Image credit: Marguerite Tonjes

Efficient interpolation and practical observables

Nick Smith
LHC EFT Working Group
2 Dec 2022

Goals

- Present scaling matrix approach for linear-in-amplitude EFT expansion
- Demonstrate useful EFT fit speedup opportunity
- Suggest a systematic approach to splitting existing binning (such as STXS) towards *better* sensitivity

Scaling matrix

- Linear-in-amplitude parameterization of a differential cross section
  - $\frac{d\sigma}{d\mathbf{x}}(\mathbf{x},\mathbf{c}) = |\, c_0 a_0(\mathbf{x}) + c_1 a_1(\mathbf{x}) + \cdots |^2 = \mathbf{c}^\top \mathbf{a}^*(\mathbf{x}) \mathbf{a}^\top(\mathbf{x}) \mathbf{c}$
  - For parameter vector **c**, amplitude vector **a** as a function of phase space point **x**
  - This can be rewritten to be real-valued: $\mathbf{A}(\mathbf{x}) = (\mathbf{a}^* \mathbf{a}^\top + \mathbf{a}\mathbf{a}^{*\top})/2$
- So it is a matrix norm:
  - $\frac{d\sigma}{d\mathbf{x}}(\mathbf{x},\mathbf{c}) = \mathbf{c}^\top \mathbf{A}(\mathbf{x}) \mathbf{c}$
  - **A** is computed per event with usual tricks (MG reweighing) and is rank <= 2
- Integrate over some region to get a bin yield *y*
  - $y = \mathbf{c}^\top \left(L \int_X \epsilon(\mathbf{x}) \mathbf{A}(\mathbf{x}) d\mathbf{x} \right) \mathbf{c} = \mathbf{c}^\top \mathbf{M} \mathbf{c}$
  - For luminosity *L*, efficiency ϵ, neglecting smearing (does not change picture)
  - **A** continuous in **x**, so rank(**M**) will grow as region grows

What good is that?

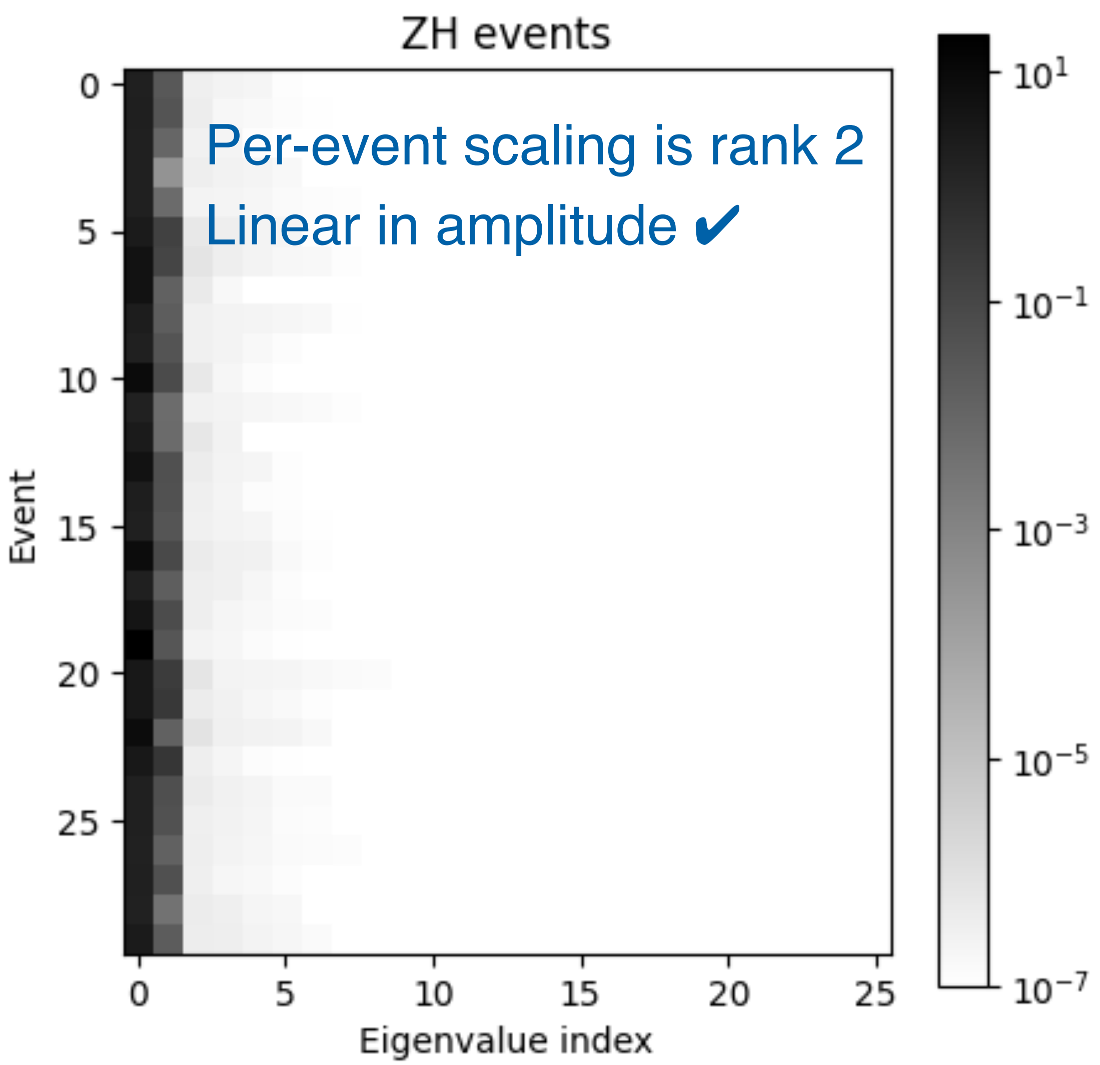
- A linear-in-amplitude dim-6 SMEFT parameterization then becomes:

$$y = y_{SM}\left(1 + \sum_i c_i A_i + \sum_{i \leq j} c_i c_j B_{ij}\right) \quad \Leftrightarrow \quad y = y_{SM}\left(\begin{bmatrix} 1 \\ c_1 \\ c_2 \\ \vdots \end{bmatrix}^\top \begin{bmatrix} 1 & A_1/2 & A_2/2 & \cdots \\ A_1/2 & B_{11} & B_{12}/2 & \cdots \\ A_2/2 & B_{12}/2 & B_{22} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} 1 \\ c_1 \\ c_2 \\ \vdots \end{bmatrix}\right)$$

  - Easy to check pos-def
- When rank(**M**) < n (# coefficients) we can speed up interpolation in fits
  - Precompute eigendecomposition and keep top k rows $\mathbf{M}^{1/2}_{:k} = (\mathbf{Q}\boldsymbol{\Lambda}^{1/2})_{:k}$ per bin
  - Then $y \approx |\mathbf{M}^{1/2}_{:k}\mathbf{c}|^2$, **which is O(kn) instead of O(n²)**

Quick check

- Simulate Z(ee,vv)H production with SMEFTsim
  - With linearized propagator correction, patch to restrict all coupling orders <=2
    - Thanks to M. Knight
  - Calculate per-event scaling matrix for 25 NP operators with nonzero effect (+1 for SM)
    - $\Im c_{bB}, \Re c_{bB}, \Im c_{bH}, \Re c_{bH}, \Im c_{bW}, \Re c_{bW}, c_{HB}, c_{H\Box}, c_{Hb}, c_{H\tilde{B}}, c_{Hd}, c_{HD}, c_{He}, c_{Hq}^{(1)}, c_{Hq}^{(3)}, c_{Hl}^{(1)}, c_{Hl}^{(3)}, c_{HQ}^{(1)}, c_{HQ}^{(3)}, c_{Hu}, c_{HW}, c_{HWB}, c_{H\tilde{W}B}, c_{H\tilde{W}}, c'_{ll}$

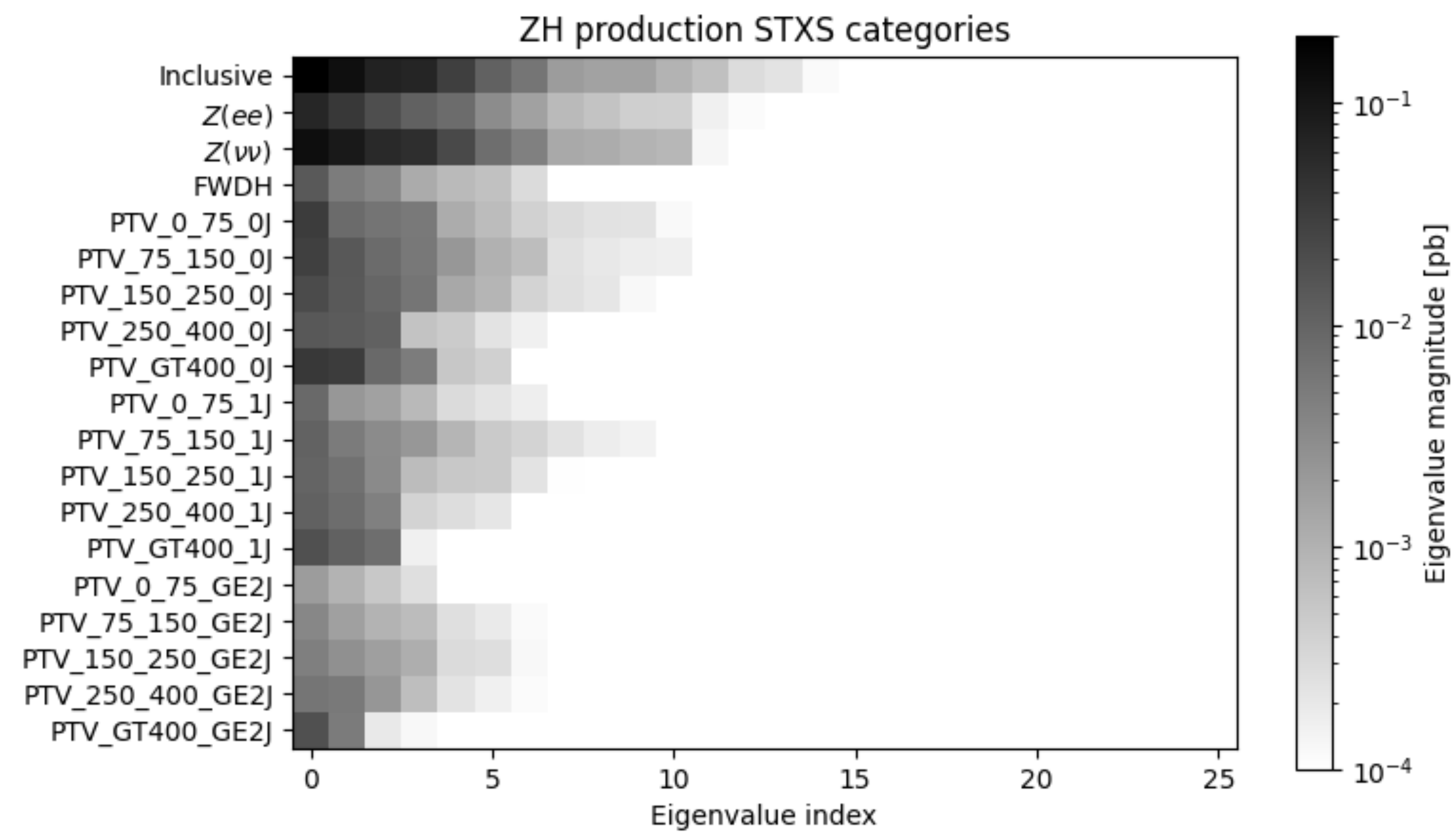


Per-event scaling is rank 2

Linear in amplitude ✔

Sum events in STXS bins

Rank goes up with inclusivity, still << 26



Fermilab

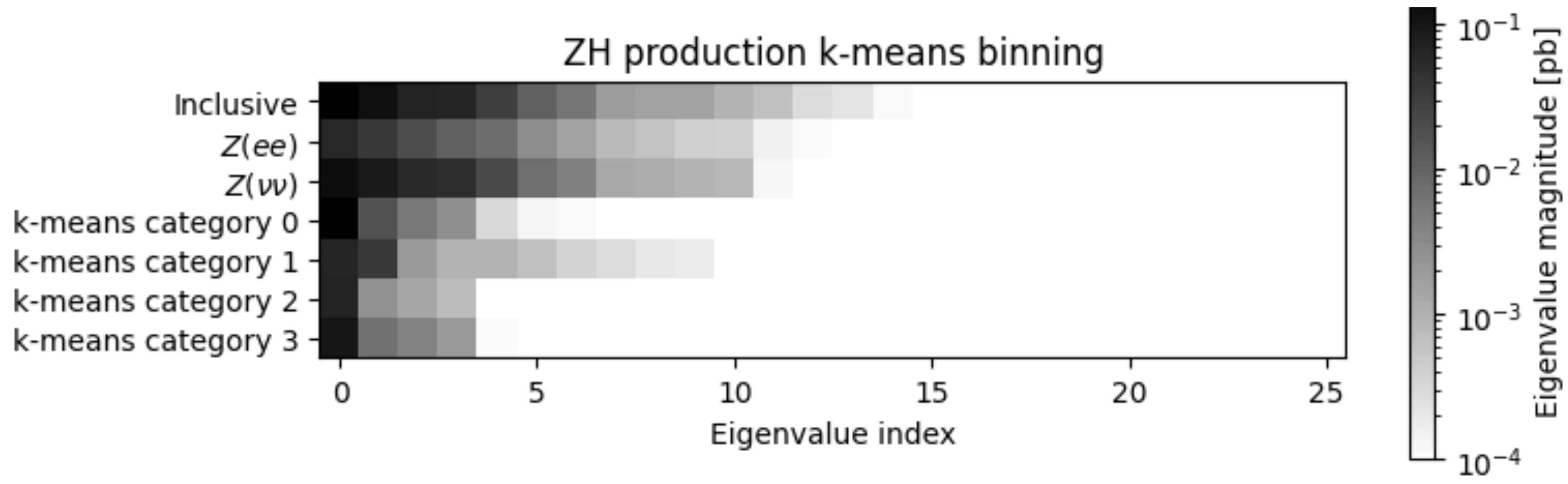
What this looks like in a fit

- The binned likelihood is approximately
  - $$-2\ln L(\mathbf{c}) \approx \sum_b \frac{(y_{b,obs} - \mathbf{c}^\top \mathbf{M}_b \mathbf{c})^2}{\sigma_b^2}$$
- The Fisher information matrix is then*
  - $$\mathbf{I}(\mathbf{c}) = \sum_b \frac{4}{\sigma_b^2}(\mathbf{M}_b\mathbf{c})(\mathbf{M}_b\mathbf{c})^\top$$
  - Note: not constant, unlike for linear cross section expansion
  - Of course, we are really interested in the 1:n submatrix since $c_0 \equiv 1$ (SM)
- rank(**M**) is telling us how much coefficient-dependence is projected out
  - Split bin into sub-bins of lower rank that are “orthogonal” ➜ more information (but σ up)

* Assuming regularity conditions, e.g. domain of y independent of **c**

Squeezing information out

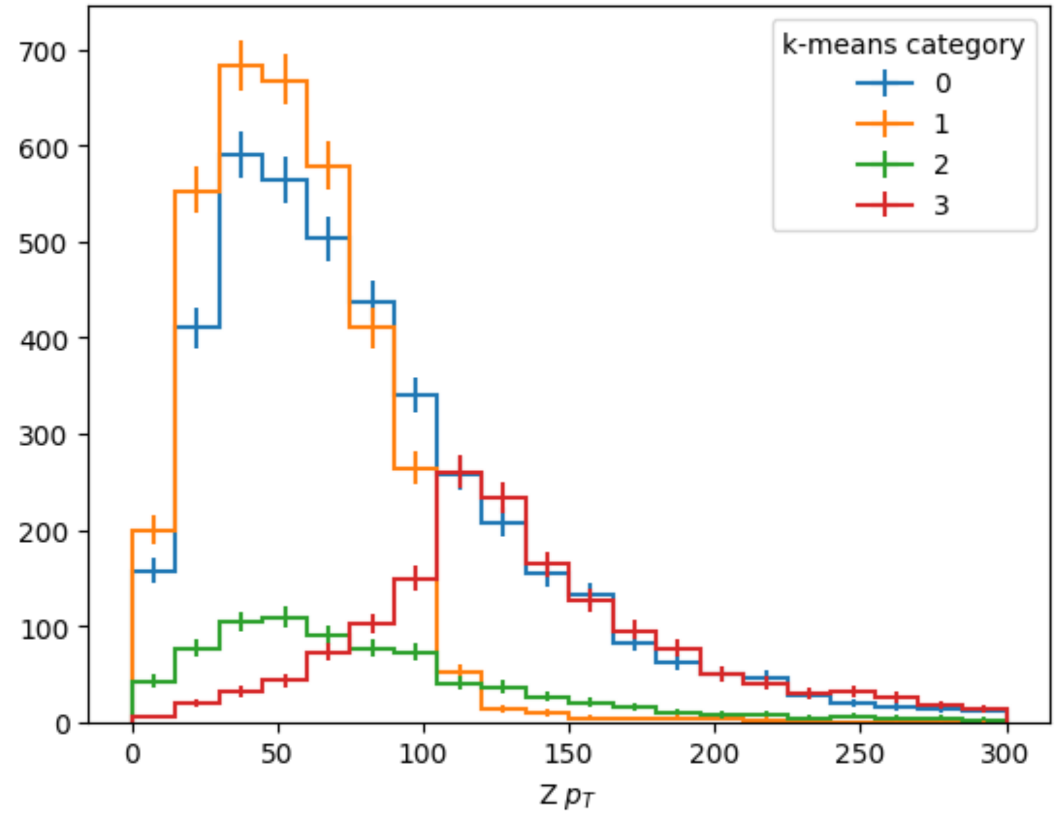
- Splitting a bin: need definition of "orthogonal"
  - At SM expectation ($\mathbf{c} = [1,0,\ldots]$), find vectors $\mathbf{M}_{1:n,0}$ with low inner product
  - Can also just take Frobenius product
    - $\langle M_1, M_2 \rangle_F = \mathrm{tr}(M_1^\top M_2)$
    - Dominated by interference terms anyway ($1/\Lambda^2$)
    - Value in optimizing $\mathbf{I}$ for BSM points?
- Cluster same ZH events using Frobenius product as distance metric
  - Use direction rather than distance, i.e. inner product / sqrt(norms)
  - Simple k-means for now
- Categories are lower rank!

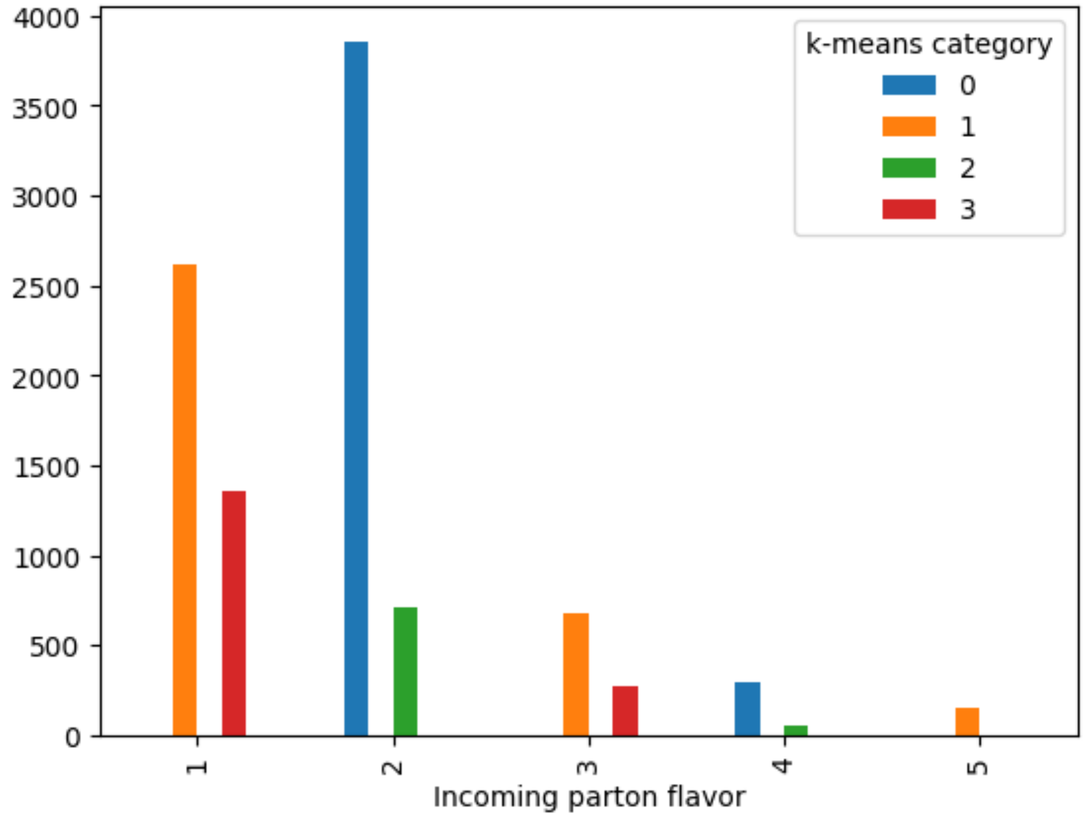


Fermilab

Towards observables

- Great, but scaling matrices are not observable
- What can identify clusters?
  - Z pT 😀
  - Incoming parton flavor 🙁
- Next step: train NN on observables to separate
  - Supervised on cluster? Contrastive loss? Information matrix as loss (a la INFERNO)?
    - Early tests with the latter show promise





Fermilab

Summary

- I think this is a new way to look at linear-in-amplitude EFT expansion
  - Unlocks a lot of linear algebra stuff
- Alternative to optimal observables approach
  - Start from inclusive binning and divide up
  - vs. learning unbinned likelihood ratio
- If all that fails, at least it allows us to run fits faster