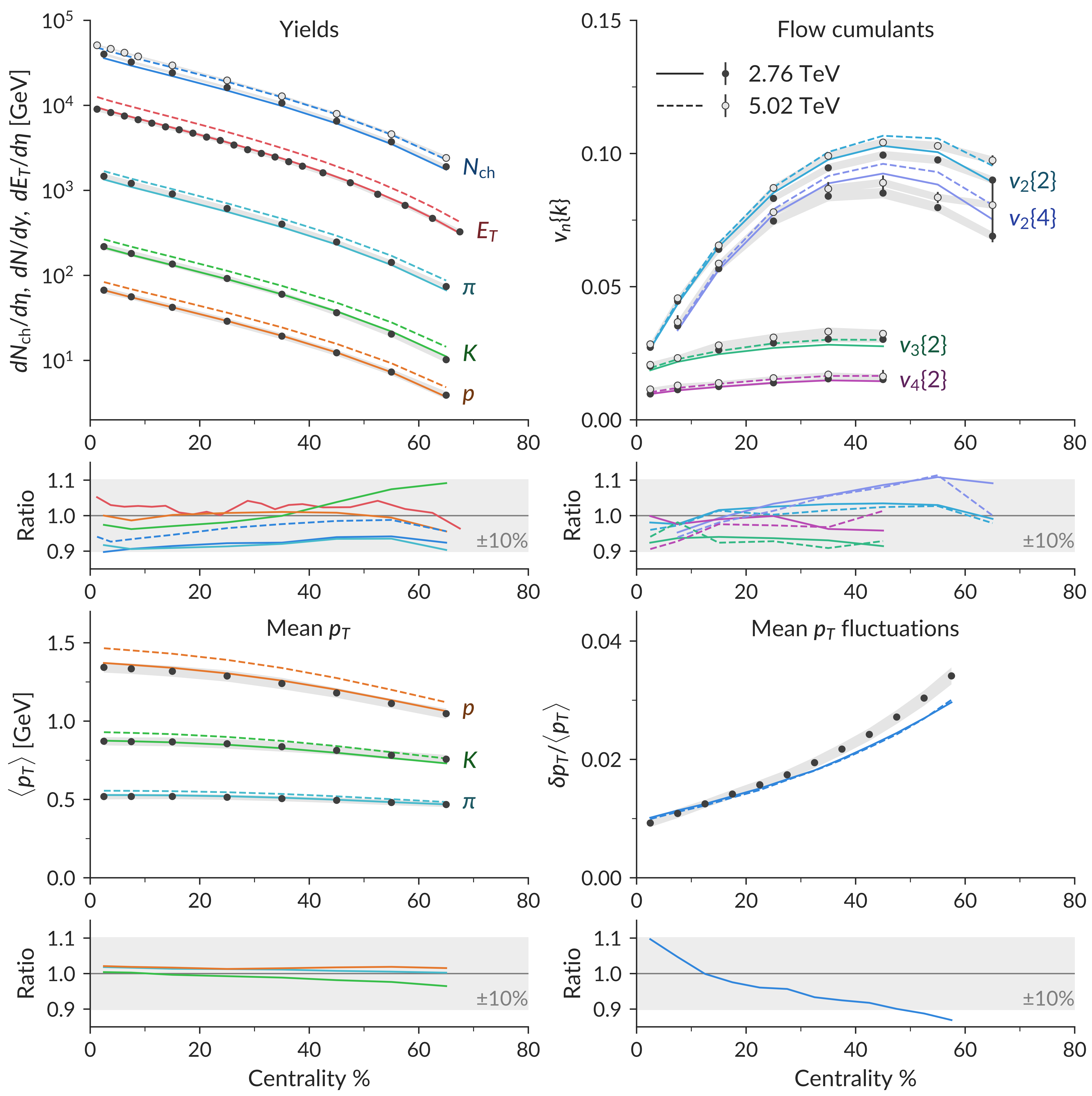
Validation

- generate a separate Latin hypercube validation design with 50 points
- evaluate the full physics model at each validation point
- compare physics model output to that of the previously conditioned GP emulators:



- note that since GPEs are stochastic functions, only ~68% of predictions need to fall within 1 standard deviation

Verification: Explicit Model Calculation



- explicit physics model calculations (no emulator) with parameter values set to the maximum of the posterior probability distributions yield excellent agreement with data!

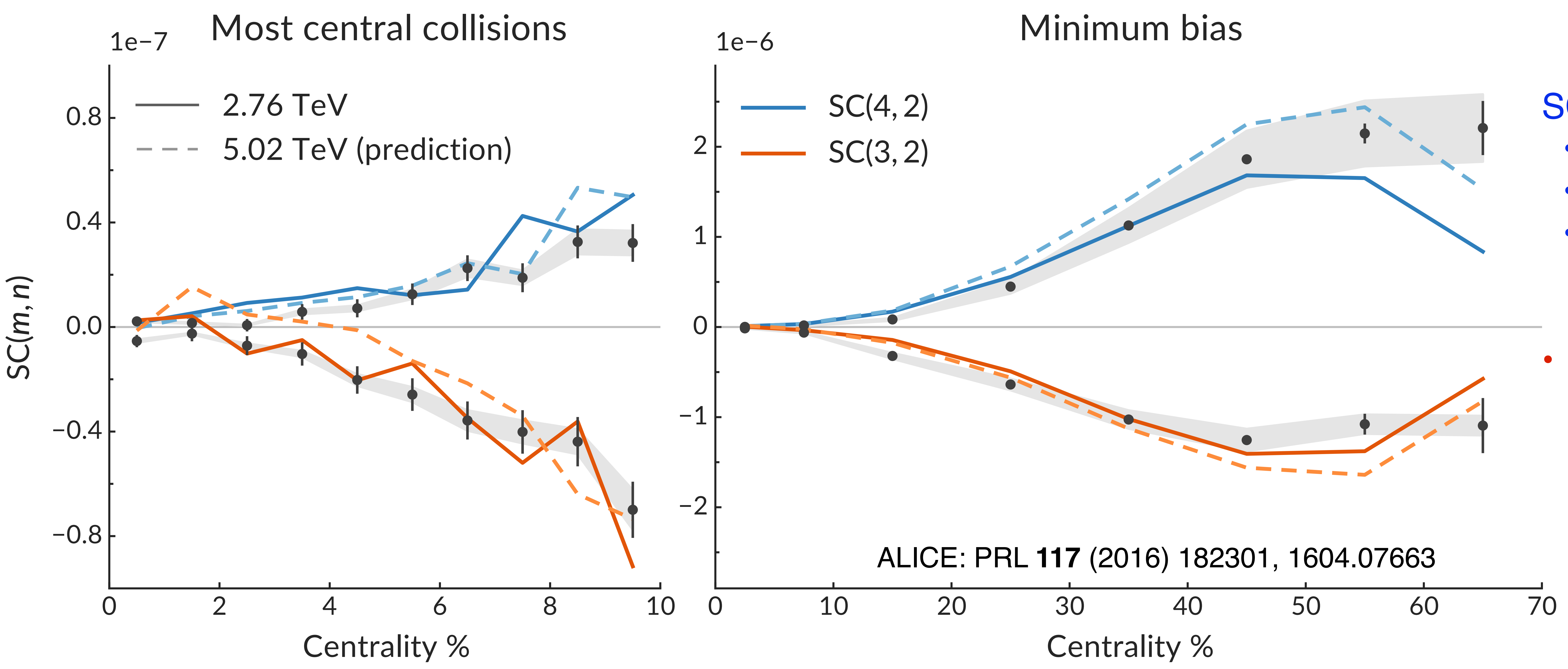
- description of data to within ±10% accuracy

Non-Calibrated Observables

The robustness and quality of the Physics Model can be tested by making predictions on observables not used during calibration using highest likelihood parameter values.

Example: correlations between event-by-event fluctuations of flow harmonics

$SC(m,n) = \langle v^2_m v^2_n \rangle - \langle v^2_m \rangle \langle v^2_n \rangle$



SC(m,n) are sensitive to:

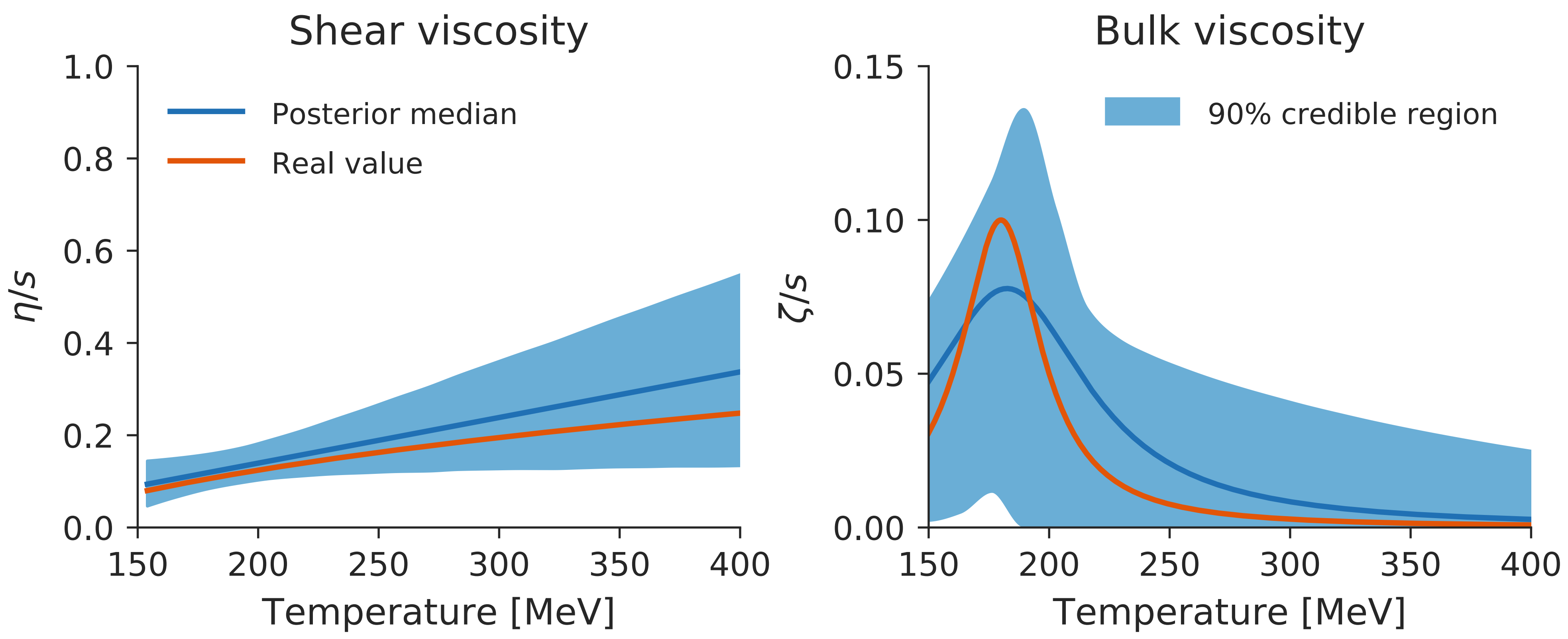
- initial conditions
- evolution model
- QGP transport coefficients

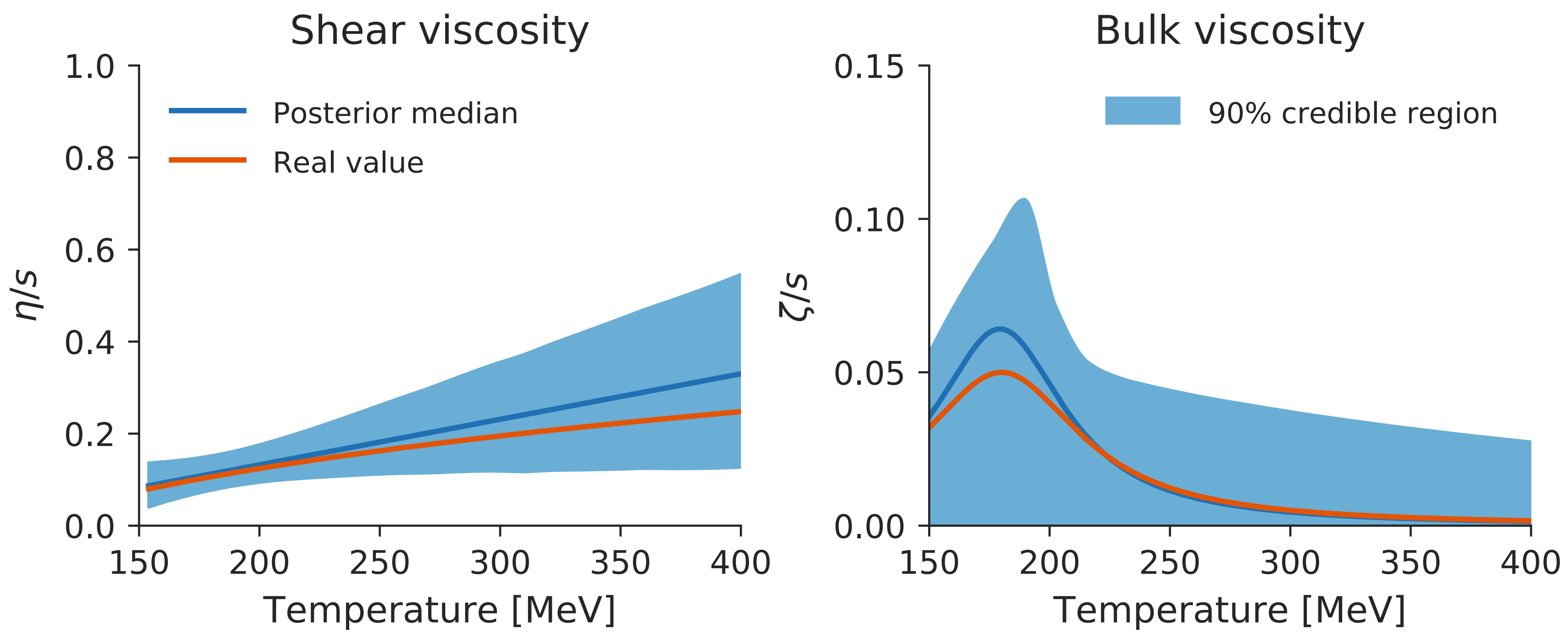
- excellent agreement of model prediction to data!

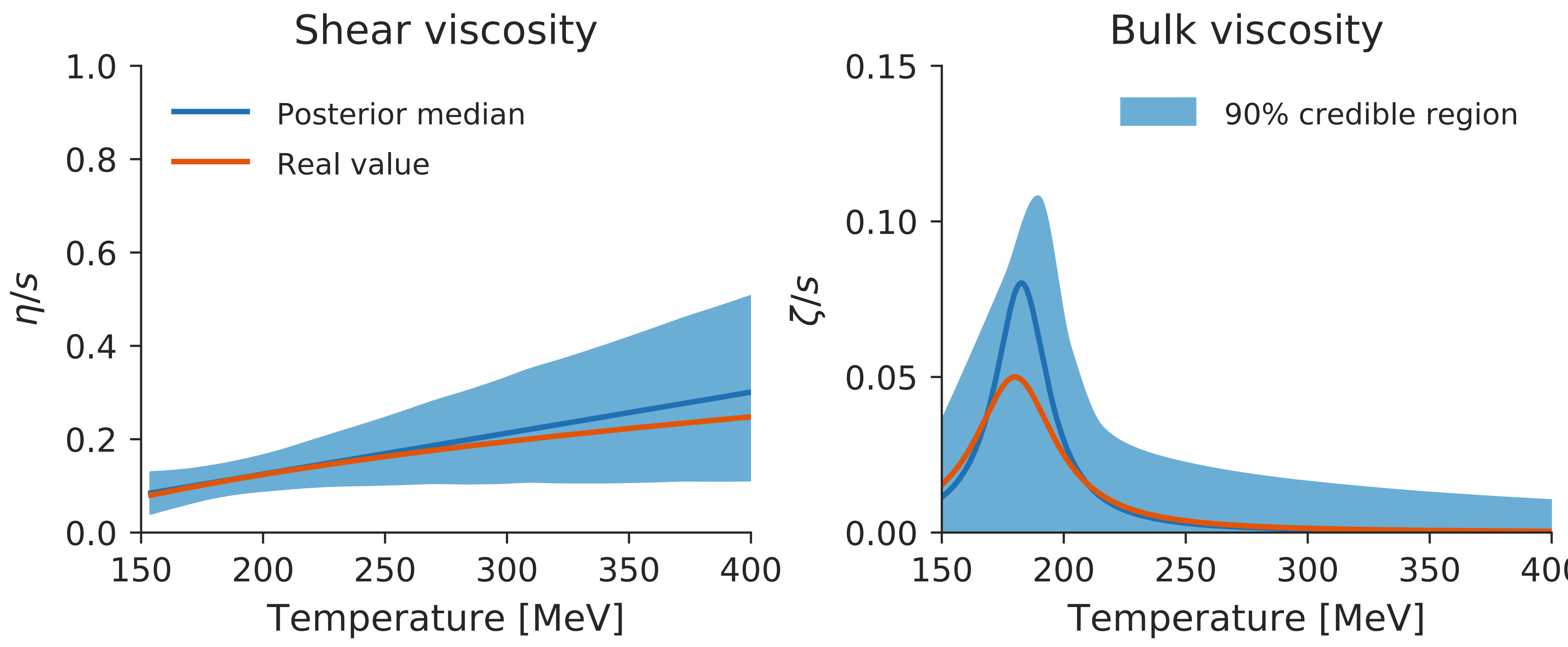
Closure Test

Need to verify that analysis can recover “true” values for the parameters: run physics model with chosen set of parameters, generate “fake data” from model output and then conduct analysis on that fake data to test if the input parameters can be recovered!

- both, smooth functions as well as peaked functions, can be reproduced well within the 90% CR
- note: due to reduction of information when going from model output to observables & model/GP uncertainties one should not expect a one-to-one reconstruction
- bulk analysis is mostly sensitive to area under bulk peak, not peak position, height & width independently







Next steps:

- scaling to 3D Hydro: multi-fidelity GPs

Y. Ji, S. Mak, D. Soeder, J.-F. Paquet & S.A. Bass: arXiv:2108.00306

Need for Multi-Fidelity Emulation

Problem:

- current setup uses state-of-the-art (i.e. high-fidelity) physics model
- computationally expensive, despite restriction to physics at mid-rapidity to save cost
- Inclusion of forward/backward physics requires 3D viscous RFD, raises cpu cost by a factor 100

Need for Multi-Fidelity Emulation

Problem:

- current setup uses state-of-the-art (i.e. high-fidelity) physics model
- computationally expensive, despite restriction to physics at mid-rapidity to save cost
- Inclusion of forward/backward physics requires 3D viscous RFD, raises cpu cost by a factor 100

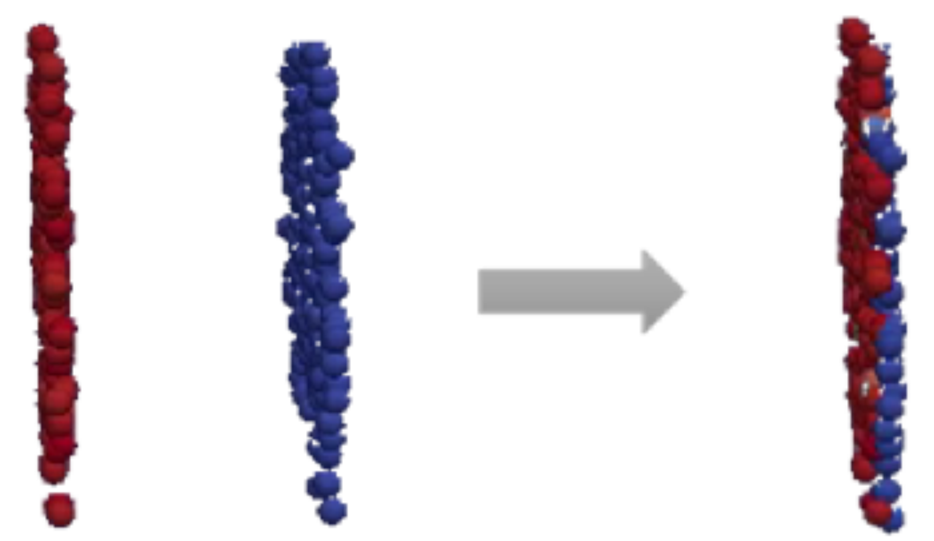
Idea:

- leverage lower-fidelity simulations (lower accuracy, but quicker to simulate) as data to train the high-fidelity emulator:

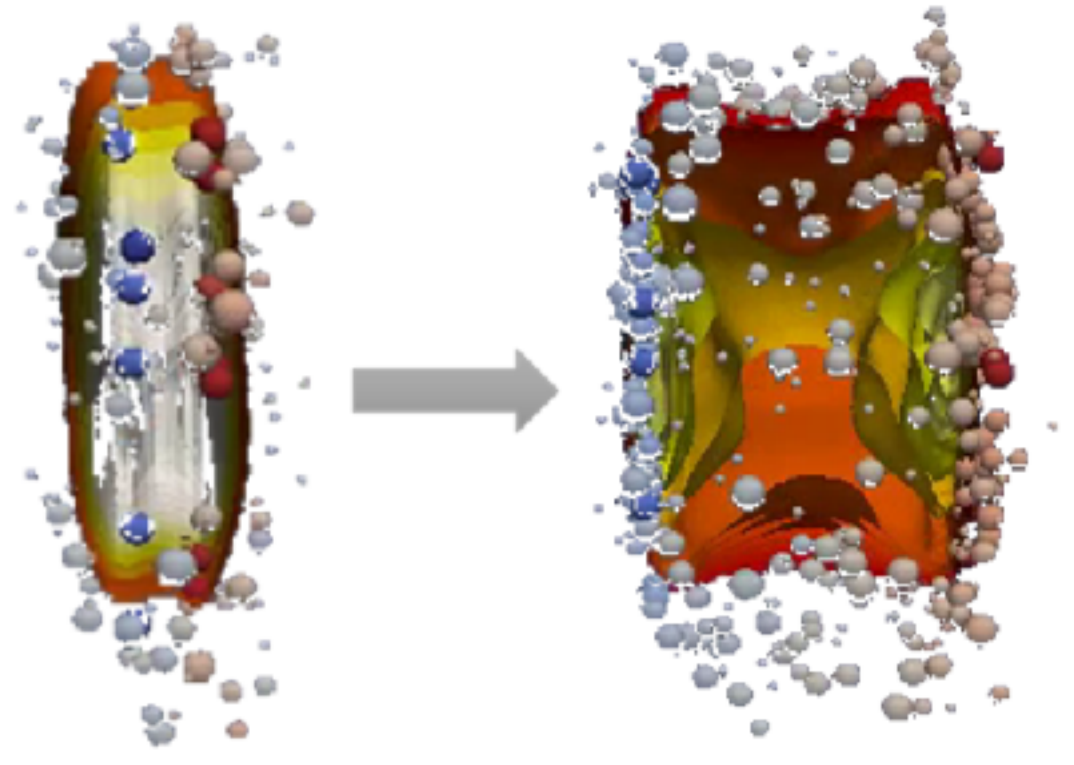
low-fidelity	10^0 cpu hours/run
medium-fidelity	10^2 cpu hours/run
high-fidelity	10^3 cpu hours/run

These types of approaches are referred to as **multi-fidelity emulation** or **data fusion**

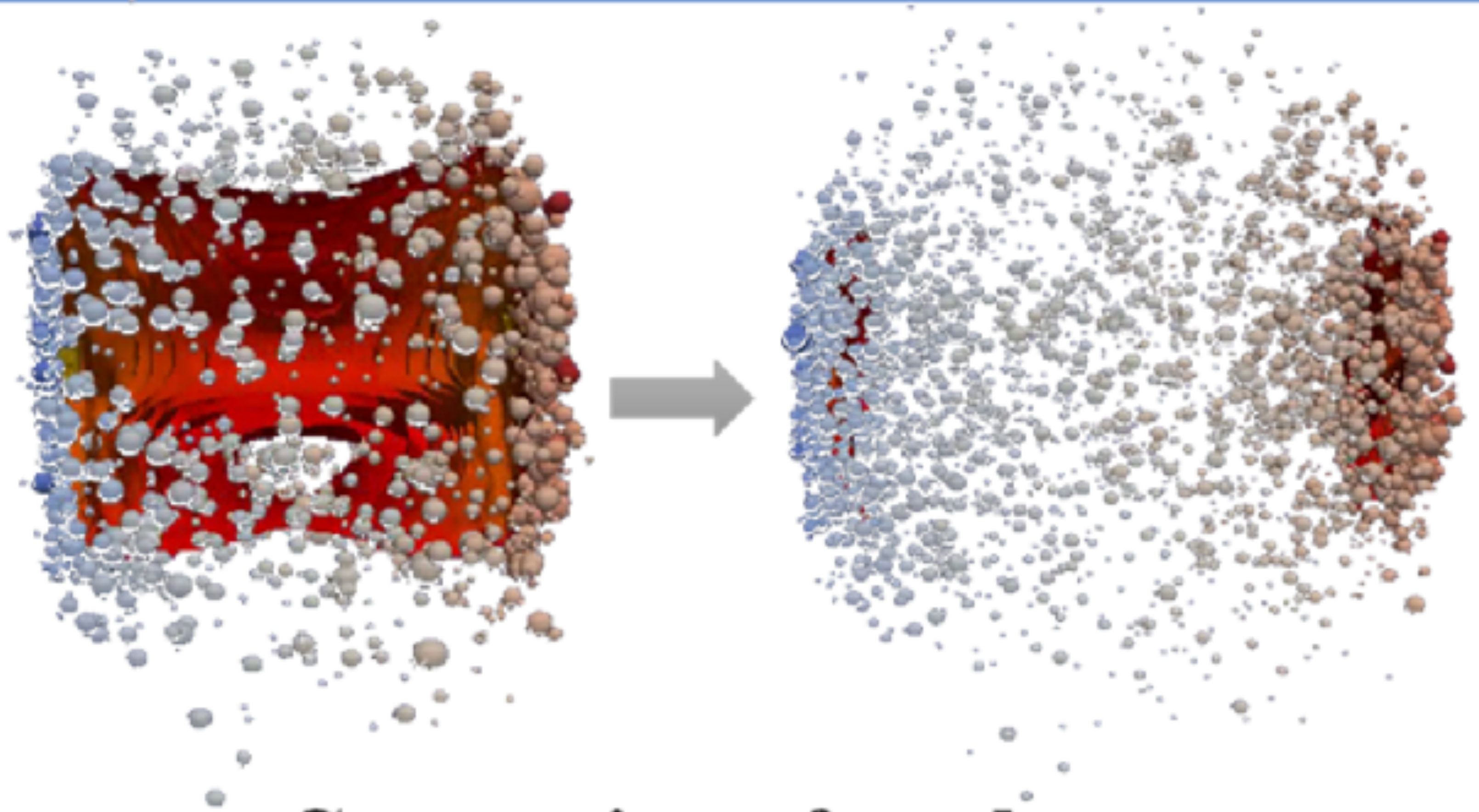
Construction of Physics Models with Varying Fidelity



Initial energy disposition of colliding nuclei



Hydrodynamic evolution of plasma



Conversion of nuclear fluid into particles

Fidelity ↓

Pre-hydrodynamics
(A_1) Trento 3D
(A_2) Gluon-saturation-based models

Fidelity ↓

Hydrodynamics
(B_1) Linearized conformal hydro
(B_2) 1+1D ideal QCD hydro
(B_3) 3+1D ideal QCD hydro
(B_4) 3+1D viscous QCD hydro

Fidelity ↓

Post-hydrodynamics
(C_1) None
(C_2) Thermal hadrons (Cooper-Frye)
(C_3) Hadrons with decays
(C_4) Hadrons with interactions and decays

Construct models of varying fidelity by combining modules of varying fidelity for initial conditions, hydro evolution and hadronic final state interactions:

low fidelity: A_1 - B_1 - C_1
medium fidelity: A_1 - B_3 - C_3
high fidelity: A_2 - B_4 - C_4

Kennedy-O'Hagan Model (KO)

- assume T levels of simulation codes, from lowest fidelity η_1 to highest fidelity η_T:

$$\eta_1 \longrightarrow \eta_2 \longrightarrow \eta_3 \cdots\cdots\longrightarrow \eta_T$$

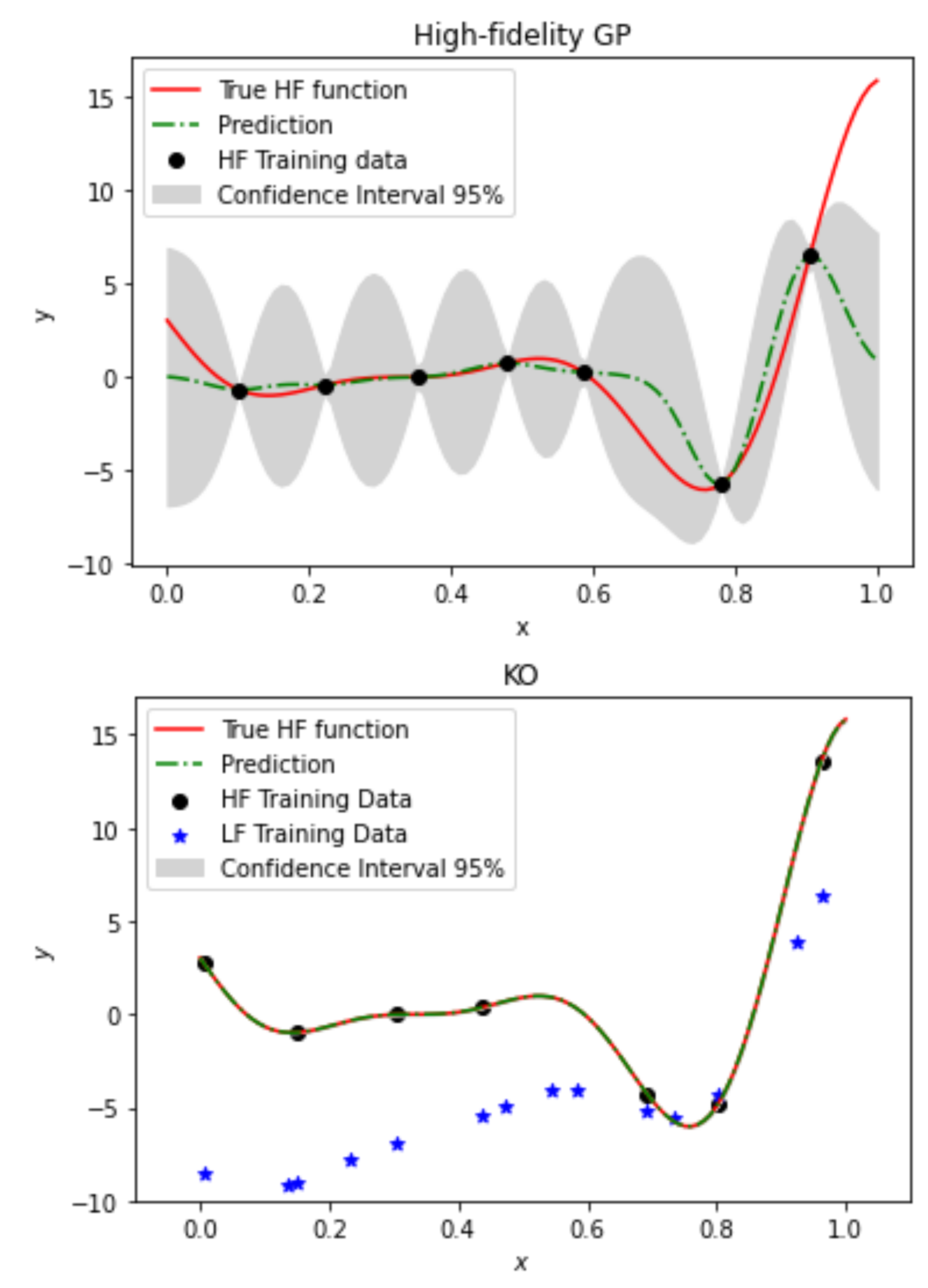
- assume first-order autoregressive prior model:

$$\begin{cases} \eta_t(\theta) = \rho_t \eta_{t-1}(\theta) + \delta_t(\theta) \qquad t = 2, 3, \cdots, T \\ \eta_{t-1}(\theta) \perp \delta_t(\theta) \end{cases}$$

where $\eta_1(\cdot) \sim GP(\mu_1(\cdot), k_1(\cdot,\cdot))$ and $\delta_t(\cdot) \sim GP(\mu_t(\cdot), k_t(\cdot,\cdot))$

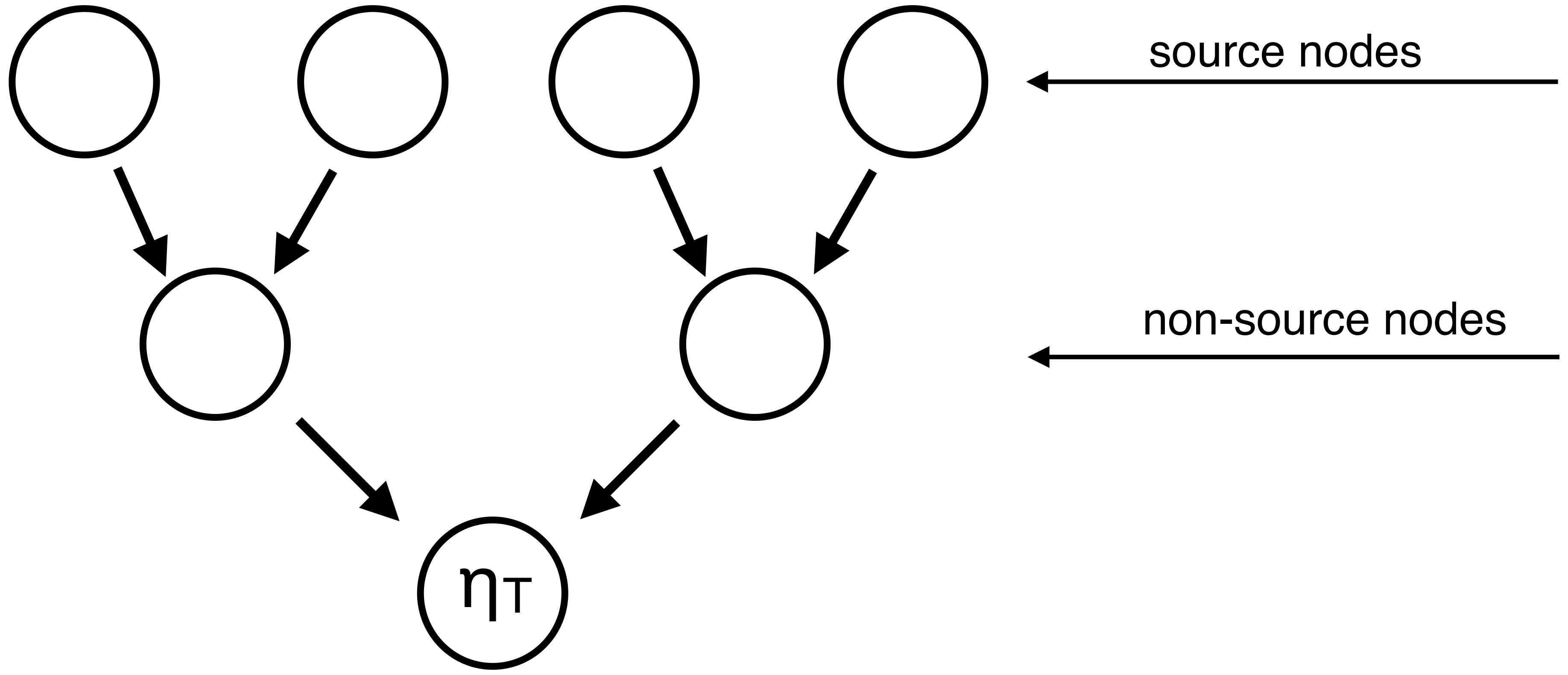
- $\delta_t(\theta)$: discrepancy function between η_{t-1} and η_t (beyond scaling)
- it can be shown that the highest-fidelity $\eta_T(\cdot)$ is a GP with closed form posterior mean and variance
- nested designs allow for efficient prediction!

Marc C. Kennedy and Anthony O'Hagan: J. R. Statist. Soc. B (2001) 63, Part 3, pp. 425-464



Limitation: the KO model assumes a clear hierarchy in the fidelity of the physics models. What if that hierarchy cannot be unambiguously established?

Graphical Multi-Fidelity Gaussian Process Model (GMGP)

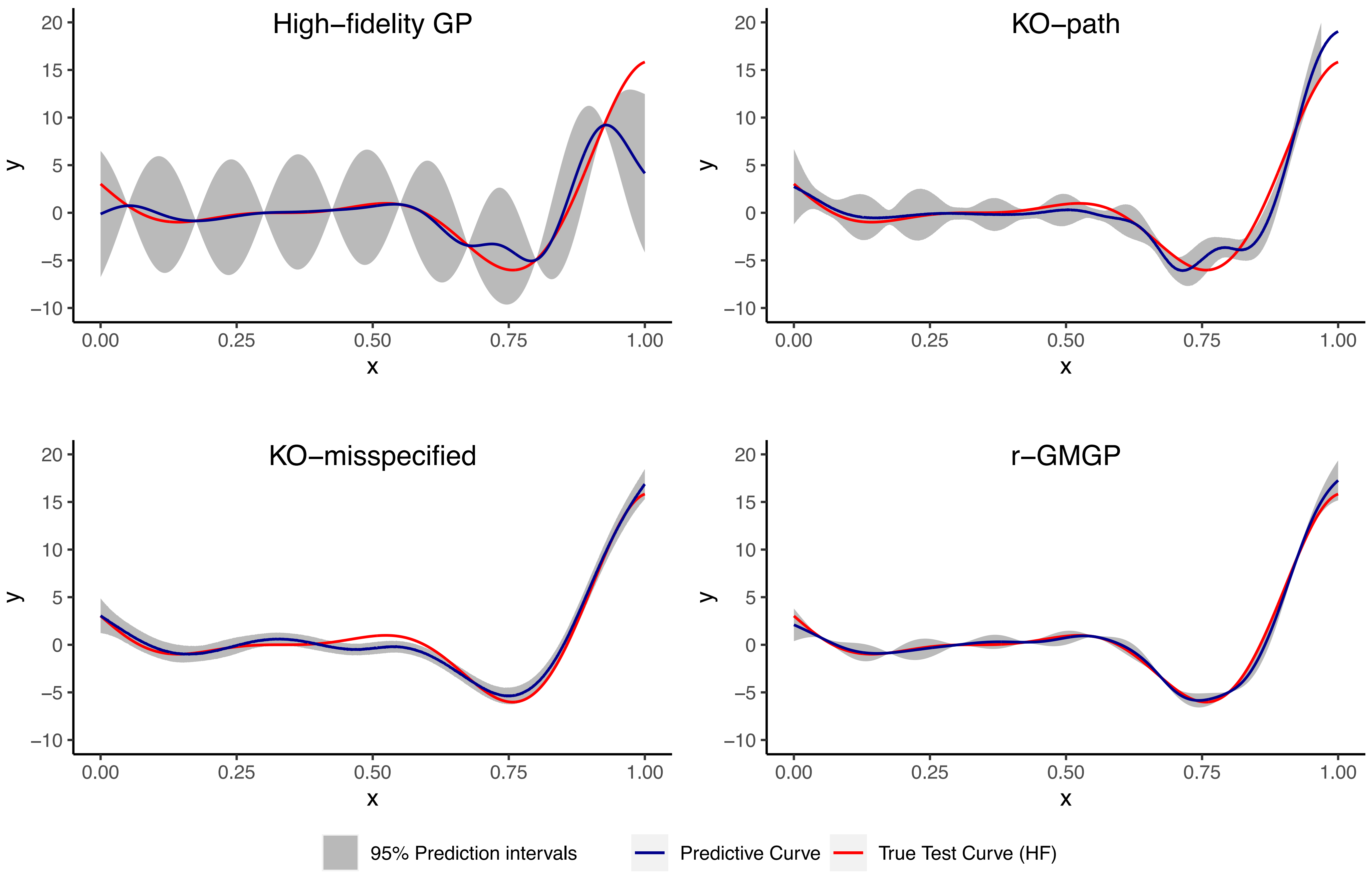


Idea: represent the underlying multi-fidelity simulation structure via a **directed acyclic graph** (DAG)

- given the DAG, one can set up a **Graphical Multi-Fidelity Process Model (GMGP)** via two autoregressive priors, one for the source nodes and one for the non-source nodes
    - as with the KO model, it can be shown that the highest-fidelity $\eta_T(\cdot)$ is a GP with closed form posterior mean and variance
    - the GMPG can handle fidelity hierarchies far more complex than the KO model

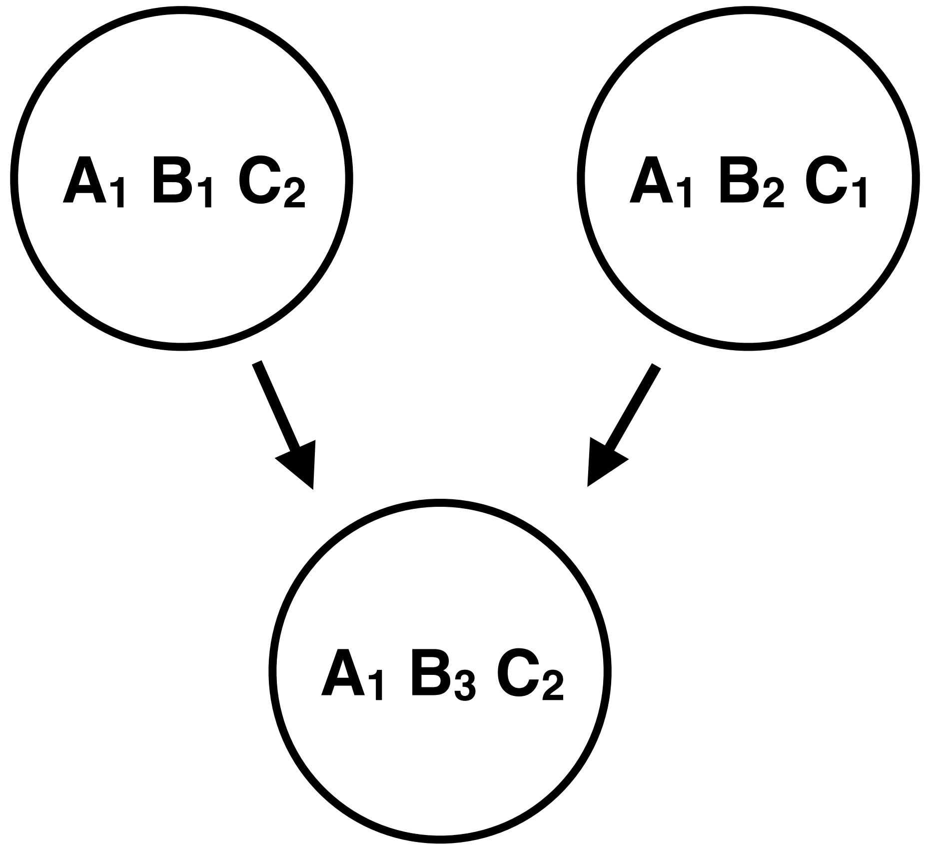
Yi Ji, Simon Mak, Derek Soeder, J-F Paquet, Steffen A. Bass: arXiv:2108.00306 [stat.ME]

- **complication:** the GMGP requires the inversion of a large N×N covariance matrix (N = total sample size over graph): very computationally expensive!
- **solution:** a recursive formulation of the GMGP model (r-GMGP)
    - the posterior predictive distribution for η_t can be calculated recursively over subgraphs, from leaf nodes to the root
    - computational savings: from $O(N^3)$ to $O(\,|V|\max(N_t)^3\,)$



Proof of Concept Application to Heavy-Ion Collisions

DAG with three different physics models:



Computer experiment design:

- 9 parameters
- Sliced Latin Hypercube Design (SLHD) with
  - 200 points each for the lower-fidelity models
  - 25 points for the high-fidelity model
- Predictive performance tested with a 75 point Latin Hypercube Design

Predictive Performance:

RMSE: Root-mean-squared-error
p-RMSE: probabilistic RMSE (accounts for bias and variance)

Model	RMSE	p-RMSE
High-fidelity GP	5.49	5.83
KO-path	3.46	3.80
KO-misspecified	3.95	4.45
d-GMGP	2.17	2.44

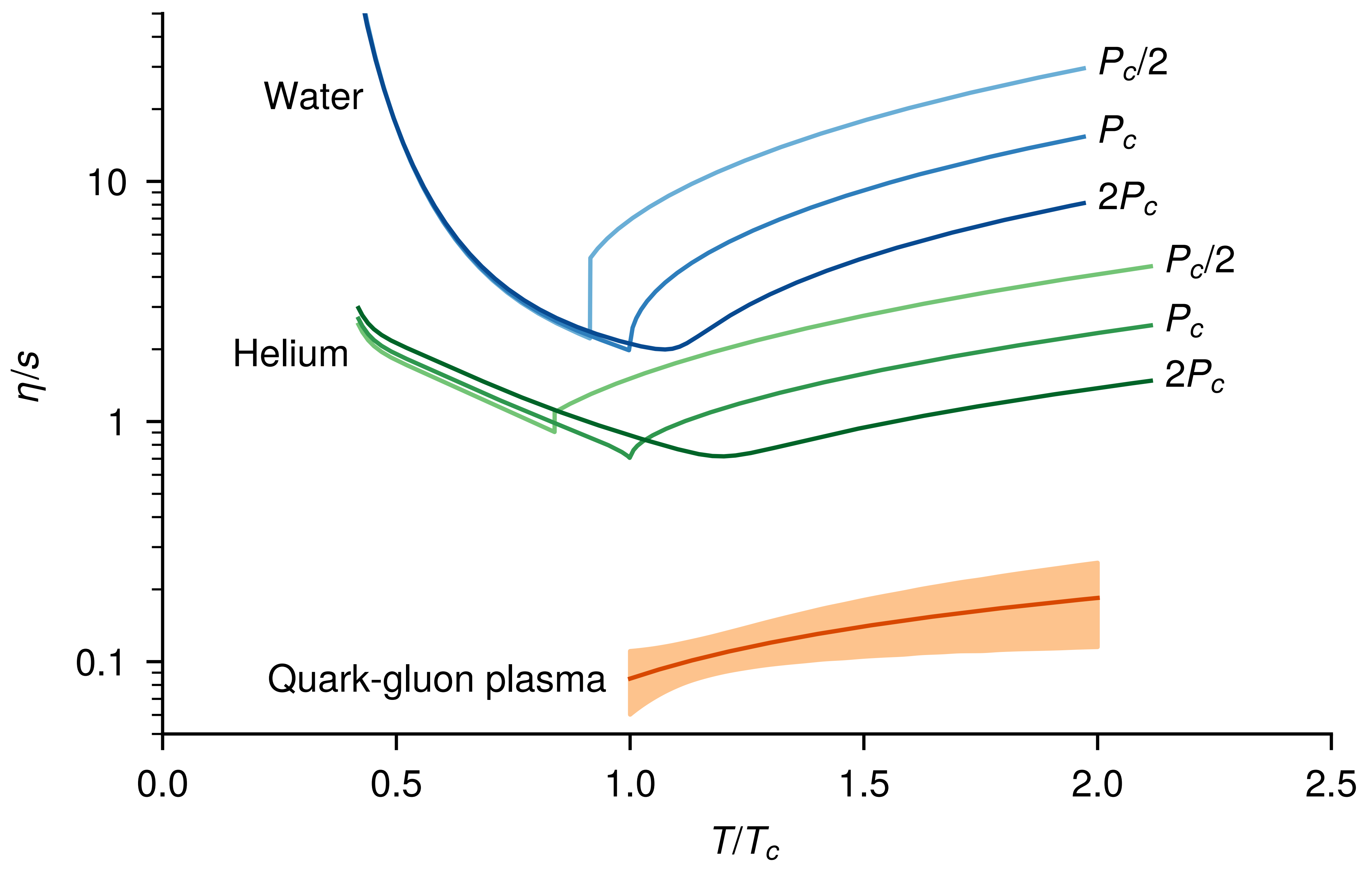
- GMGP model provides the best performance
- proper treatment of model hierarchy as “prior” plays important role in performance

Yi Ji, Simon Mak, Derek Soeder, J-F Paquet, Steffen A. Bass: arXiv:2108.00306 [stat.ME]

Summary & Outlook

Physics highlight: shear-viscosity of the QGP

- Apply Data Science and Bayesian Statistics for large scale model-to-data analysis
- most ideal fluid ever found in nature



Bernhard, Moreland & Bass, *Nature Physics* **15** (2019) 11, 1113-1117

Knowledge Extraction via Model-to-Data Calibration:

- **Bayesian inference** is crucial for calibration of sophisticated multi-parameter models to large data sets
- **Gaussian Process Emulators** make these calibrations feasible - cost savings of 10 orders of magnitude in cpu expenditure
- further refinement with more complex physics models requires application of advanced GP techniques to overcome cpu cost limitations: **multi-fidelity emulation & data fusion**
- **Graphical Multi-Fidelity Gaussian Process Model** as a new approach to deal with physics models of varying fidelity that don't exhibit a clear hierarchy

Collaborators:

Duke QCD Group:

- Jonah Bernhard (now Lowe's)
- J. Scott Moreland (now Google)
- Derek Soeder
- Jean-Francois Paquet

Duke Dept. of Statistical Sciences:

- Robert E. Wolpert
- Simon Mak
- Yi Ji

Funding:

US Dept. of Energy

National Science Foundation

NERSC National Energy Research Scientific Computing Center

Resources

Trento:

- J. Scott Moreland, Jonah E. Bernhard & Steffen A. Bass: Phys. Rev. C 92, 011901(R)
- https://github.com/Duke-QCD/trento

iEbE-VISHNU:

- Chun Shen, Zhi Qiu, Huichao Song, Jonah Bernhard, Steffen A. Bass & Ulrich Heinz: Computer Physics Communications in print, arXiv:1409.8164
- http://u.osu.edu/vishnu/

UrQMD:

- Steffen A. Bass et al. Prog. Part. Nucl. Phys. 41 (1998) 225-370 , arXiv:nucl-th/9803035
- Marcus Bleicher et al. J.Phys. G25 (1999) 1859-1896 , arXiv:hep-ph/9909407
- http://urqmd.org

MADAI Collaboration:

- Visualization and Bayesian Analysis packages
- https://madai-public.cs.unc.edu

Duke Bayesian Analysis Package:

- https://github.com/jbernhard/mtd

The End

Initial Condition Model: Trento

- effective, parametric, description of entropy production prior to thermalization
- based on **reduced thickness* T_R** as ansatz for *dS/dy*:

$$dS/dy\,|_{\tau=\tau_0} \propto T_R(p; T_A, T_B) \equiv \left(\frac{T_A^p + T_B^p}{2}\right)^{1/p}$$

- determine participant nucleons in A, B by sampling for each nucleon pair:

$$P_{\rm coll} = 1 - \exp\left[-\sigma_{gg} \int dx\, dy \int dz\, \rho_A \int dz\, \rho_B\right]$$

Nuclear Thickness*:

$$T_A = \sum_i \gamma_i \int dz\, \rho_{\rm nucleon}(x - x_i, y - y_i, z - z_i)$$

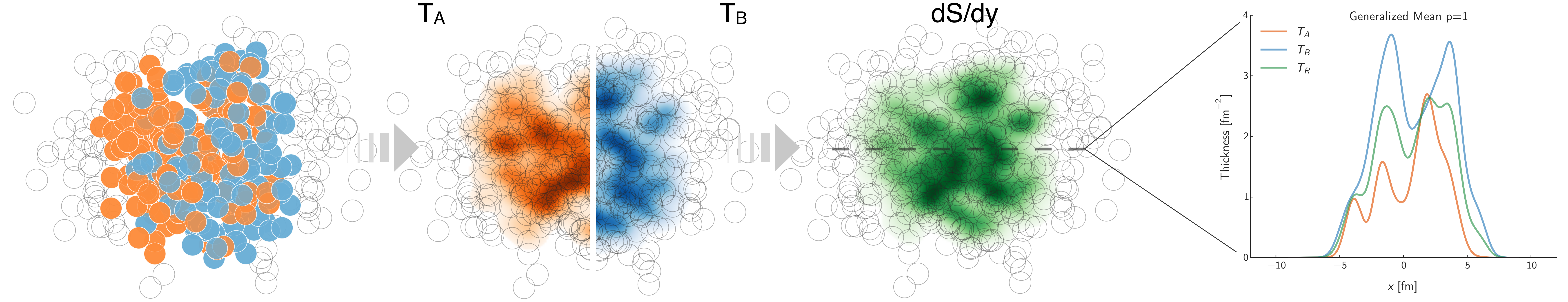
- sum is over participant nucleons with positions sampled from an uncorrelated Woods-Saxon distribution or correlated nuclear configurations when available
- introduce fluctuations via γ_i, sampled from a gamma distribution with unit mean:

$$P_k(\gamma) = \frac{k^k}{\Gamma(k)} \gamma^{k-1} e^{-k\gamma}$$

- nucleon density $\rho_{\rm nucleon}$ modeled as Gaussian in transverse plane

$$\int dz\, \rho_{\rm proton} = \frac{1}{2\pi w^2} \exp\left(-\frac{x^2 + y^2}{2w^2}\right)$$

(1) determine participants: **(2) construct thickness functions:** **(3) calculate entropy deposition:**

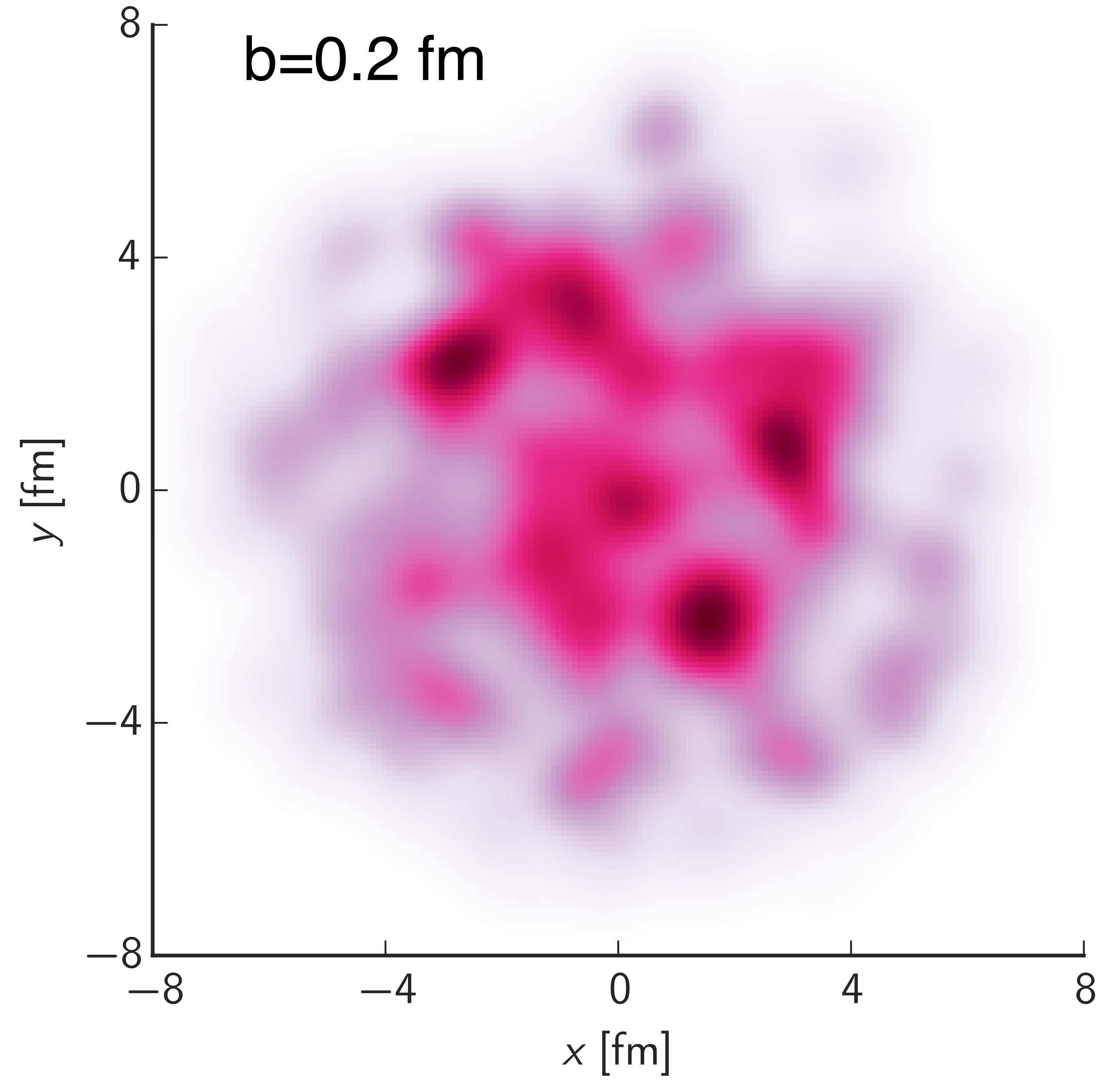


model parameters:

- attenuation parameter: p
- fluctuation parameter: k
- width of nucleon: w
- overall normalization: $C_{\rm norm}$

model output:

- event by event spatial entropy density distribution at mid-rapidity at thermalization time τ_0



Multivariate Output

Scaling of analysis with # of observables:

- independent emulators for each output?
  - neglects correlations among outputs
  - what if # of outputs scales to 100?
  - ▸ training of individual GPE's may become unfeasible and unnecessary in case of strong correlations

Multivariate Output

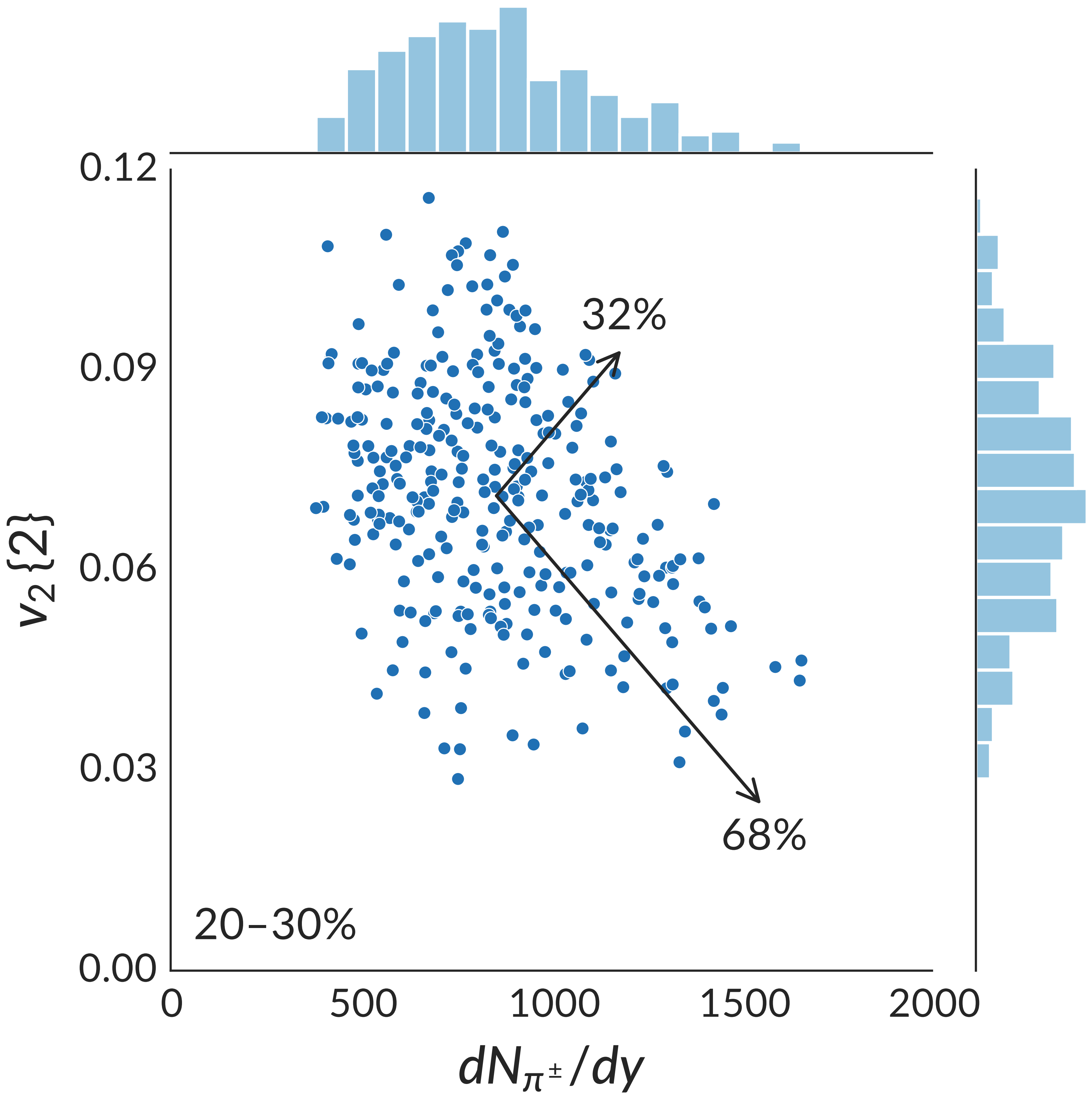
Scaling of analysis with # of observables:

- independent emulators for each output?
  - neglects correlations among outputs
  - what if # of outputs scales to 100?
  - ▸ training of individual GPE's may become unfeasible and unnecessary in case of strong correlations

Principal Components:

- linear combinations of model output
- orthogonal and uncorrelated

⇒ emulate each PC



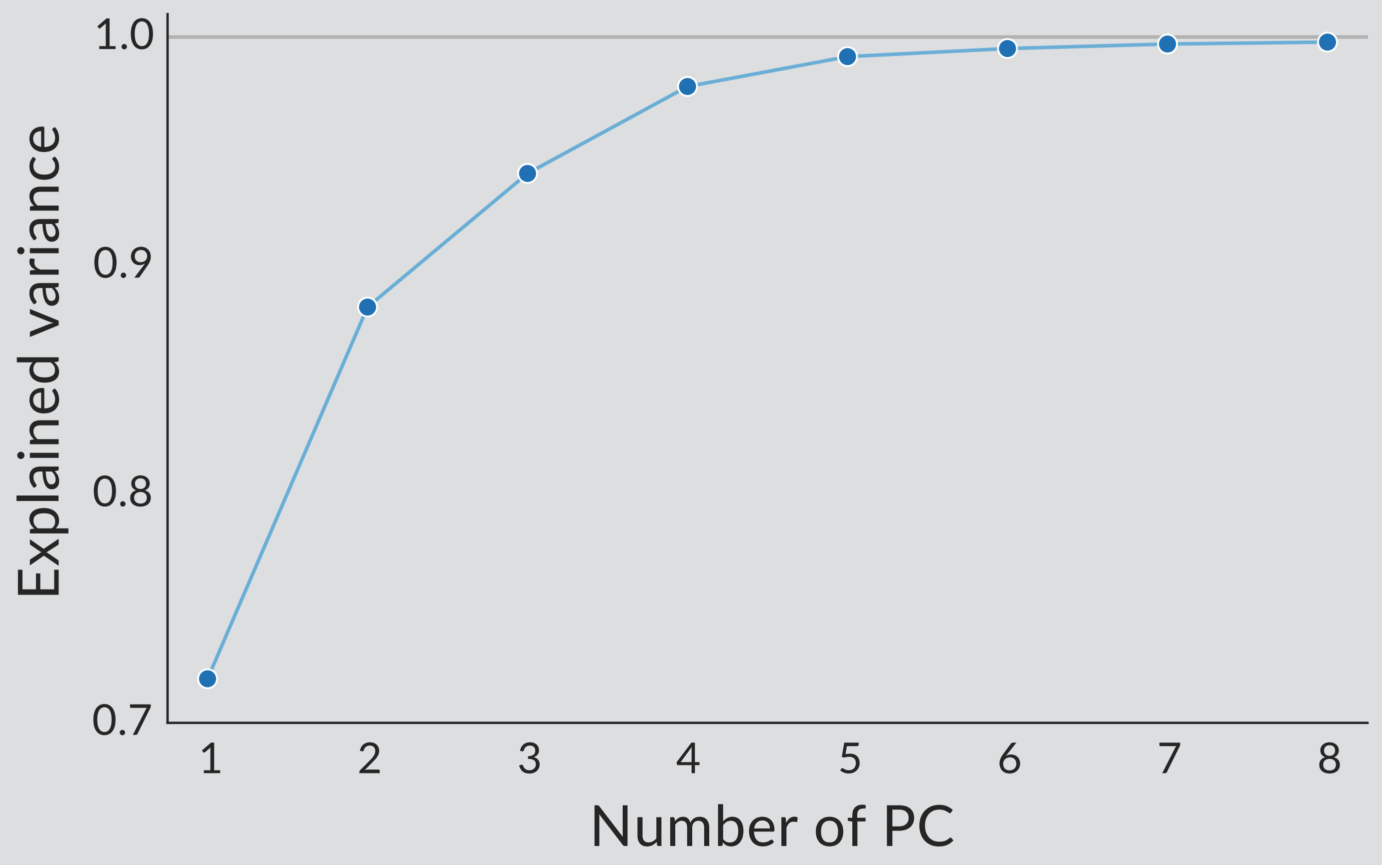
Multivariate Output

Scaling of analysis with # of observables:

- independent emulators for each output?
  - neglects correlations among outputs
  - what if # of outputs scales to 100?
  - ▸ training of individual GPE's may become unfeasible and unnecessary in case of strong correlations

this analysis:

- model outputs are yields, $\langle p_T \rangle$, v_2, v_3 and v_4
- 68 original output dimensions
- 8 principal components used



Principal Components:

- linear combinations of model output
- orthogonal and uncorrelated

⇒ emulate each PC