How does magnetic geometry affect turbulence?

An interpretable machine learning approach

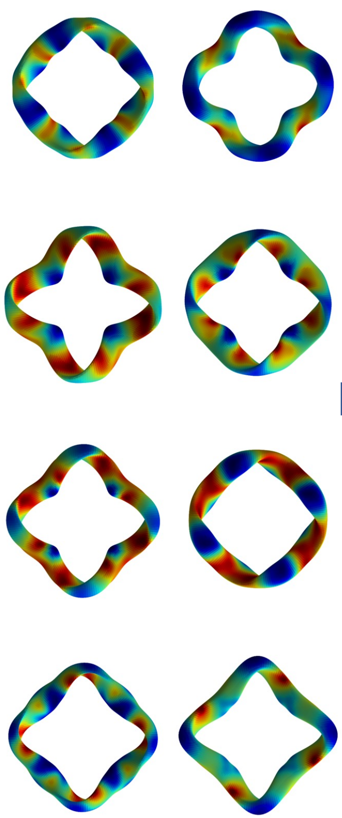
M Landreman, I Abel, G Abreu, C Alves, P Balaprakash, R M Churchill, R Conlin,

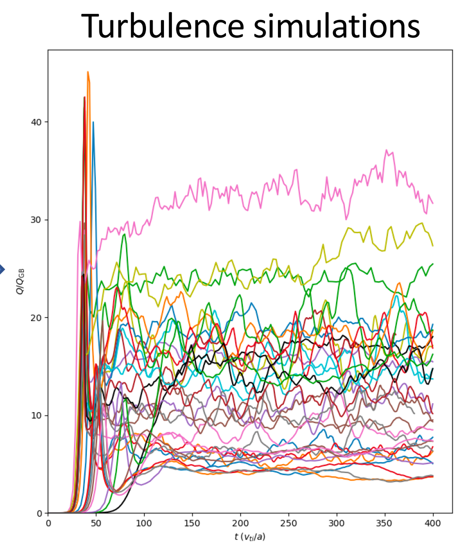
W Dorland, G Hammett, S Hurwitz, B Jang, R Jorge, A Kaptanoglu

Supported by the US DOE StellFoundry SciDAC & G.H. Distinguished Scientist Fellow Award

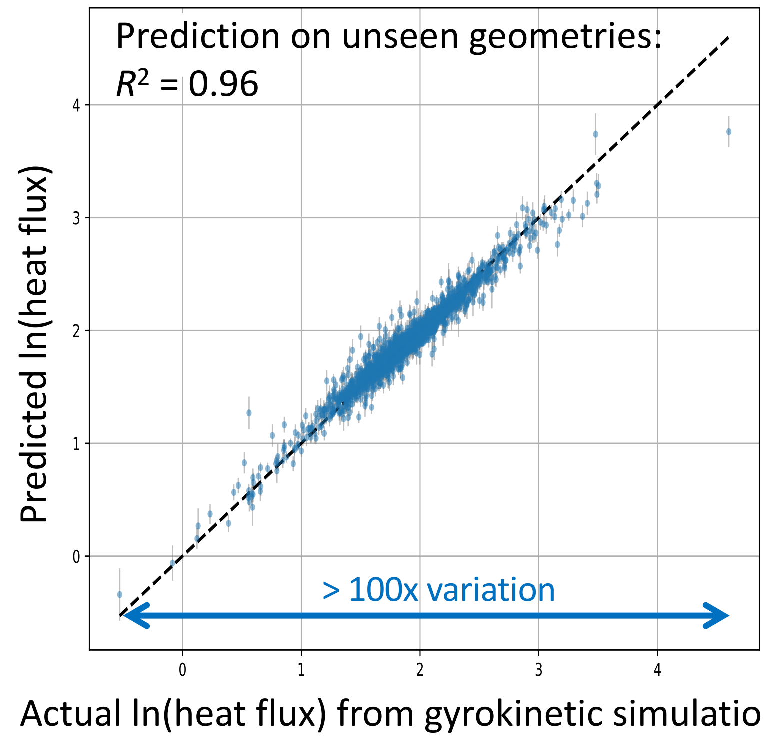
APS-DPP meeting, Atlanta, October 2024

Applying machine learning to a set of random stellarator geometries, we can predict the geometry-dependence of turbulence

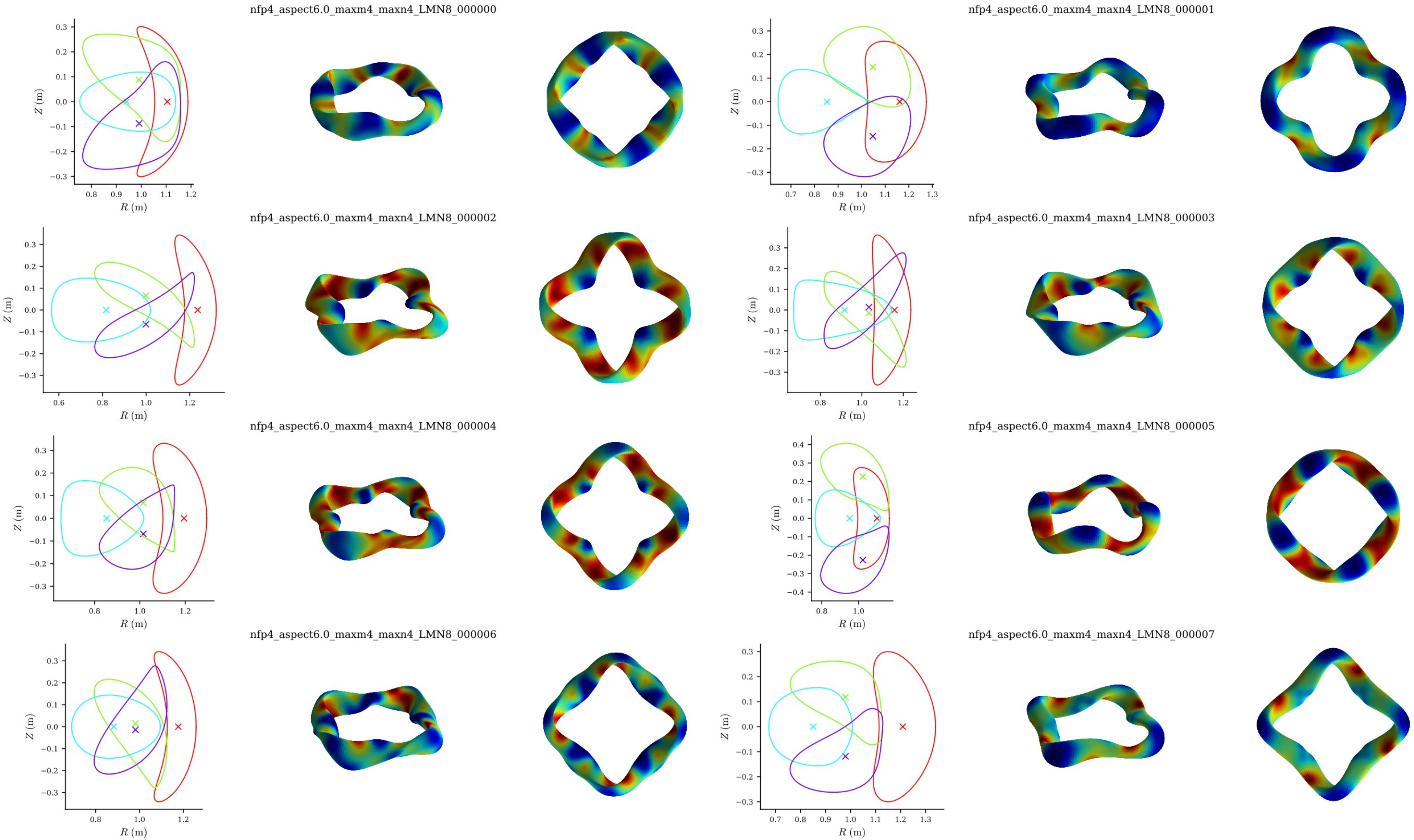




Regression

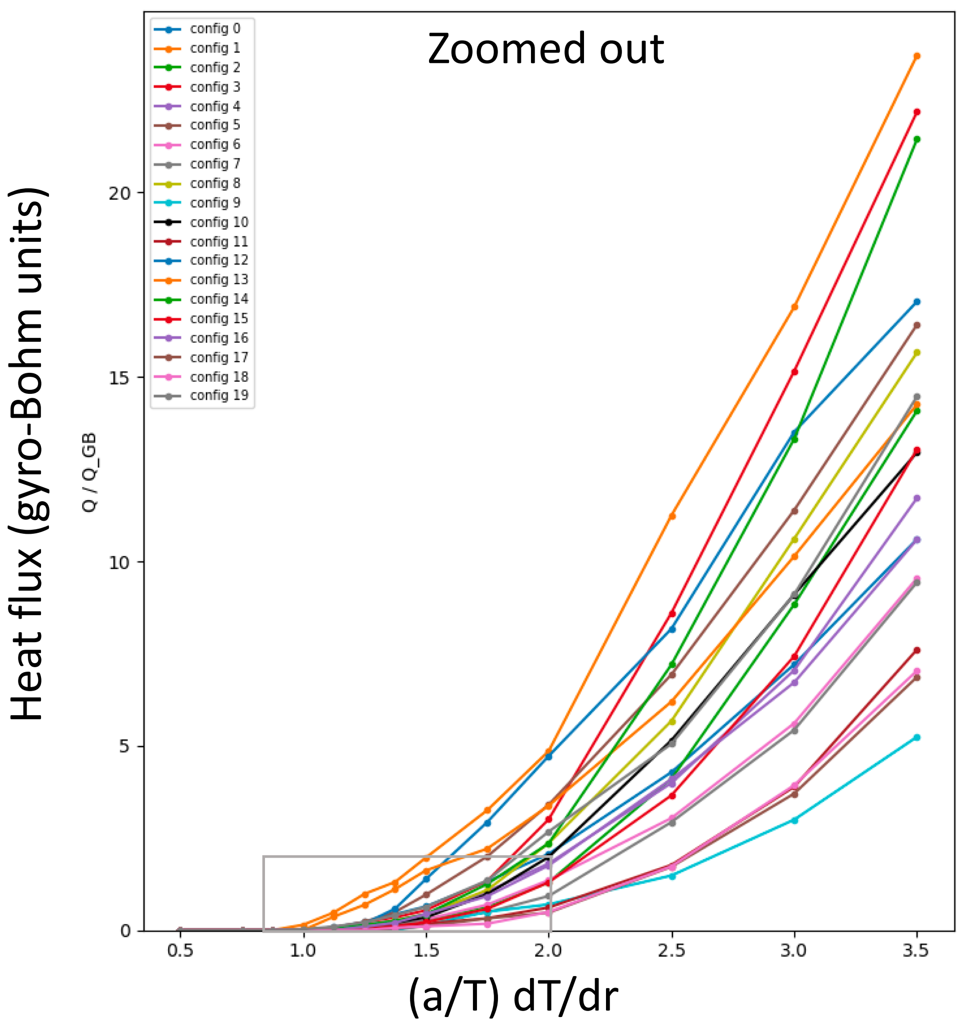


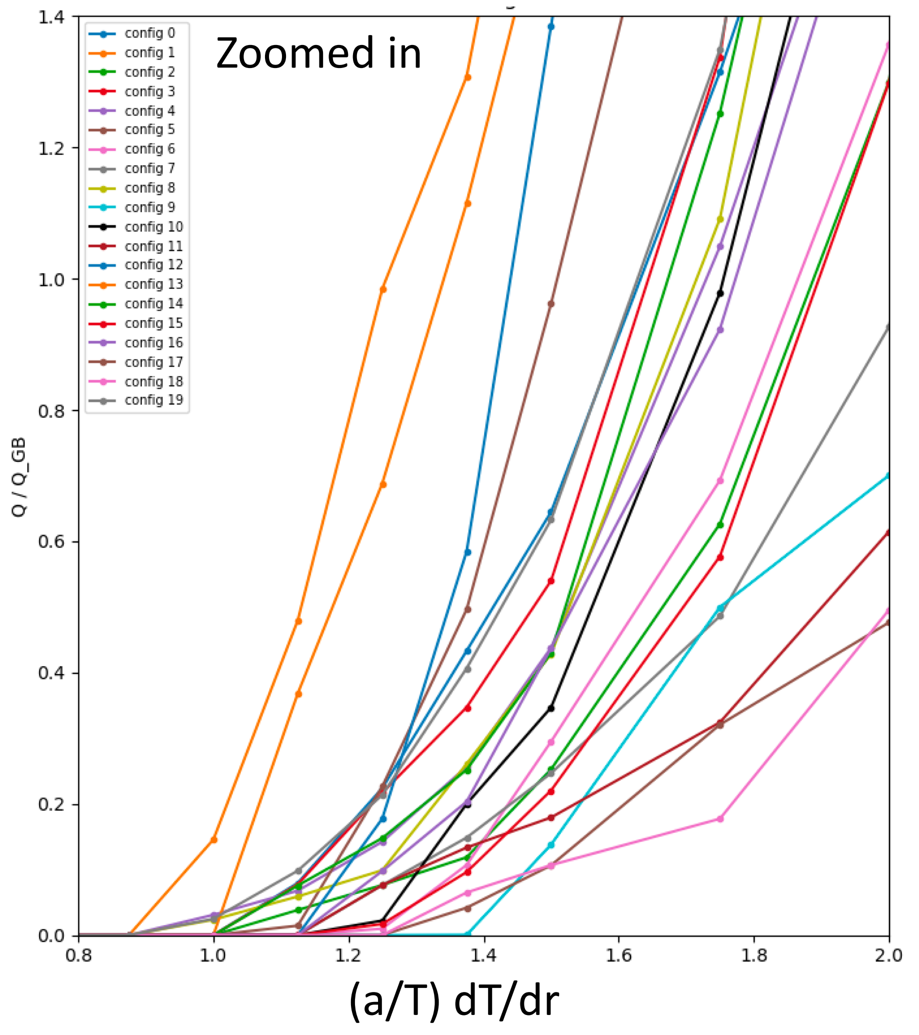
Step 1: Generate equilibria with random geometries



- Each RBC and ZBS coefficient sampled from normal distribution, fit to collection of well-known stellarators.
- All aspect ratio 6, same major & minor radius, same toroidal flux, so identical gyro-Bohm normalizations.

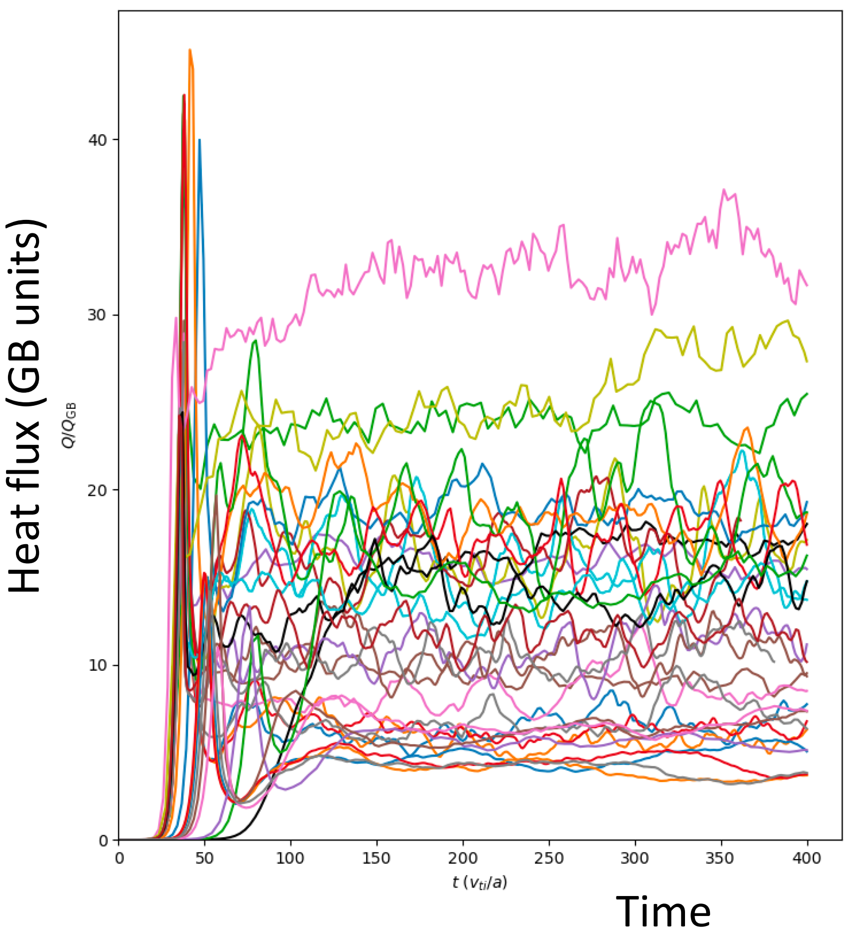
dT/dr scanned in nonlinear GX turbulence simulations for a few configurations. Critical gradient and stiffness are correlated.

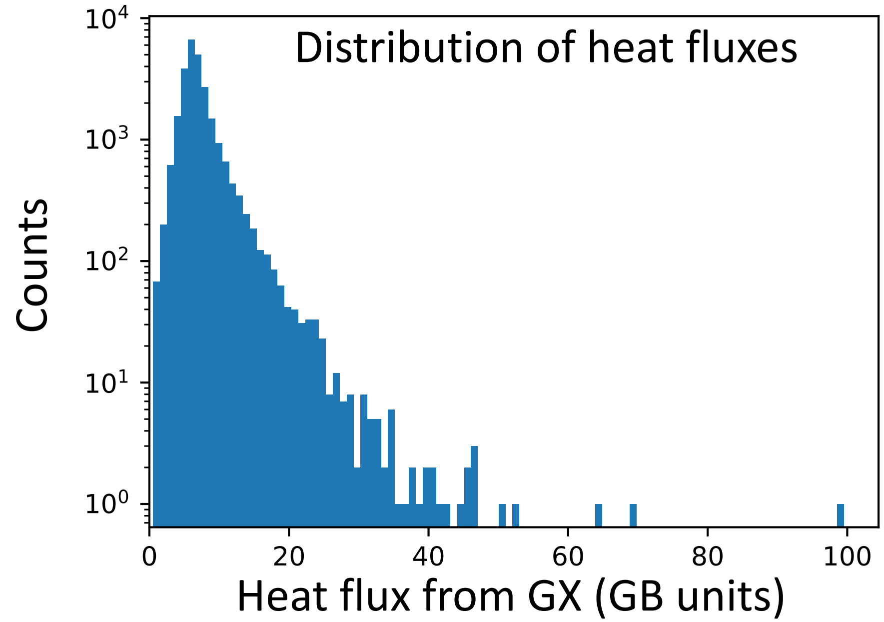


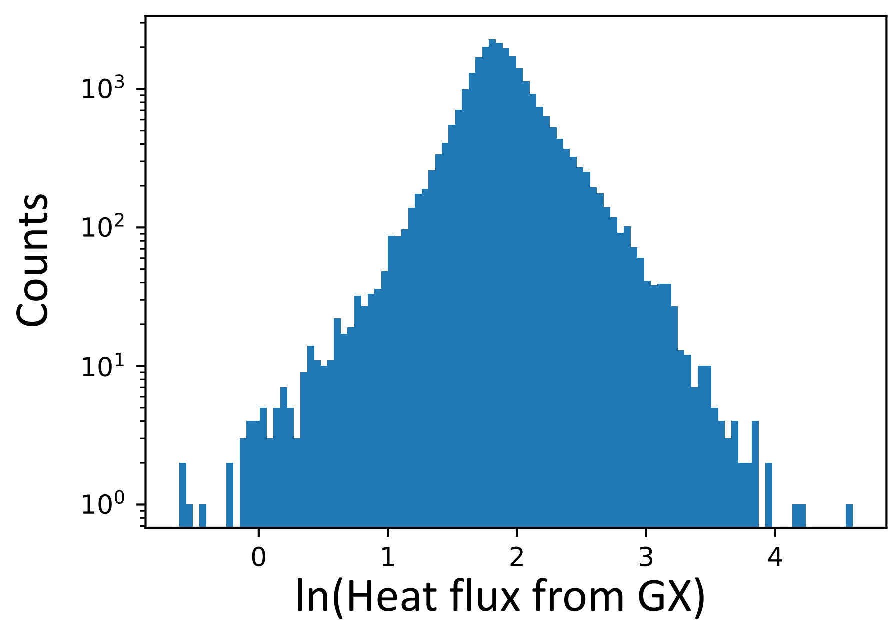


Nonlinear turbulence simulations run with GX in every equilibrium

- Fixed (a/T) dT/dr = 3, adiabatic electrons.
- ~ 8 minutes to get heat flux on 1 GPU
- So far N > 32k (8k equilibria * 4 flux tubes in each).

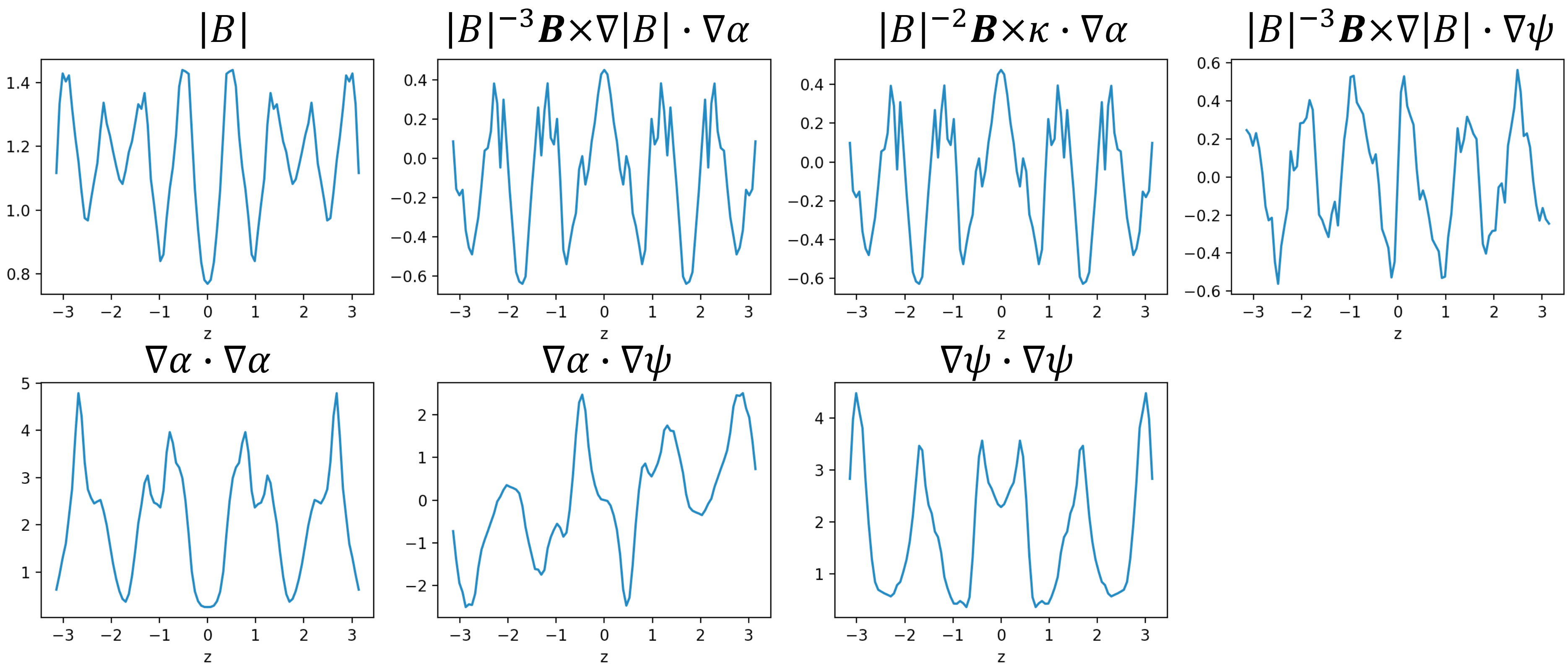




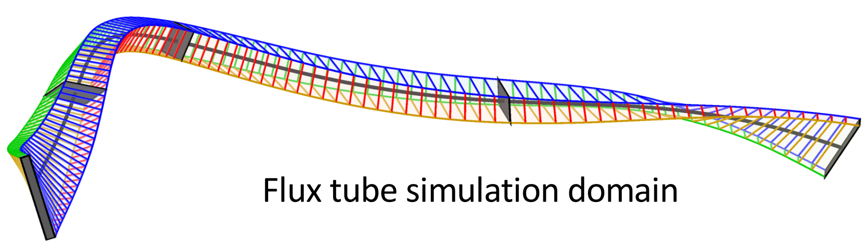


Raw feature space: 7x 1D functions that enter the turbulence simulations

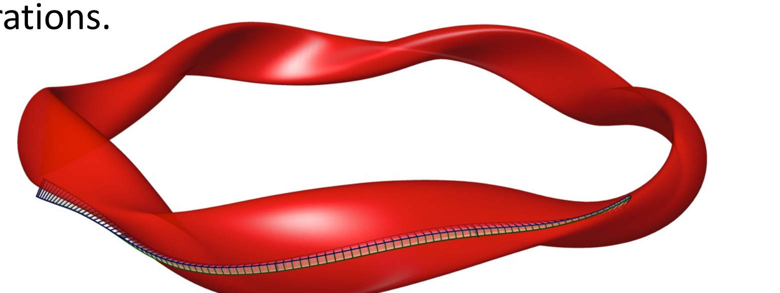
$$\boldsymbol{B} = \nabla\psi \times \nabla\alpha$$



$\boldsymbol{b}\cdot\nabla z$ is constant and the same for all configs, as are tube lengths in meters, so Fourier modes ($k_{||}$) can be compared between configurations.

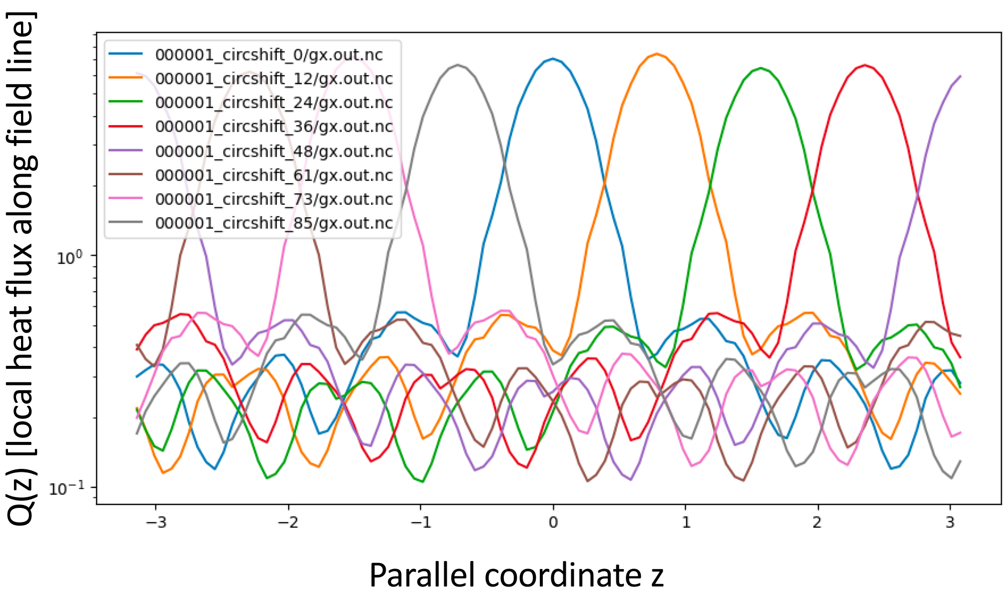


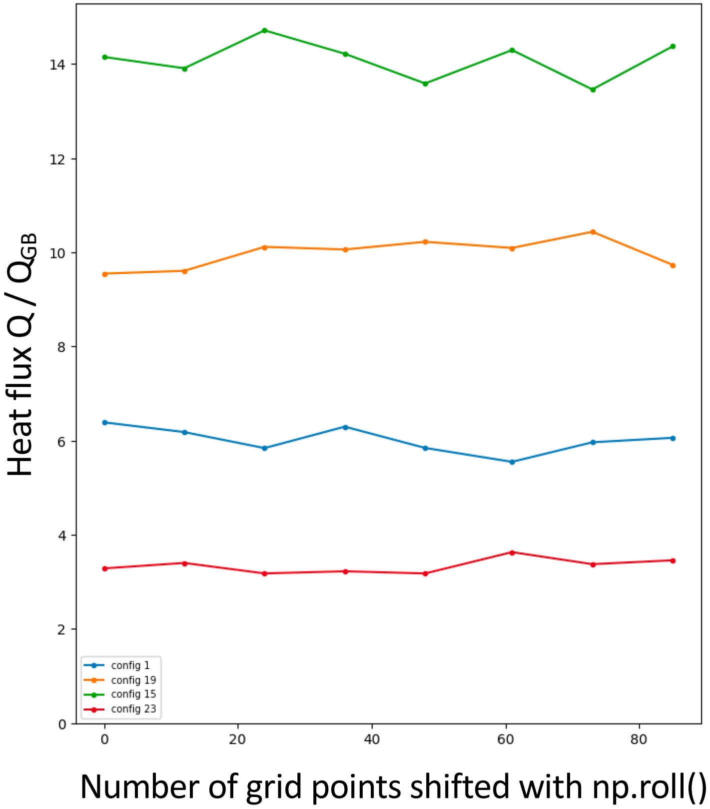
Flux tube simulation domain



Raw features should *not* be directly fed to classical regression or fully-connected neural network, since model should be translation-invariant

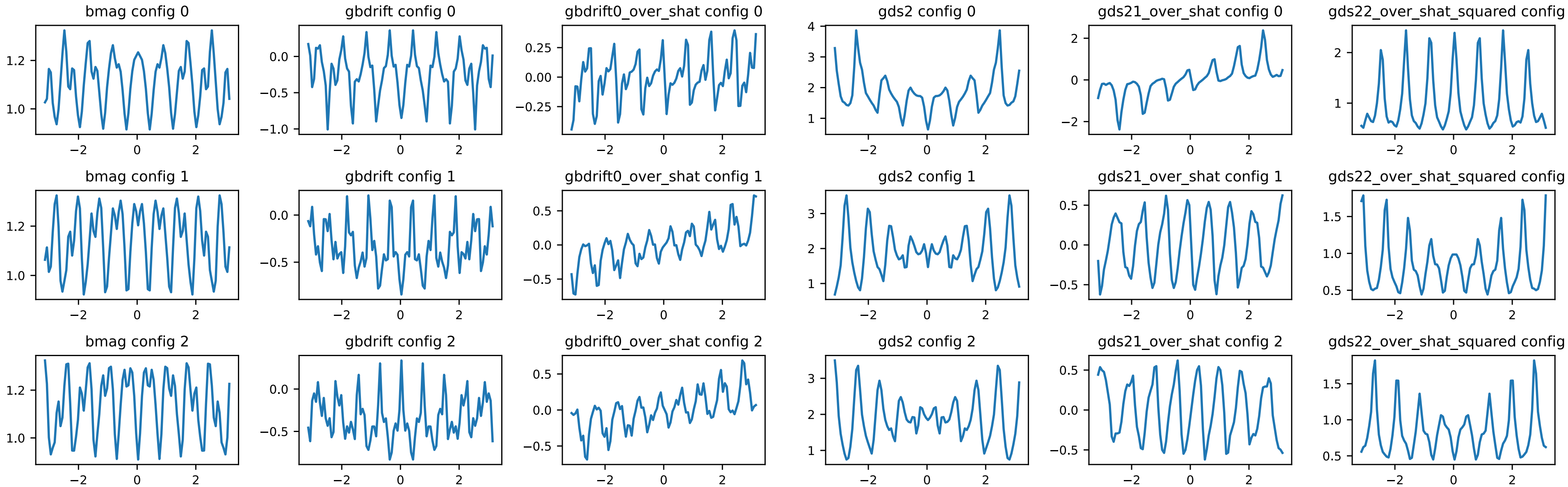
- GK equation, hence heat flux, is invariant under periodic translation of the raw features.
- Similar to computer vision, where convolutional neural networks give approximate translation-invariance.
- Demo: apply np.roll() to GX geometry input arrays, then re-run GX



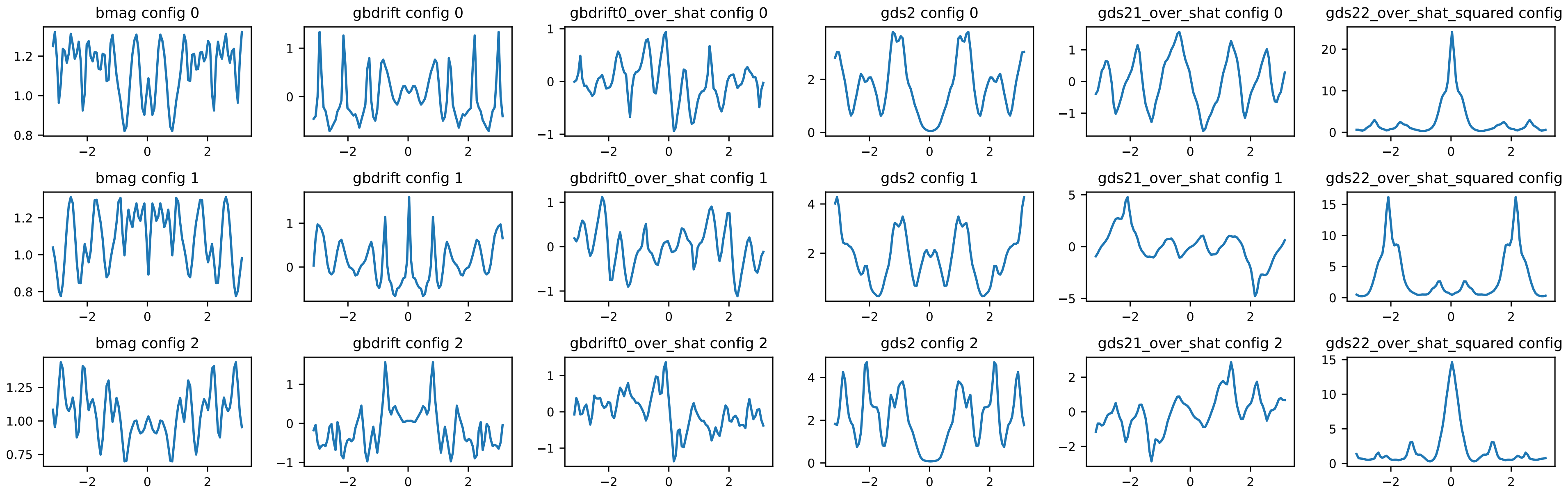


The configurations with lowest & highest heat flux have distinctive features

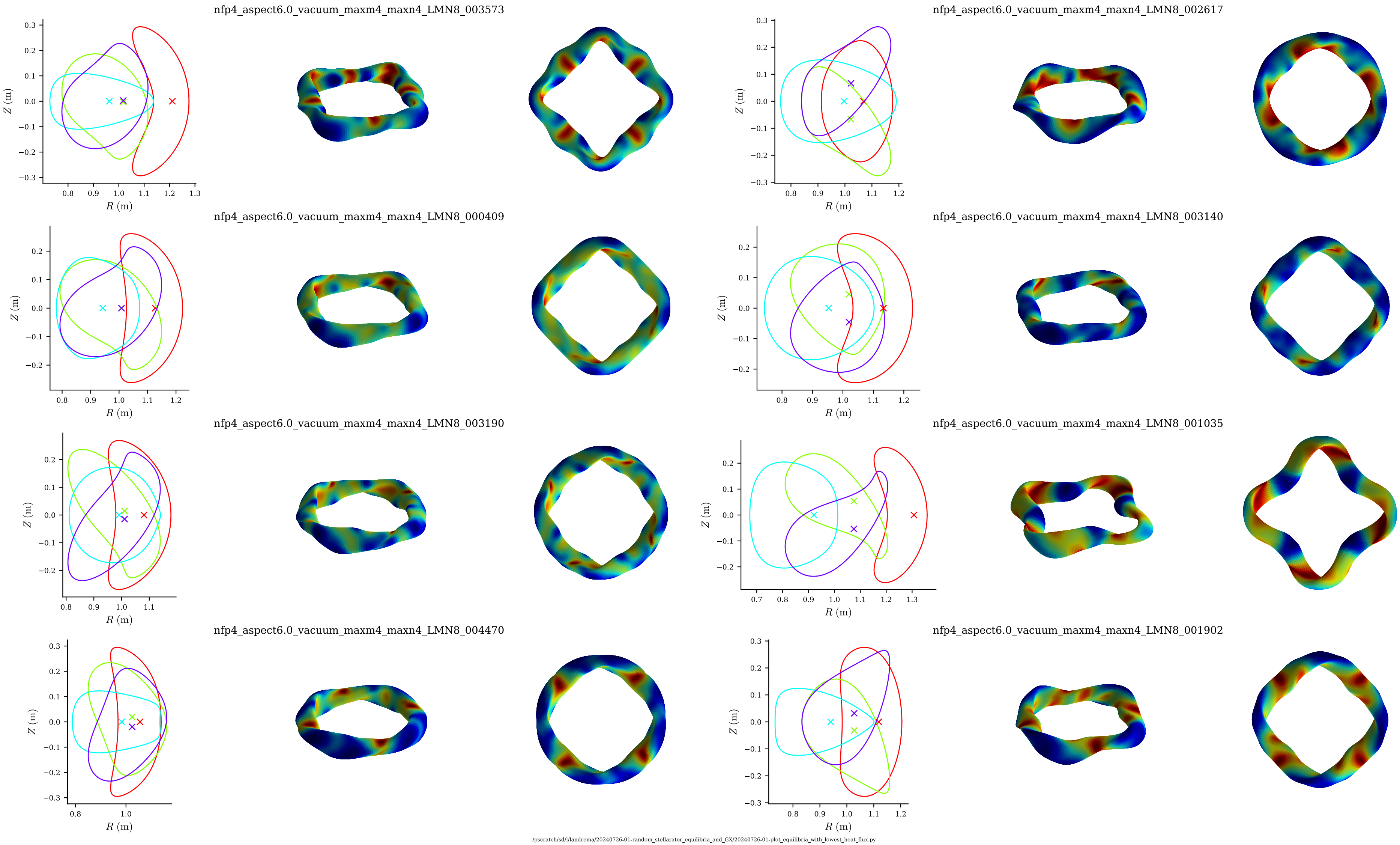
Smallest heat flux:



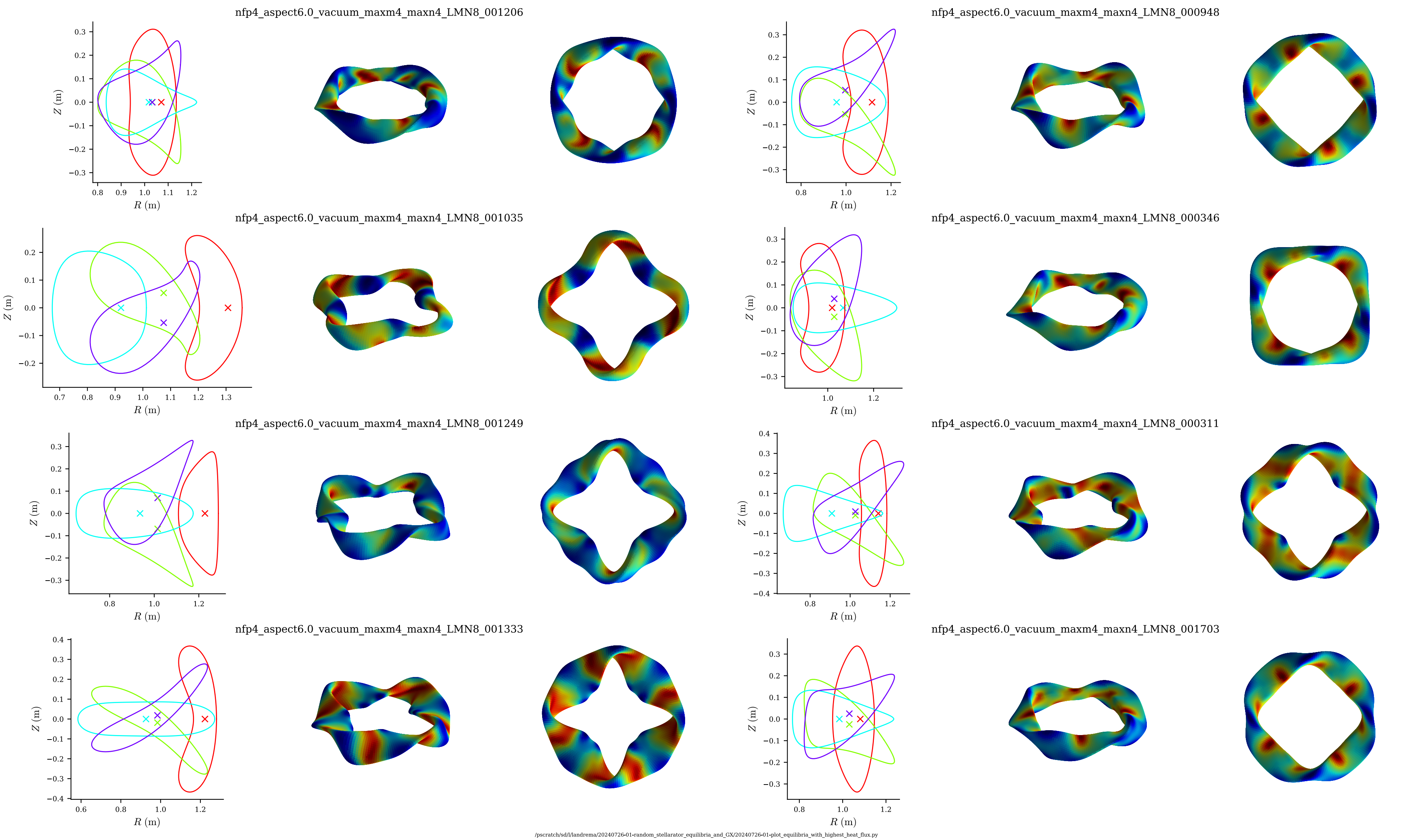
Largest heat flux:



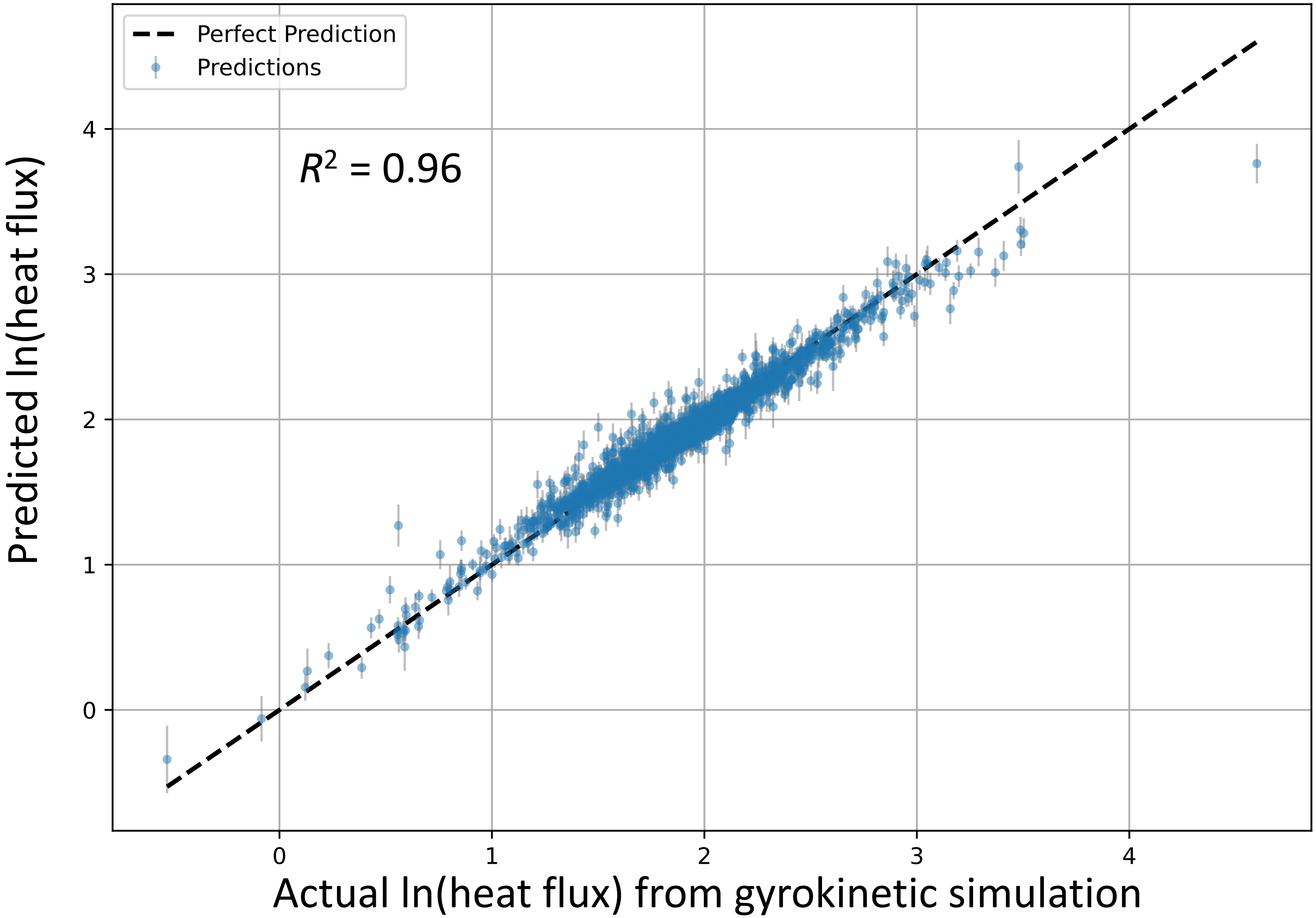
Configurations with smallest heat flux



Configurations with largest heat flux



Convolutional neural networks give a very accurate prediction of the turbulence



DeepHyper analysis by Caio Alvez, ORNL

A more interpretable ML approach: Feature engineering & selection

- Supplement the original GX inputs ($|B|$, $\mathbf{B}\times\nabla B\cdot\nabla\alpha$, $|\nabla\alpha|^2$, …) with local shear = $d/dz\,(\nabla\psi\cdot\nabla\alpha\,/\,|\nabla\psi|^2)$.
- Make many combinations of these “raw features”, giving new functions of z (e.g. $|\nabla\psi|^2 * \mathbf{B}\times\nabla B\cdot\nabla\psi$).
- To each, apply many different reductions over z that preserve translation-invariance (e.g. max, mean, etc). The results become the features to use for machine learning.
- Discard any features that have the same value for > 95% of the data.
- Apply forward sequential feature selection to pick out the few features that contribute most to R^2.

Feature set

Start with GX inputs, local shear, & inverses of the positive-definite quantities:
$F = \{B, B^{-3}\mathbf{B}\times\nabla B\cdot\nabla\alpha, B^{-3}\mathbf{B}\times\nabla B\cdot\nabla\psi, |\nabla\alpha|^2, \nabla\psi\cdot\nabla\alpha, |\nabla\psi|^2, \mathrm{localShear}, 1/B, 1/|\nabla\alpha|^2, 1/|\nabla\psi|^2\}$. Use vacuum fields to reduce raw features by 1 since $\mathbf{B}\times\boldsymbol{\kappa}\cdot\nabla\alpha = \mathbf{b}\times\nabla B\cdot\nabla\alpha$.

C(F) = all pairwise products in F (excluding x/x for x in B, $|\nabla\alpha|^2$, $|\nabla\psi|^2$).

M(F) = {Heaviside(x), Heaviside(-x)} for each x in F that can be both >0 and <0:
$B^{-3}\mathbf{B}\times\nabla B\cdot\nabla\alpha$, $B^{-3}\mathbf{B}\times\nabla B\cdot\nabla\psi$, $\nabla\psi\cdot\nabla\alpha$, localShear

J = {1, 1/B}, in case Jacobian $\propto 1/B$ helps.

Reductions: R = {min, max, max-min, mean, median, RMS, variance, skewness, quantiles 0.1..0.9, abs of fft coefficients 1-3, $k_{||}$ with largest amplitude, expected $k_{||}$, count above [-2, -1.5, … 6]}

Features: R(F * J), R(M * F * J), R(C * J), R(M * C * J)
Includes e.g. mean[$|\nabla\psi|^2$ * Heaviside($\mathbf{B}\times\nabla B\cdot\nabla\alpha$)], similar to Goodman et al (2024)

$\Longrightarrow$ 22,446 features

Apply forward sequential feature selection

Features added when using LightGBM:

1: meanSquared(Heaviside($\mathbf{B}\times\nabla B\cdot\nabla\alpha$) * $|\nabla\psi|^2 / B^2$)

2: mean(Heaviside($\mathbf{B}\times\nabla B\cdot\nabla\psi$) * ($\mathbf{B}\times\nabla B\cdot\nabla\psi$) / B^5)

3: argmax_kpar(Heaviside($\mathbf{B}\times\nabla B\cdot\nabla\alpha$) / B^2)

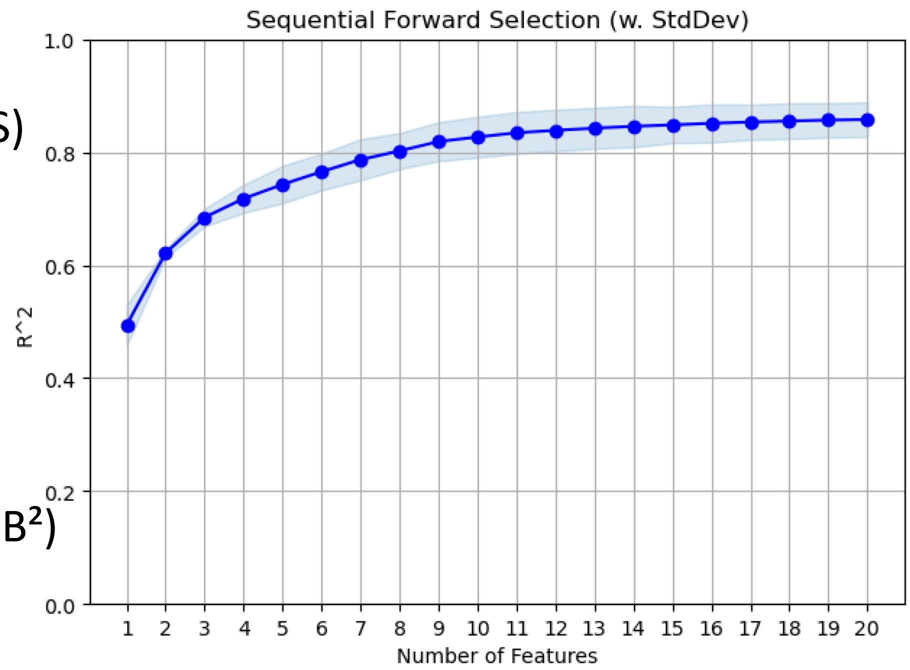
4: |FFTCoefficient1(B)|

5: quantile0.2(Heaviside($-\mathbf{B}\times\nabla B\cdot\nabla\psi$) * $|\nabla\psi|^2$ * S)

6: mean(Heaviside(-S) * ($\mathbf{B}\times\nabla B\cdot\nabla\alpha$) / B^4)

7: |FFTCoefficient1($B^{-3}\mathbf{B}\times\nabla B\cdot\nabla\psi$)|

8: mean(B * $|\nabla\psi|^2$)



Features added when using XGBoost:

1: meanSquared(Heaviside($\mathbf{B}\times\nabla B\cdot\nabla\alpha$) * $|\nabla\psi|^2 / B^2$)

2: mean(Heaviside($\mathbf{B}\times\nabla B\cdot\nabla\psi$) * ($\mathbf{B}\times\nabla B\cdot\nabla\psi$) / B^5)

3: argmax_kpar(Heaviside($\mathbf{B}\times\nabla B\cdot\nabla\alpha$) / B^2)

4: quantile0.4(Heaviside(-S) * S * $|\nabla\psi|^2$ / B)

5: |FFTCoefficient1($B^{-3}\mathbf{B}\times\nabla B\cdot\nabla\psi$)|

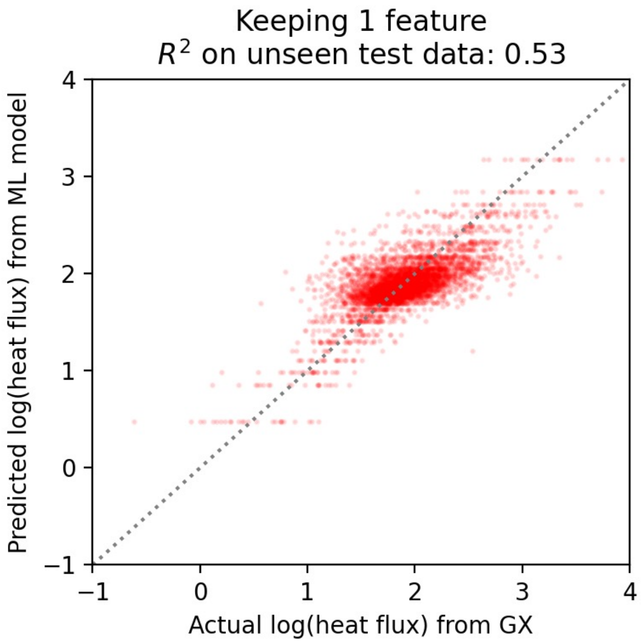
6: mean(($\mathbf{B}\times\nabla B\cdot\nabla\alpha$) / B^2)

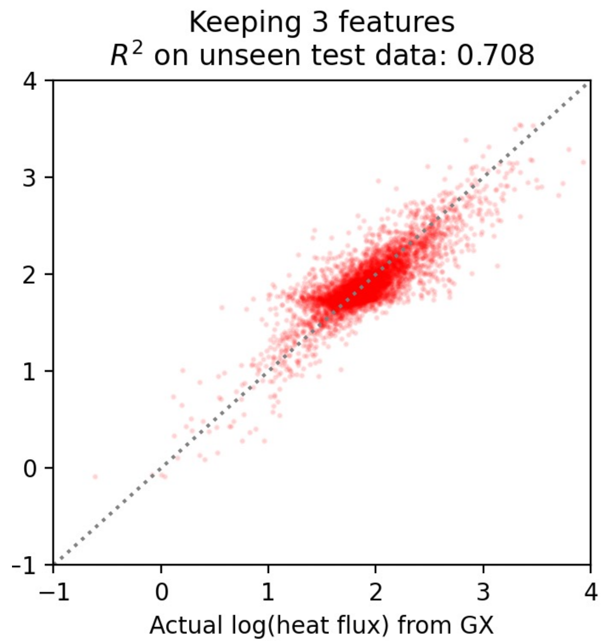
7: mean(Heaviside($\nabla\psi\cdot\nabla\alpha$) * B * $|\nabla\psi|^2$)

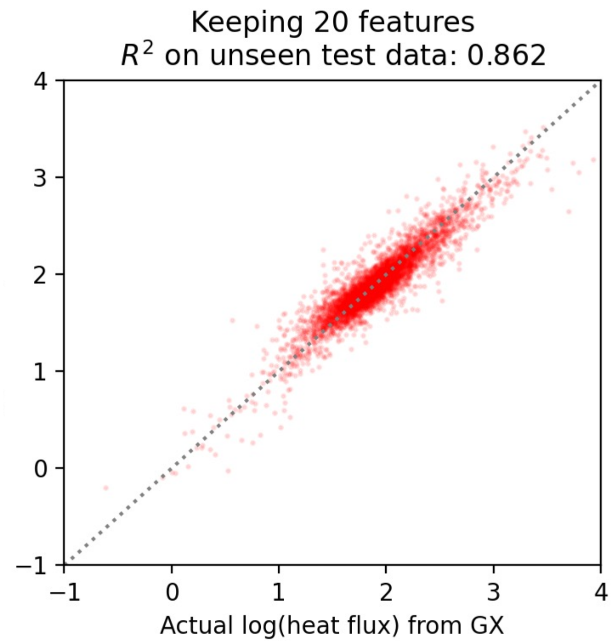
8: |FFTCoefficient2($B^{-3}\mathbf{B}\times\nabla B\cdot\nabla\psi$)|

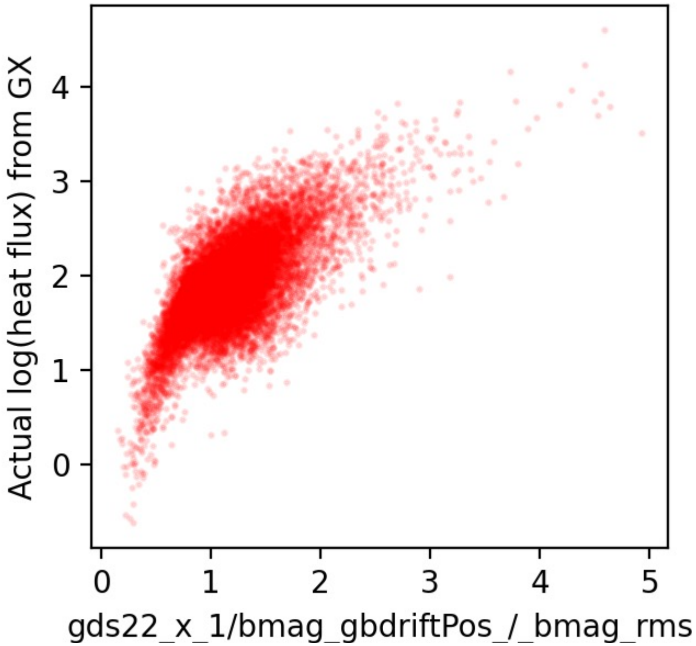
S = local shear = d/dz ($\nabla\psi\cdot\nabla\alpha$ / $|\nabla\psi|^2$).

Results of forward sequential feature selection



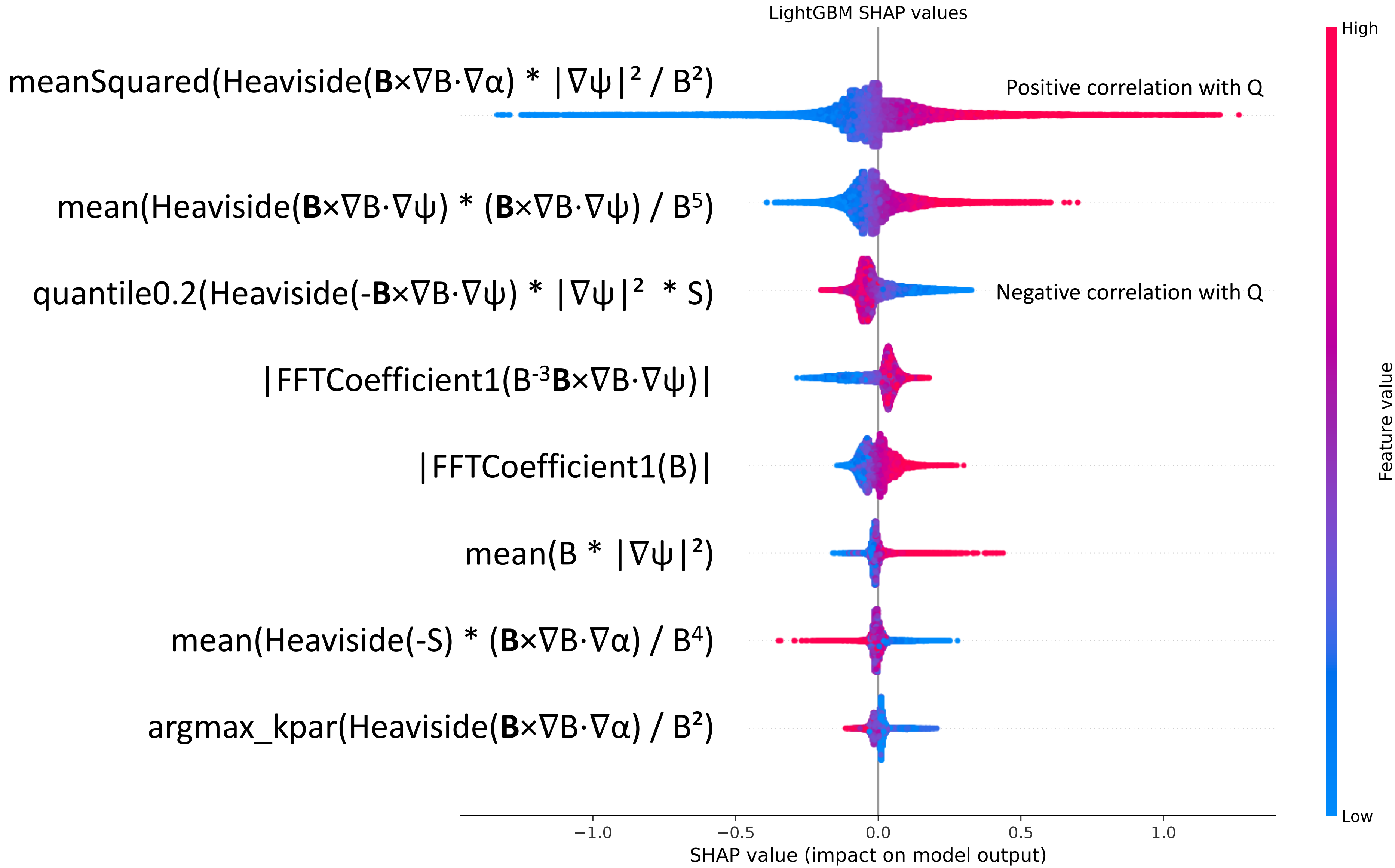






← No regression model,
Just showing correlation with just the 1 top feature

Shapley values show the sign and magnitude of each feature’s effect



The most important features are consistent with recent quantities suggested from theory

<u>Top features selected:</u>

1: meanSquared(Heaviside($\mathbf{B}\times\nabla B\cdot\nabla\alpha$) * $|\nabla\psi|^2$ / B^2), positive correlation with Q

Jacobian (squared)

Where there is bad curvature,

local temperature gradient (squared) in real space

Similar to ideas in Stroteich (2022), Goodman (2024).

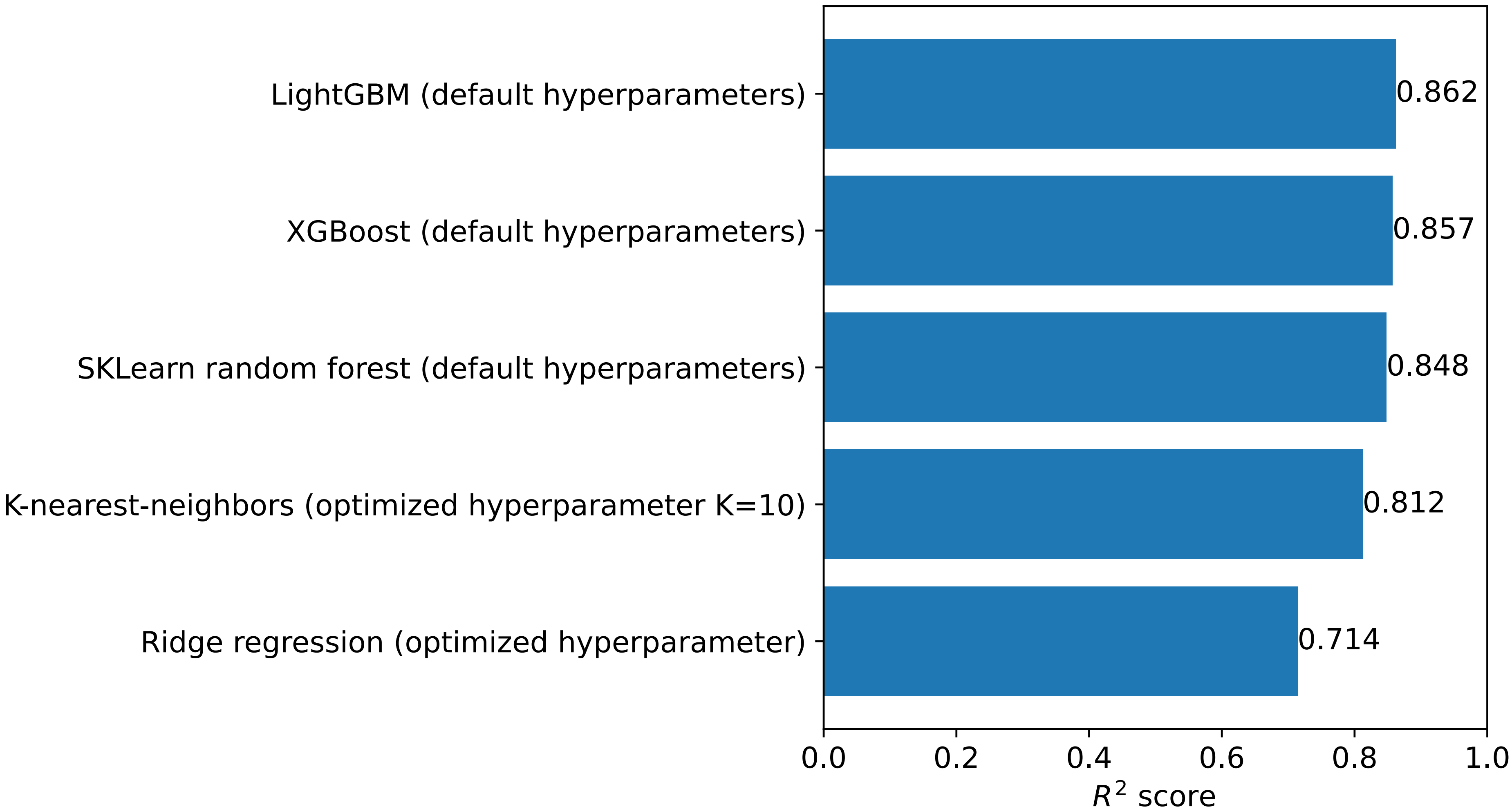
2: mean(Heaviside($\mathbf{B}\times\nabla B\cdot\nabla\psi$) * ($\mathbf{B}\times\nabla B\cdot\nabla\psi$) / B^5), positive correlation with Q

Consistent with Nakata (2022):
Larger geodesic curvature (radial drift) → smaller zonal flows → higher Q

3: argmax_kpar(Heaviside($\mathbf{B}\times\nabla B\cdot\nabla\alpha$) / B^2), negative correlation with Q

Dominant $k_{||}$ of the bad curvature - possibly related to critical balance?
Sign of correlation is consistent with Barnes et al PRL (2011).

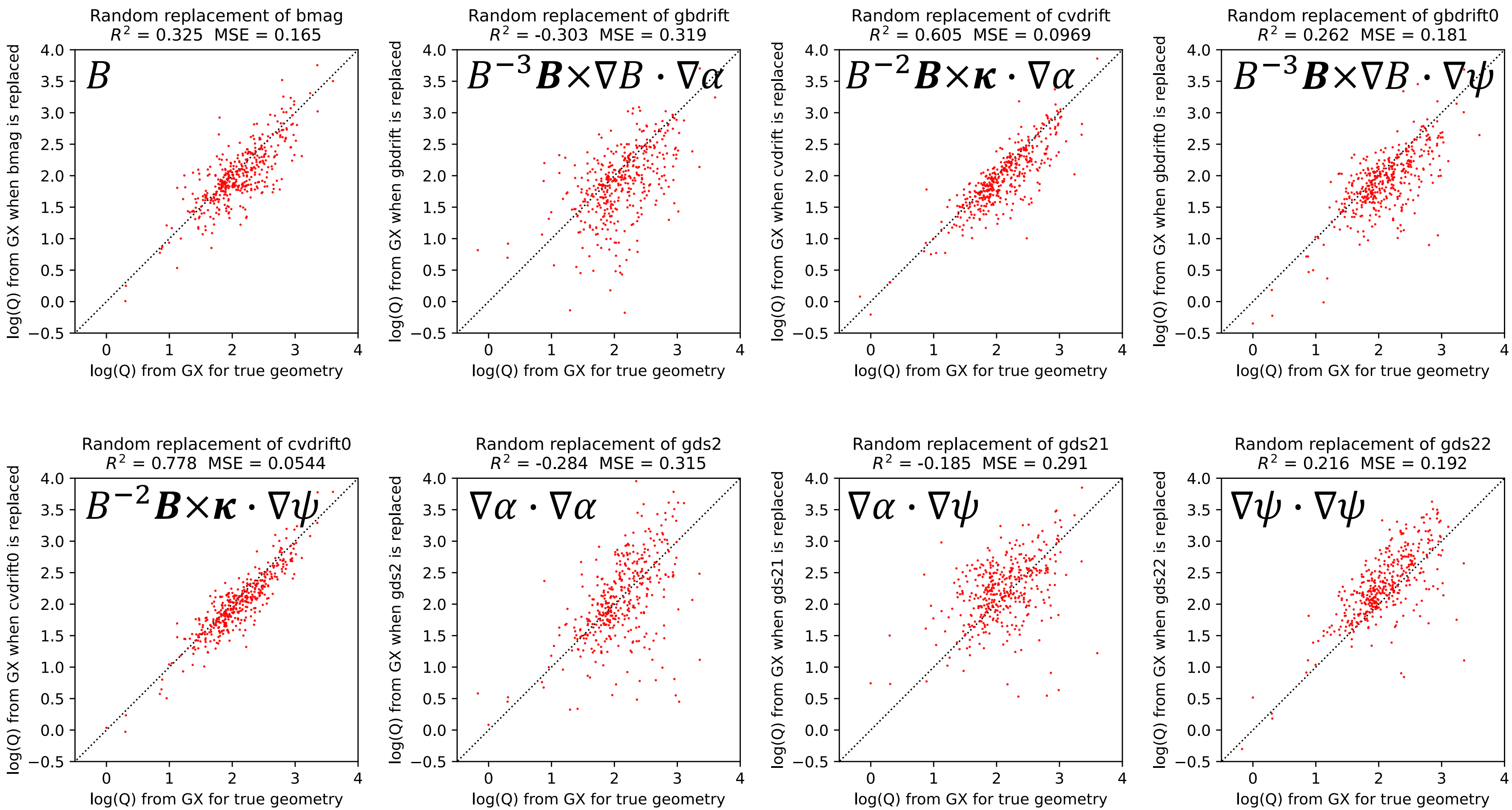
Other classical ML regression methods work also but are somewhat less accurate



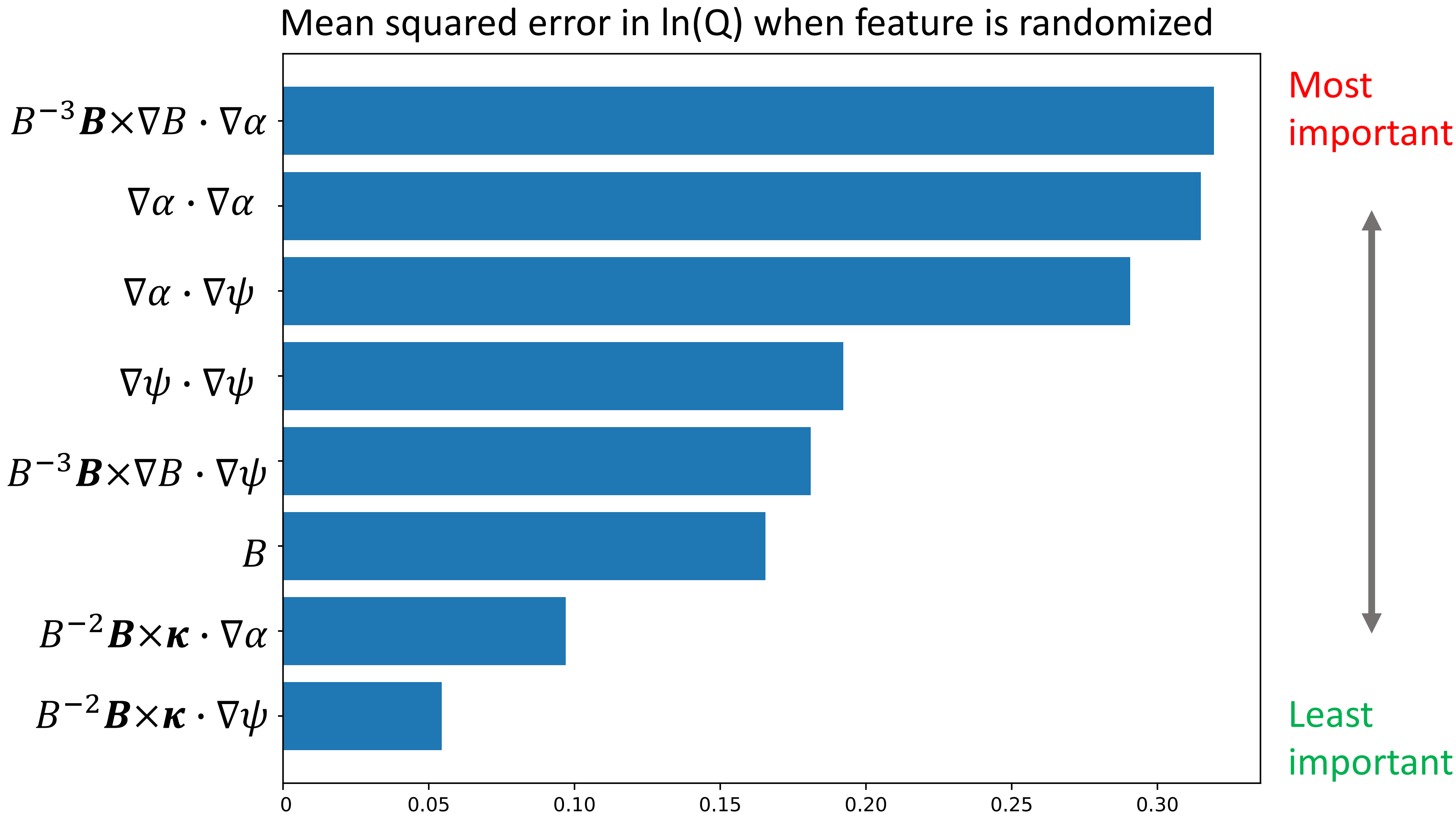
All using the set of 20 features selected via LightGBM

Importance of a feature can be measured by randomly permuting it

- For each of the raw features, randomly swap it with another configuration.
- To ensure $k_\perp \geq 0$, also cap $|\nabla\alpha \cdot \nabla\psi|$ at $\sqrt{(\nabla\psi \cdot \nabla\psi)(\nabla\alpha \cdot \nabla\alpha)}$.
- No machine learning model – Just re-run GX with the altered geometry inputs.



Importance of a feature can be measured by randomly permuting it



Curvature drift is the least important, while ∇B drift is the most important!

Can we understand this physically from the gyrokinetic equation?

There are many extensions possible

- Try larger sets of possible features
- Understand physically why later features affect turbulence.
- Understand why curvature drift is $\ll$ important than ∇B drift.
- Saliency maps to understand the features learned by the CNNs.
- Symbolic regression.
- Kolmogorov-Arnold Networks.
- Expand to multiple values of nfp, aspect ratio, gradients.
- Kinetic electrons.