QGP parameter extraction via a global analysis of event-by-event flow coefficient distributions

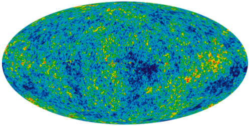
Jonah Bernhard

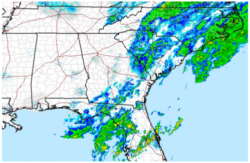
JET group meeting

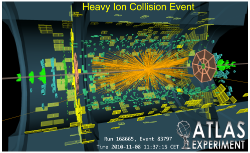
March 24, 2014

Model-to-data comparison

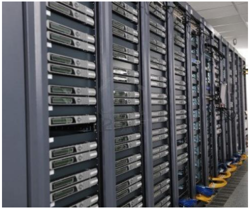
Data







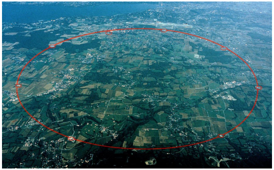
Computer model

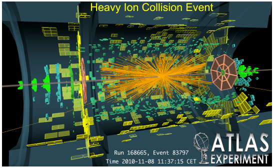


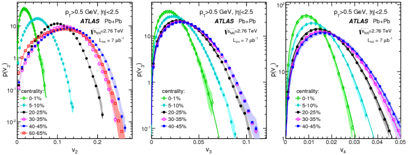
Inferences

Predictions

Model-to-data comparison: heavy-ion collisions



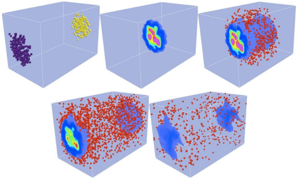


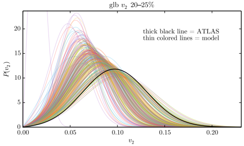


Model

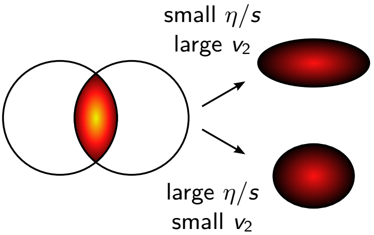
Initial conditions,

τ_0, η/s, ...

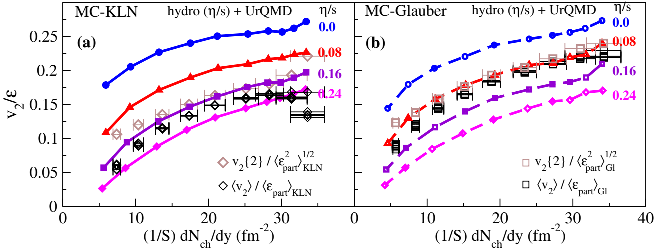




Measuring QGP η/s



- Observe experimental v_n.
- Run model with variable η/s.
- Constrain η/s by matching v_n.



H. Song, S. A. Bass, U. Heinz, T. Hirano and C. Shen, PRL **106**, 192301 (2011).

Extracting QGP properties

- Average flow $\langle v_n \rangle$.
- Vary only η/s, other parameters fixed.
- Only several discrete values.
- Qualitative constraints lacking uncertainty.

$\Longrightarrow$

- Event-by-event flow $P(v_n)$.
- Vary all salient parameters: η/s, τ_0, IC parameters, ...
- Continuous parameter space.
- Quantitative constraints including uncertainty.

Computer experiments with slow models

Challenges

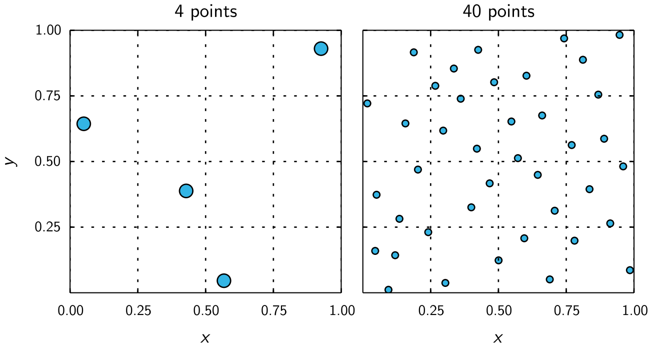
- Event-by-event models very computationally expensive, $\sim$1 hour per event.
- Need $\mathcal{O}(10^3)$ events per parameter-point to study fluctuations.
- Must vary all parameters simultaneously.

Strategies

- Evaluate model at efficient pre-determined parameter points.
  - Latin-hypercube sampling.
- Interpolate between explicitly calculated points.
  - Gaussian process emulator.

Latin-hypercube sampling

- Random set of parameter points.
- Maximizes CPU time efficiency.
- Skeleton of parameter space.



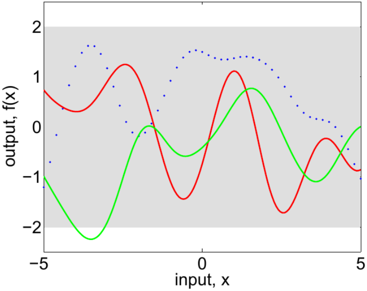
Gaussian processes

- *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*
- Instead of drawing variables from a distribution, functions are drawn from a process.

Require a covariance function, e.g.

$$\mathrm{cov}(x_1, x_2) \propto \exp\left[-\frac{(x_1 - x_2)^2}{2\ell^2}\right]$$

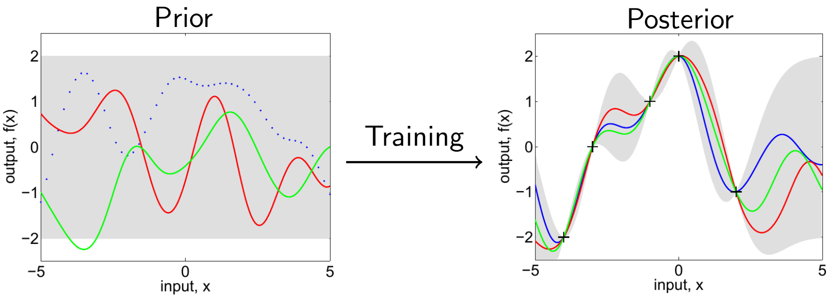
Nearby points correlated, distant points independent.



Gaussian Processes for Machine Learning, Rasmussen and Williams, 2006.

Gaussian process emulators

- Prior: the model is a Gaussian process.
- Posterior: Gaussian process conditioned on model outputs.



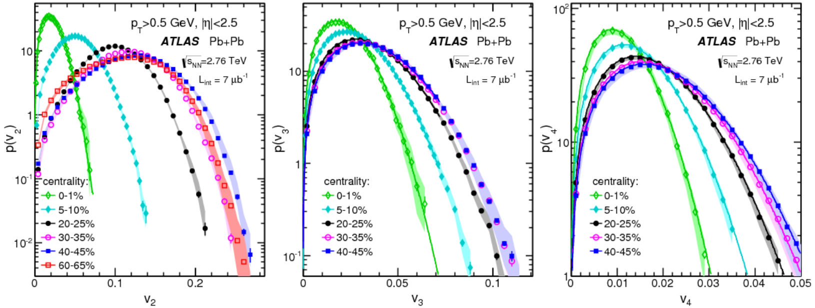
- Emulator is a fast surrogate to the actual model.
  - More certain near calculated points.
  - Less certain in gaps.

Experimental data

- ATLAS event-by-event flow distributions v_2, v_3, v_4.
- Fit to Rice / Bessel-Gaussian distribution

$$P(v_n) = \frac{v_n}{\delta_{v_n}^2} e^{-\frac{(v_n)^2+(v_n^{RP})^2}{2\delta_{v_n}^2}} I_0\left(\frac{v_n^{RP} v_n}{\delta_{v_n}^2}\right)$$

- Reduce to parameters v_n^{RP}, δ_{v_n}.



ATLAS Collaboration, JHEP **1311**, 183 (2013).

Event-by-event model

Modern version of Duke+OSU model VISHNU
(Viscous Hydro and UrQMD):

- MC-Glauber & MC-KLN initial conditions

  H.-J. Drescher and Y. Nara, Phys. Rev. C **74**, 044905 (2006).

- Viscous hydro

  H. Song and U. Heinz, Phys. Rev. C **77**, 064901 (2008).

- Cooper-Frye sampler

  Z. Qiu and C. Shen, arXiv:1308.2182 [nucl-th].

- UrQMD (Ultrarelativistic Quantum Molecular Dynamics)

  S. Bass *et. al.*, Prog. Part. Nucl. Phys. **41**, 255 (1998).
  M. Bleicher *et. al.*, J. Phys. G **25**, 1859 (1999).

$\rightarrow$ Tailored for running many events on Open Science Grid.

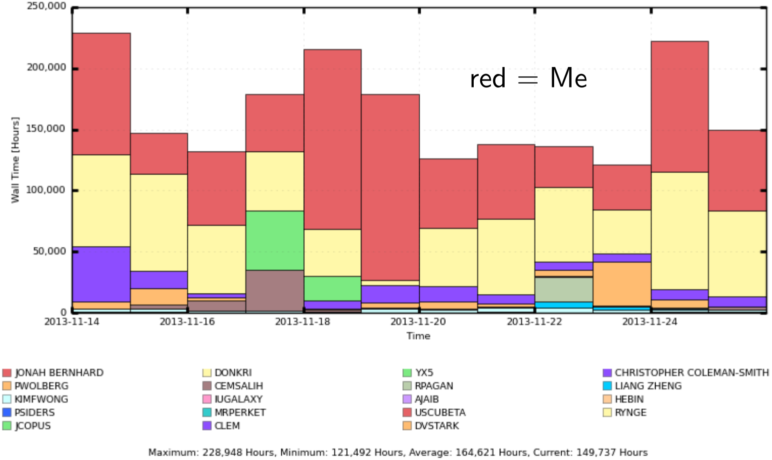
Computer experiment design

- Six centrality bins 0–5%, 10–15%, ... 50–55%.
- 256 Latin-hypercube points, five input parameters:
  - Normalization
  - IC-specific parameter
  - Thermalization time τ_0
  - Viscosity η/s
  - Shear relaxation time τ_Π
- Massive parallelization on Open Science Grid.
- Completed 1000–2000 events per centrality bin and input-parameter point.
  - 3.5 million total
  - 0.5 $\mu\mathrm{b}^{-1}$ (ATLAS: 7 $\mu\mathrm{b}^{-1}$)

Open Science Grid usage

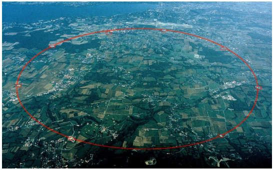
250,000

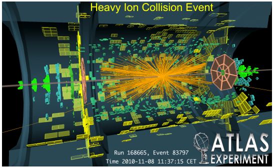
CPU hours per day

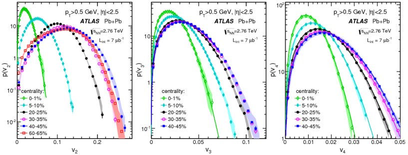


Completed KLN design (1.5 million events) in two weeks.

Model-to-data comparison: heavy-ion collisions



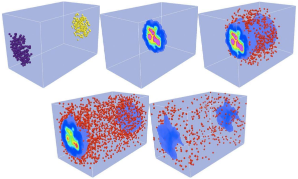


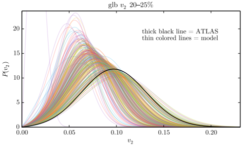


Model

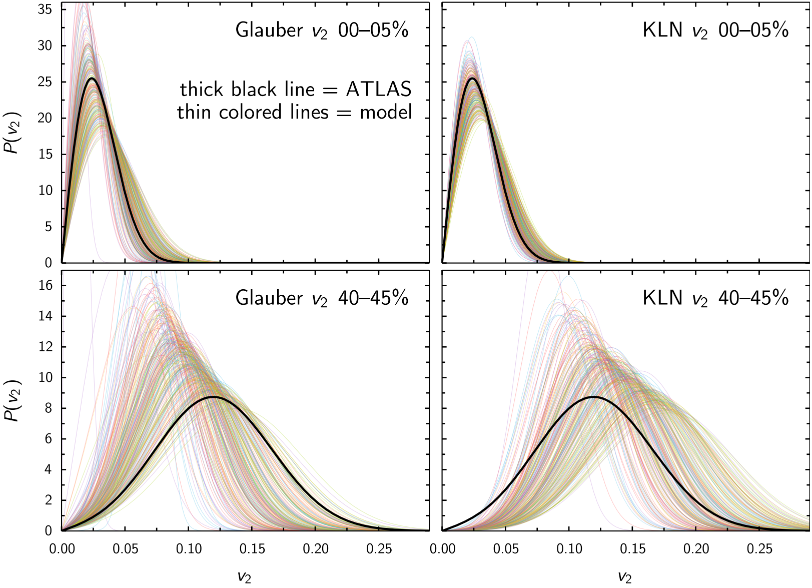
Initial conditions,

τ_0, η/s, . . .

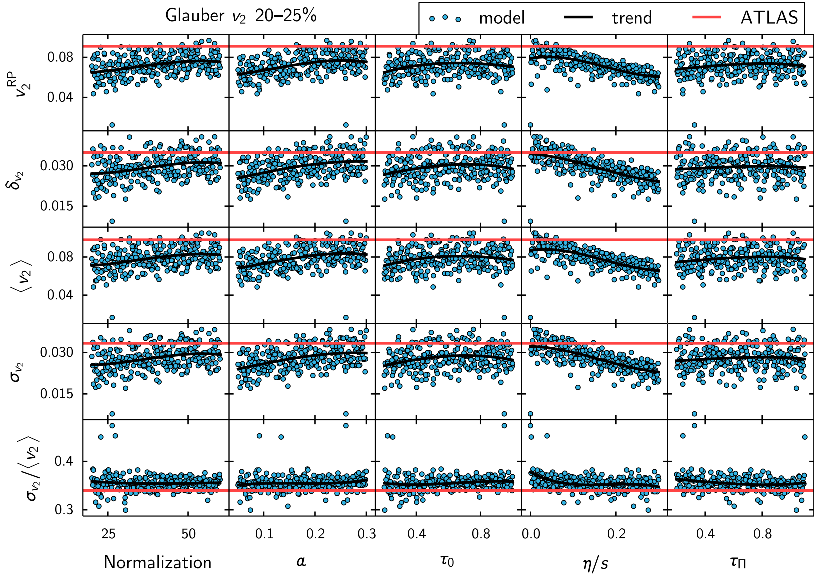




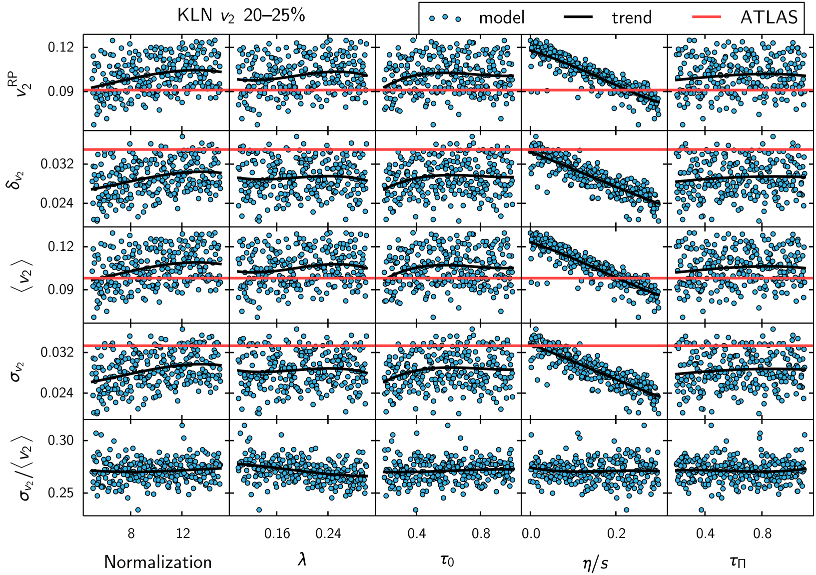
Model flow distributions



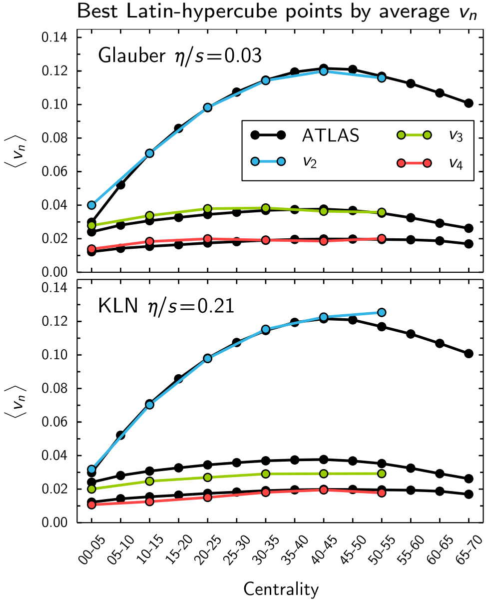
Input-output summary



Input-output summary

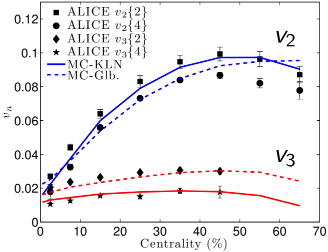


Best parameter points



OSU results, same model

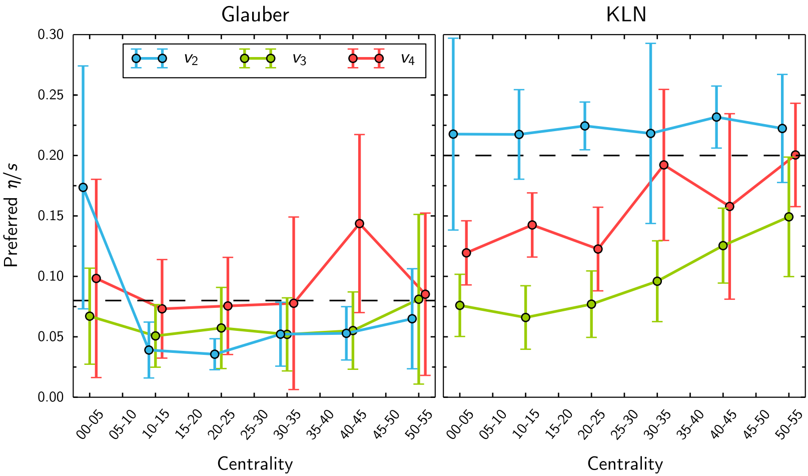
dashed: Glauber $\eta/s = 0.08$
solid: KLN $\eta/s = 0.20$



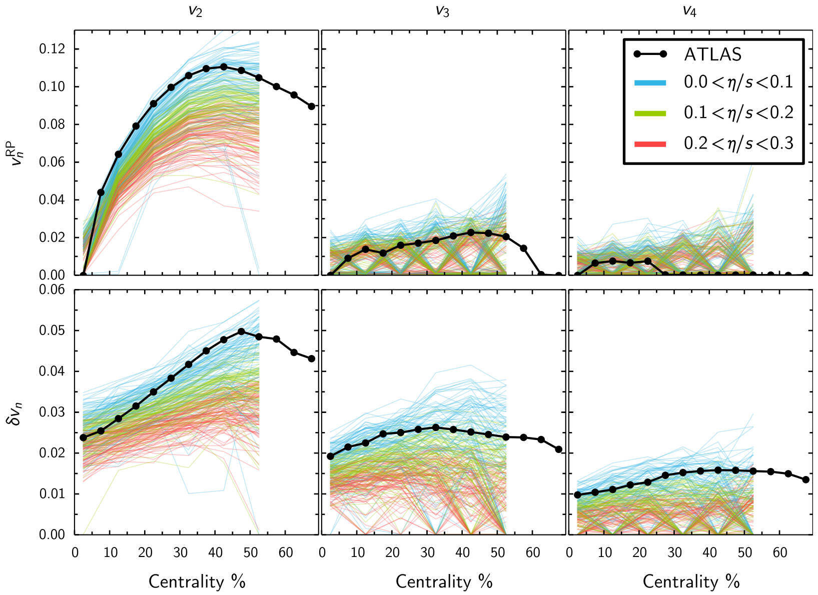
Z. Qiu, C. Shen, and U. Heinz, Phys. Lett. B **707**, 151 (2012).

Constraining η/s

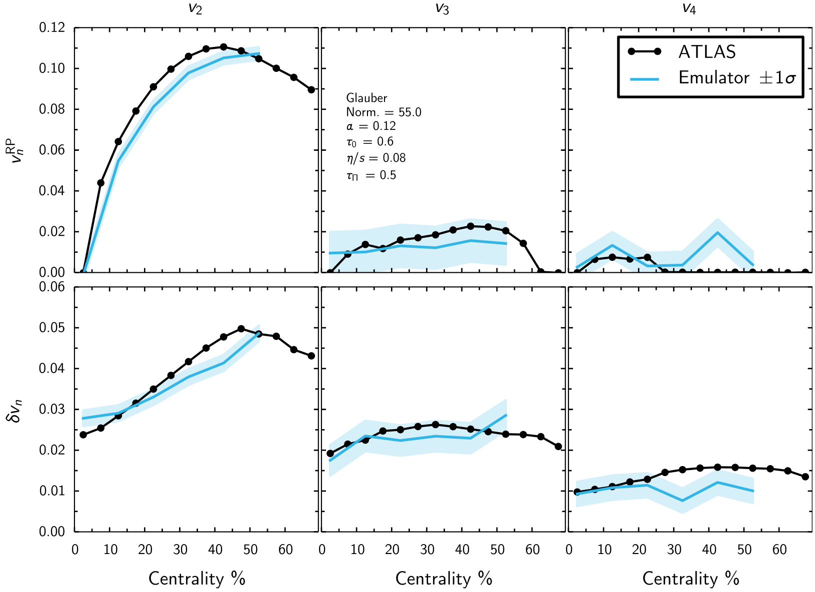
Points: average η/s of best 10 Latin-hypercube points by average v_n
Error bars: standard deviation of best 10
Dashed lines: canonical η/s (Glauber 0.08, KLN 0.20)



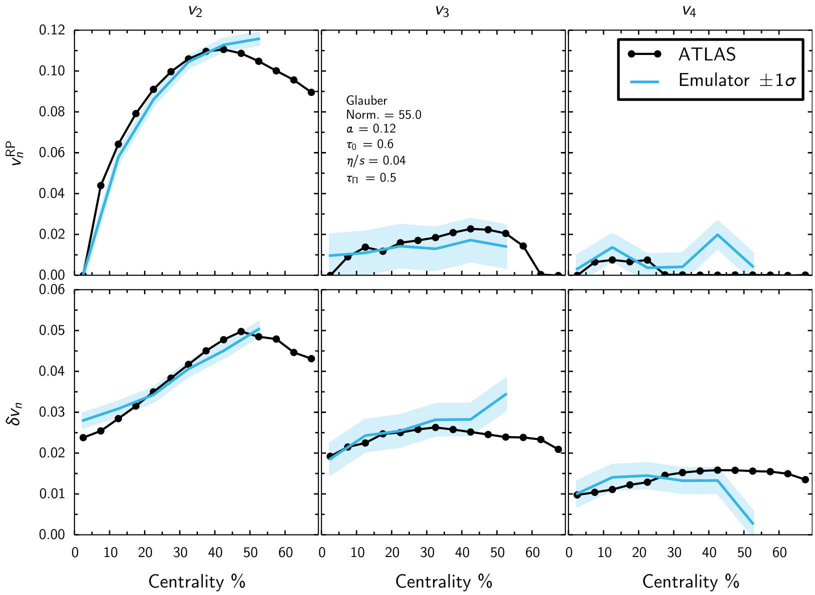
Bessel-Gaussian parameters (Glauber)



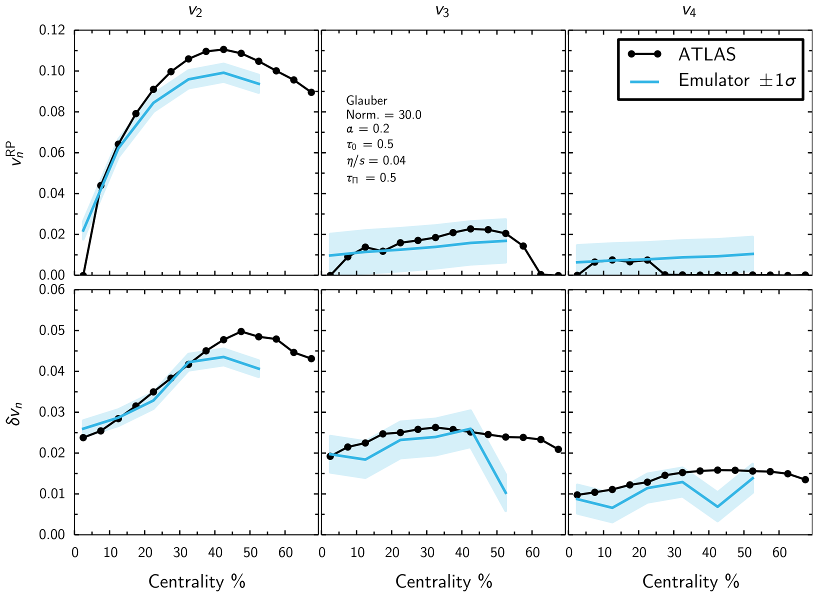
Emulated Bessel-Gaussian parameters



Emulated Bessel-Gaussian parameters



Emulated Bessel-Gaussian parameters



Summary & outlook

- Framework for massive event-by-event model-to-data comparison: new level of knowledge-extraction capability.
- Preliminary results consistent with previous work.

- Improve flow distribution parameter estimates.
- Validate emulator.
- Calculate posterior distributions
  $\rightarrow$ extract optimal values of parameters with uncertainty.
- Calibrate simultaneously on other observables, e.g. multiplicity.
- Repeat with more advanced models, especially initial conditions.

backup slides

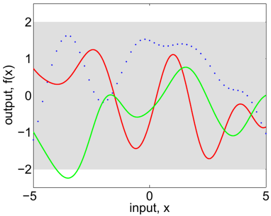
Generating Gaussian processes

- Choose a set of input points X_*.
- Choose a covariance function, e.g.

$$k(x_i, x_j) = \exp[-(x_i - x_j)^2/2]$$

  and create covariance matrix $K(X_*, X_*)$.
- Generate MVN samples (GPs)

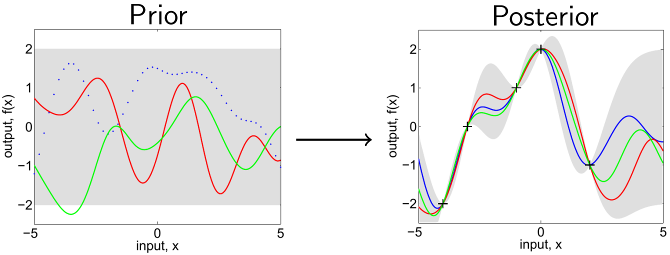
$$\vec{f}_* \sim \mathcal{N}[\vec{0}, K(X_*, X_*)].$$



Training the emulator

- Make observations $\vec{f}$ at training points X.
- Generate conditioned GPs

$$\vec{f}_*|X_*, X, \vec{f} \sim \mathcal{N}[K(X_*, X)K(X, X)^{-1}\vec{f},$$
$$K(X_*, X_*) - K(X_*, X)K(X, X)^{-1}K(X, X_*)].$$



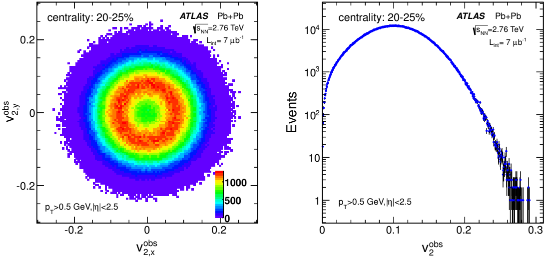
Rice / Bessel-Gaussian distribution

- Flow vectors follow bivariate Gaussian

$$P(\vec{v}_n) = \frac{1}{2\pi\delta_{v_n}^2} e^{-\frac{(\vec{v}_n - \vec{v}_n^{\mathrm{RP}})^2}{2\delta_{v_n}^2}}.$$

- Integrate out angle

$$P(v_n) = \frac{v_n}{\delta_{v_n}^2} e^{-\frac{(v_n)^2 + (v_n^{\mathrm{RP}})^2}{2\delta_{v_n}^2}} I_0\left(\frac{v_n^{\mathrm{RP}} v_n}{\delta_{v_n}^2}\right).$$



Finite multiplicity and unfolding

- Observed flow smeared by finite multiplicity and nonflow

$$P(v_n^{\text{obs}}) = \int P(v_n^{\text{obs}}|v_n)P(v_n)\,dv_n$$

where $P(v_n^{\text{obs}}|v_n)$ is the response function.

- Pure statistical smearing $\rightarrow$ Gaussian response

$$P(v_n^{\text{obs}}|v_n) = \frac{v_n^{\text{obs}}}{\delta_{v_n}^2} e^{-\frac{(v_n^{\text{obs}})^2+(v_n)^2}{2\delta_{v_n}^2}} I_0\left(\frac{v_n v_n^{\text{obs}}}{\delta_{v_n}^2}\right).$$

- v_n^{RP} unaffected; width increased as

$$\delta_{v_n}^2 \rightarrow \delta_{v_n}^2 + 1/2M.$$