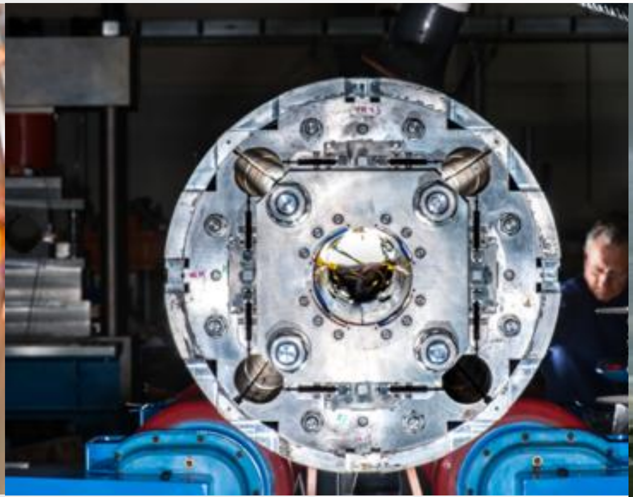
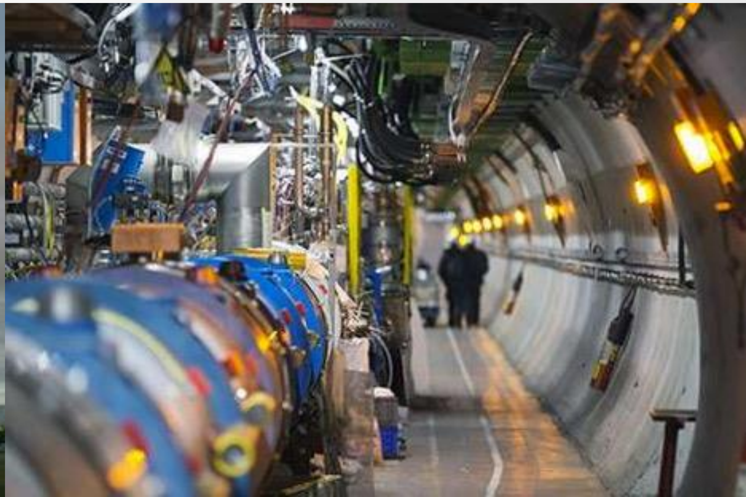
TREE BOOSTING (+NEURAL NETWORKS) FOR EFT ANALYSES

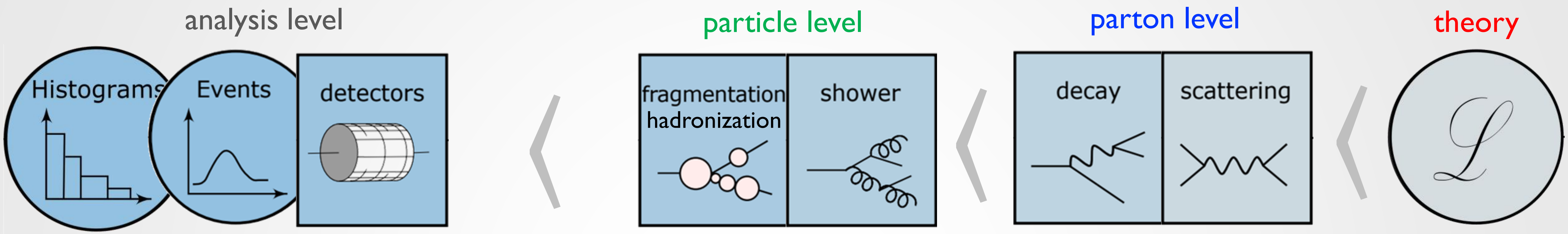
R. Schöfbeck (HEPHY Vienna), Oct.. 24th, 2023, Area 3 meeting



A CONDITIONAL SEQUENCE

adapted from arXiv:2211.01421



$$p(x_{\text{det}}|\theta) = \int \mathrm{d}z_{\text{ptl}} \int \mathrm{d}z_{\text{p}} [\cdots]\; p(x_{\text{det}}|z_{\text{ptl}}) \qquad p(z_{\text{ptl}}|z_{\text{p}}) \qquad p(z_{\text{p}}|\theta)$$

ML facilitates this inversion by exploiting that simulation samples

$$x_{\text{det}}, z_{\text{ptl}}, z_{\text{p}} \sim p(x_{\text{det}}, z_{\text{ptl}}, z_{\text{p}}|\theta)$$

"Simulation based inference"

1. Generators run in 'forward mode'
2. Pick up uncertainties
   $p(z_{\text{ptl}}|z_{\text{p}}, \nu_{\text{th.}})$
   $p(x_{\text{det}}|z_{\text{ptl}}, \nu_{\text{exp.}})$

$$\frac{1}{\sigma_\theta}\frac{\mathrm{d}\sigma_\theta}{\mathrm{d}z_p} = p(z_{\text{p}}|\theta)$$

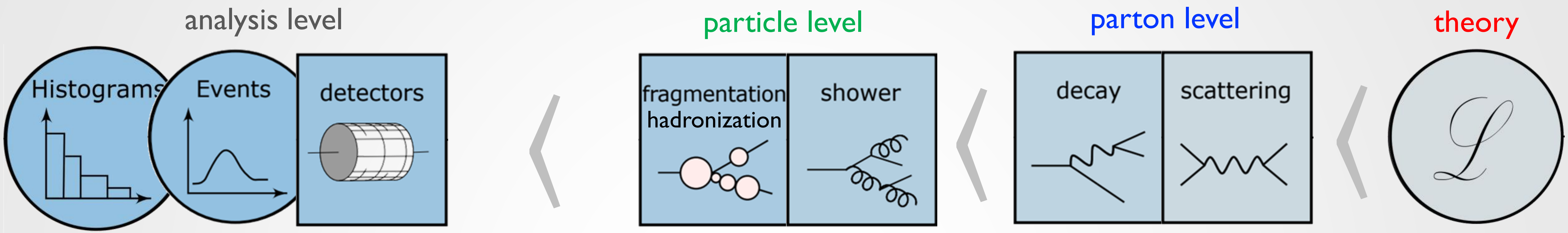
parton-level differential cross section ~ pdf

$\cancel{p(\theta)}$

θ NOT stochastic; Frequentist

THE LIKELIHOOD RATIO TRICK

adapted from arXiv:2211.01421



Event classification $\boldsymbol{\theta} \to$ isSig $\in \{0,1\}$

$$L = \langle f(x)^2 \rangle_{\mathrm{isSig}=1} + \langle (1 - f(x))^2 \rangle_{\mathrm{isSig}=0} = \sum_{\mathrm{isSig}} \int \mathrm{d}x \cdots$$

$$f^*(x) = \frac{1}{1 + \frac{p(x|\mathrm{isSig}=1)}{p(x|\mathrm{isSig}=0)}}$$

Learn LR by classification;
"Likelihood ratio trick"
achieve NP optimality (for x-sec)

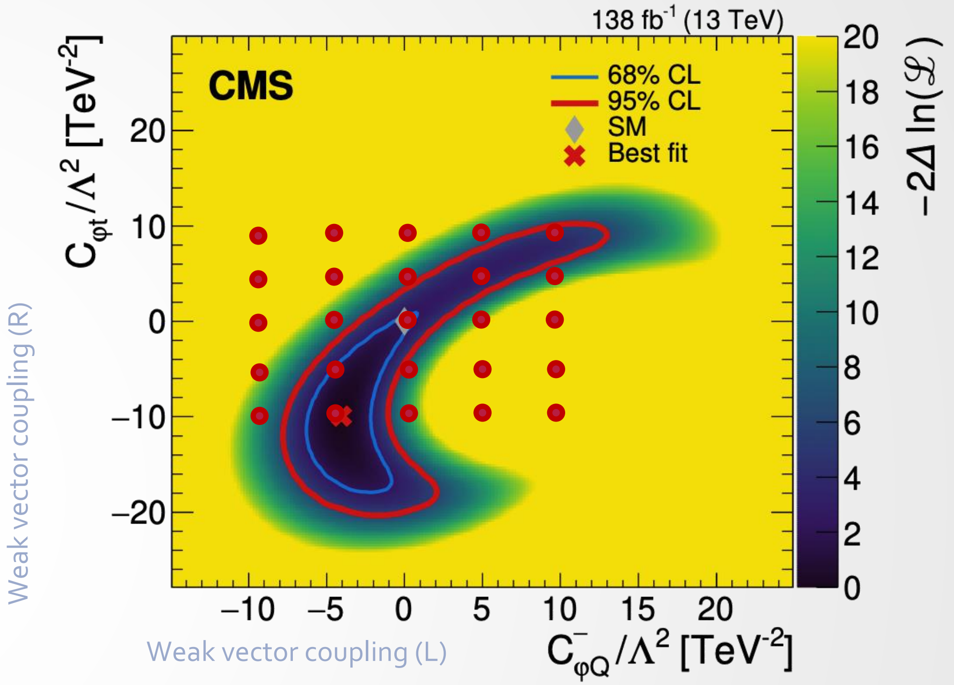
CAN WE JUST LEARN EFT EFFECTS "ON AVERAGE"?

$$L = \sum_{\theta \in \mathcal{B}} \int \mathrm{d}\boldsymbol{x} \left(p(\boldsymbol{x}|\boldsymbol{\theta}) \hat{f}(\boldsymbol{x})^2 + p(\boldsymbol{x}|\mathrm{SM})(1 - \hat{f}(\boldsymbol{x}))^2 \right)$$

$\boldsymbol{\theta}$ - ignorant

mixing signals &
case dependent mixes

$$f^*(\boldsymbol{x}) = \frac{1}{1 + r_{\mathcal{B}}(\boldsymbol{x})} \qquad r_{\mathcal{B}}(\boldsymbol{x}) = \frac{\frac{1}{|\mathcal{B}|} \sum_{\boldsymbol{\theta} \in \mathcal{B}} p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x}|\mathrm{SM})}$$



[TOP-21-001]

- We can try to learn EFT effects "on average"
- Sending 'mixed signals' to the loss function
  - Averages the training data set - suboptimal when linear effects dominate
  - Classifier does not reflect knowledge on the $\boldsymbol{\theta}$-dependence

1. Back to the drawing board & inject $\boldsymbol{\theta}$ polynomial SMEFT dependence in estimator.
2. Exploit the fact that SMEFT predictions can (optionally) be weighted: Only need one training data set

EXPLOITING SMEFT REWEIGHTING

$$L = \sum_{\theta\in\mathcal{B}} \left(\langle \hat{f}(x;\theta)^2 \rangle_\theta + \langle (1-\hat{f}(x;\theta))^2 \rangle_{\mathrm{SM}} \right)$$

$\boldsymbol{\theta}$ - *aware* *EFT sample* *SM sample*

We start with SM and BSM samples

$$= \sum_{\theta\in\mathcal{B}} \int \mathrm{d}x\,\mathrm{d}z \, \left(p(x,z|\theta)\hat{f}(x;\theta)^2 + p(x,z|\mathrm{SM})(1-\hat{f}(x;\theta))^2 \right)$$

Let's write this under one integral
z … latent space

SM sample

$$= \sum_{\theta\in\mathcal{B}} \int \mathrm{d}x\,\mathrm{d}z \, p(x,z|\mathrm{SM}) \left(r(x,z|\theta)\hat{f}(x;\theta)^2 + (1-\hat{f}(x;\theta))^2 \right)$$

"joint" likelihood ratio

… and use just one sample
& joint likelihood ratio

$$r = \frac{p(x_{\mathrm{det}},\cdots,z_{\mathrm{ptl}},\cdots,z_{\mathrm{p}}|\theta)}{p(x_{\mathrm{det}},\cdots,z_{\mathrm{ptl}},\cdots,z_{\mathrm{p}}|\mathrm{SM})} = \frac{p(x_{\mathrm{det}}|z_{\mathrm{ptl}})\cdots p(z_{\mathrm{ptl}}|z_{\mathrm{p}})\cdots p(z_{\mathrm{p}}|\theta)}{p(x_{\mathrm{det}}|z_{\mathrm{ptl}})\cdots p(z_{\mathrm{ptl}}|z_{\mathrm{p}})\cdots p(z_{\mathrm{p}}|\mathrm{SM})} = \frac{p(z_{\mathrm{p}}|\theta)}{p(z_{\mathrm{p}}|\mathrm{SM})} \sim \frac{|\mathcal{M}(z_{\mathrm{p}},\theta)|^2}{|\mathcal{M}(z_{\mathrm{p}},\mathrm{SM})|^2} = w_i(\boldsymbol{\theta})$$

Change in likelihood of simulated observation x with latent "history" z going from "SM" to $\boldsymbol{\theta}$

staged simulation in forward mode: Intractable factors cancel

re-calcuable theory prediction

weighted simulation

K. Cranmer , J. Pavez , and G. Louppe [1506.02169]
J. Brehmer, K. Cranmer, G. Louppe, J. Pavez [1805.00013]
[1805.00020]
“Madminer” [1805.12244]
J. Brehmer, F. Kling, I. Espejo, K. Cranmer [1907.10621]

PARAMETRIZED CLASSIFIERS

$$L = \sum_{\theta \in \mathcal{B}} \int \mathrm{d}x\, \mathrm{d}z\, p(x,z|\mathrm{SM}) \left(r(x,z|\theta)\hat{f}(x;\theta)^2 + (1-\hat{f}(x;\theta))^2 \right)$$

MSE or cross entropy

$$\hat{f}(x;\theta) = \frac{1}{1+\hat{R}(x;\theta)}$$

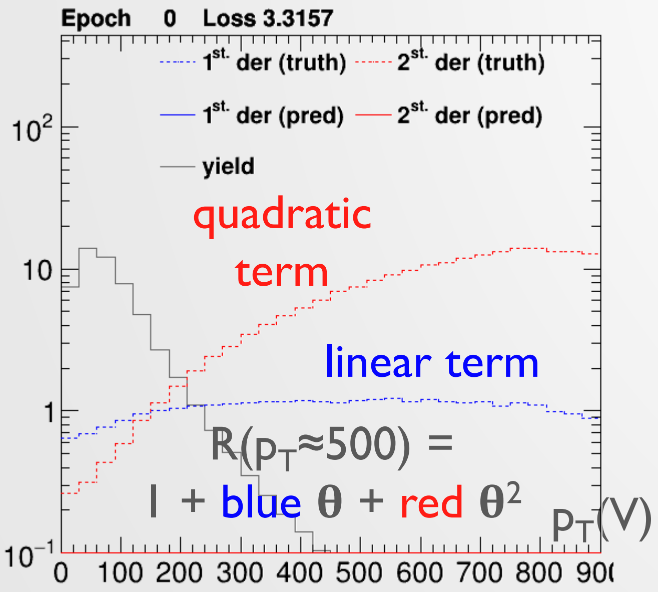
invert likelihood trick

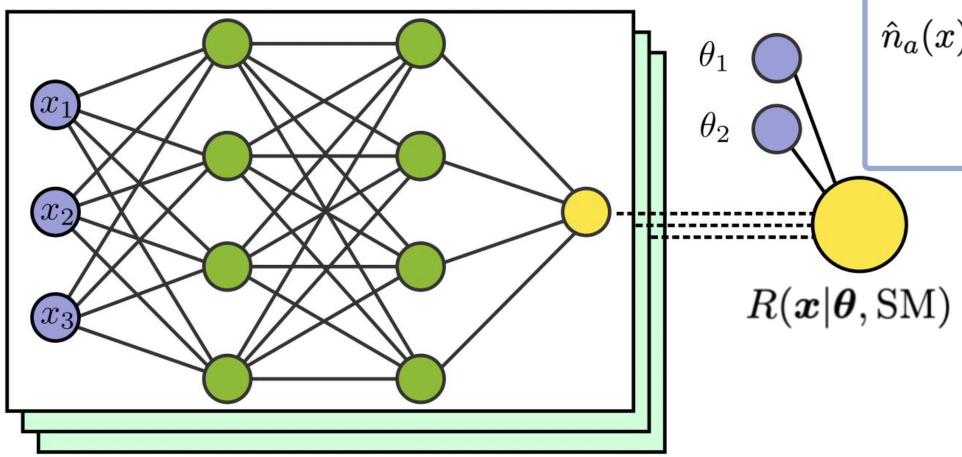
Similar to S. Chen, A. Glioti, G. Panico, A. Wulzer JHEP 05 (2021) 247

arXiv:2308.05704

insert model knowledge: fit universal coefficient functions

$$\hat{R}(x;\theta) = \left(1+\sum_a \theta_a \hat{n}_a(x)\right)^2 + \sum_a \left(\sum_{b\geq a} \theta_b \hat{n}_{ab}(x)\right)^2$$





$$\hat{n}_a(x) \rightarrow \left.\frac{\partial_a p(x|\theta)}{p(x|\mathrm{SM})}\right|_{\theta=\mathrm{SM}} = \left.\frac{\partial_a \int \mathrm{d}z\, p(x,z|\theta)}{\int \mathrm{d}z\, p(x,z|\mathrm{SM})}\right|_{\theta=\mathrm{SM}}$$

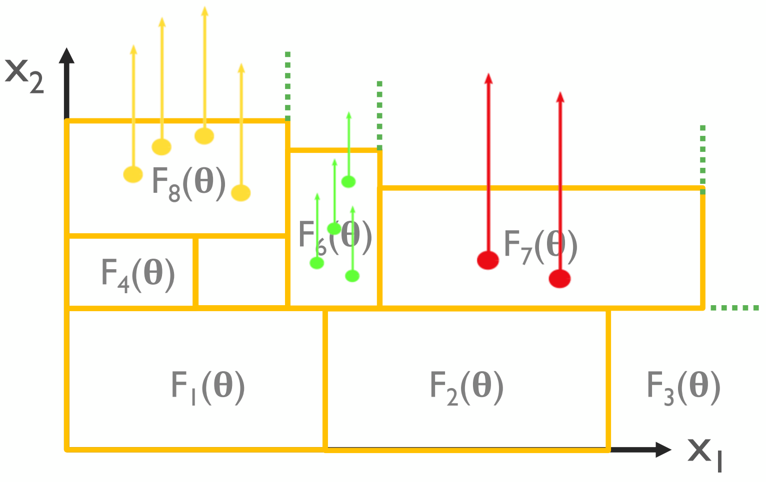
Integrates latent space!

Why would you want to use trees instead?

A SIMPLE TREE ALGORITHM

[arXiv:2107.10859, arXiv:2205:12976]

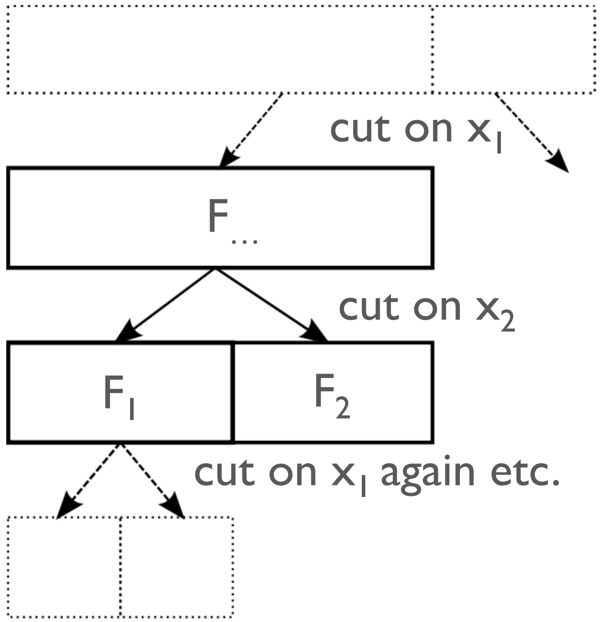
phase-space partitioning



A simple tree

$$\underbrace{\underbrace{\hat{F}(\boldsymbol{x},\boldsymbol{\theta}) = \sum_{j\in\mathcal{J}} \overbrace{\mathbb{1}_j(\boldsymbol{x})}^{\text{index-function (non-linearity)}}}_{\text{phase space partitioning J}} \underbrace{F_j(\boldsymbol{\theta})}_{\text{prediction } F_j}}_{\text{need to solve for partitioning J and } \{F_j\}}$$

training phase:
e.g. "CART" algo



- A tree is a hierarchical phase-space partitioning ($\mathcal{J}$)
  - the novelty in the Boosted Information Tree is that we associate each region j with a polynomial $F_j(\boldsymbol{\theta})$
  - Note: A tree algorithm can have an arbitrarily complicated predictive function; here it is a SMEFT polynomial
- Fitting tree: Optimize "node split positions" on some loss. Trained (e.g. greedily) on the *ensemble*.

PARAMETRIC TREES FOR SMEFT

[arXiv:2107.10859, arXiv:2205:12976]

Want to regress in r, exploiting its the polynomial $\boldsymbol{\theta}$ dependence

$$r(x, z|\theta) = \frac{\mathrm{d}\sigma(\boldsymbol{x}, \boldsymbol{\theta})/\mathrm{d}\boldsymbol{x}}{\mathrm{d}\sigma(\boldsymbol{x}, \mathrm{SM})/\mathrm{d}\boldsymbol{x}}$$

→ will allow to compute the optimal LLR test statistic $q(\mathcal{D})$

$$L = \sum_{\boldsymbol{\theta}\in\mathcal{B}} \int \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{z}\, p(\boldsymbol{x}, \boldsymbol{z}|\mathrm{SM}) \left(r(x, z|\theta) - \hat{F}(\boldsymbol{x}, \boldsymbol{\theta}) \right)^2$$

Tree ansatz

$$\hat{F}(\boldsymbol{x}, \boldsymbol{\theta}) = \underbrace{\sum_{j\in\mathcal{J}} \mathbb{1}_j(\boldsymbol{x})}_{\text{find optimal partitioning}} \underbrace{F_j(\boldsymbol{\theta})}_{\text{find optimal predictor}}$$

$F_j(\boldsymbol{\theta})$ polynomial with const. coeff. (per node)

Remove DOF from predictor:

$$F_j(\boldsymbol{\theta}) = \frac{\sum_{i\in j} w_i(\boldsymbol{\theta})}{\sum_{i\in j} w_i(\boldsymbol{\theta}_0)}$$

sum up event weights within node & divide

No trainable parameters in the predictor (Regression by means of classification)

Solve for optimal partitioning with greedy CART algorithm

$$L = -\sum_{\boldsymbol{\theta}\in\mathcal{B}} \sum_{j\in\mathcal{J}} \frac{w_j^2(\boldsymbol{\theta})}{w_j(\boldsymbol{\theta}_0)} = -\sum_{j\in\mathcal{J}} \sum_{\boldsymbol{\theta}\in\mathcal{B}} \theta^a \theta^b I_{ab}^{(j)} + \mathcal{O}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^3$$

We're optimizing the Fisher information!

We'll find an optimized tree.
→ boost

CONCRETE SOLUTION: TREE BOOSTING

[arXiv:2107.10859, arXiv:2205:12976]

- Boosting: Fit model iteratively to pseudo-residuals of the preceding iteration with learning rate η
- Ansatz :

$$\hat{F}^{(b)}(\boldsymbol{x},\boldsymbol{\theta}) = \underbrace{\hat{f}(\boldsymbol{x},\boldsymbol{\theta})}_{\text{current iteration}} + \eta \underbrace{\hat{F}^{(b-1)}(\boldsymbol{x},\boldsymbol{\theta})}_{\text{previous iteration}}$$

- Insert into the loss function:

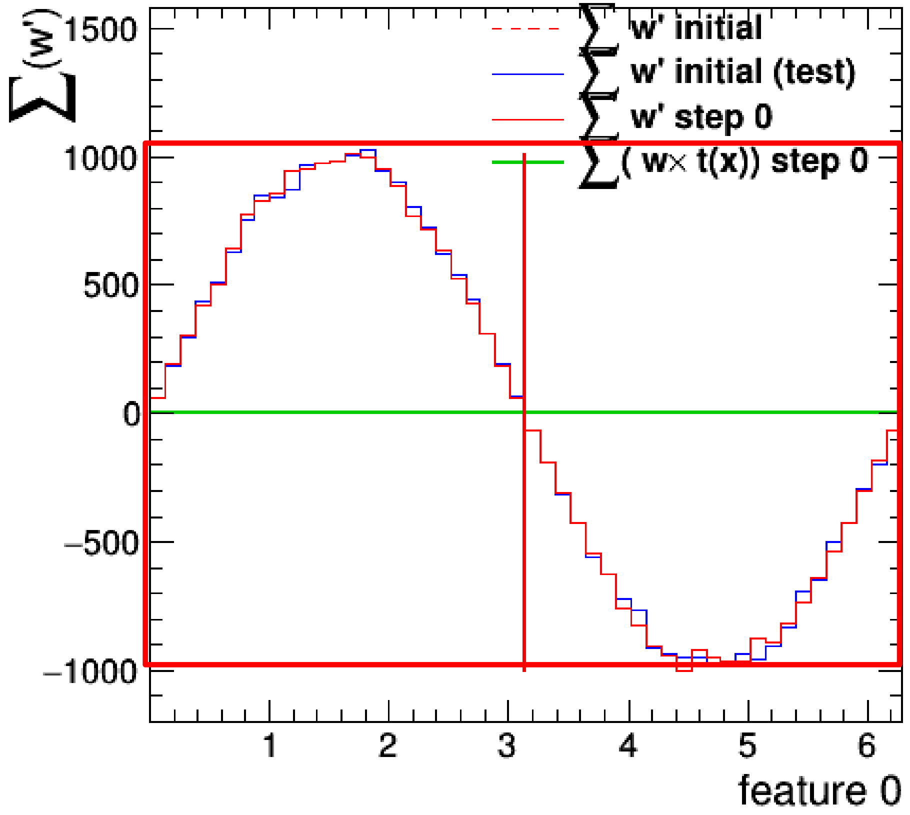
$$\underbrace{L[\hat{f}^{(b)}]}_{\text{current iteration}} = \sum_{\theta\in\mathcal{B}} \int \mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{z}\, p(\boldsymbol{x},\boldsymbol{z}|\mathrm{SM}) \Big(\underbrace{r(x,z|\theta) - \eta \underbrace{\hat{F}^{(b-1)}(\boldsymbol{x},\boldsymbol{\theta})}_{\text{previous iteration}}}_{\text{pseudo-residual, amounting to event-leve reweighting}} - \underbrace{\hat{f}^{(b)}(\boldsymbol{x},\boldsymbol{\theta})}_{\text{current iteration}} \Big)^2$$

$$w_i^{(b)}(\boldsymbol{\theta}) \quad \to \quad w_i^{(b-1)}(\boldsymbol{\theta}) - \eta\, w_i^{(b-1)}(\boldsymbol{\theta}_0)\hat{F}^{(b-1)}(\boldsymbol{x}_i,\boldsymbol{\theta})$$

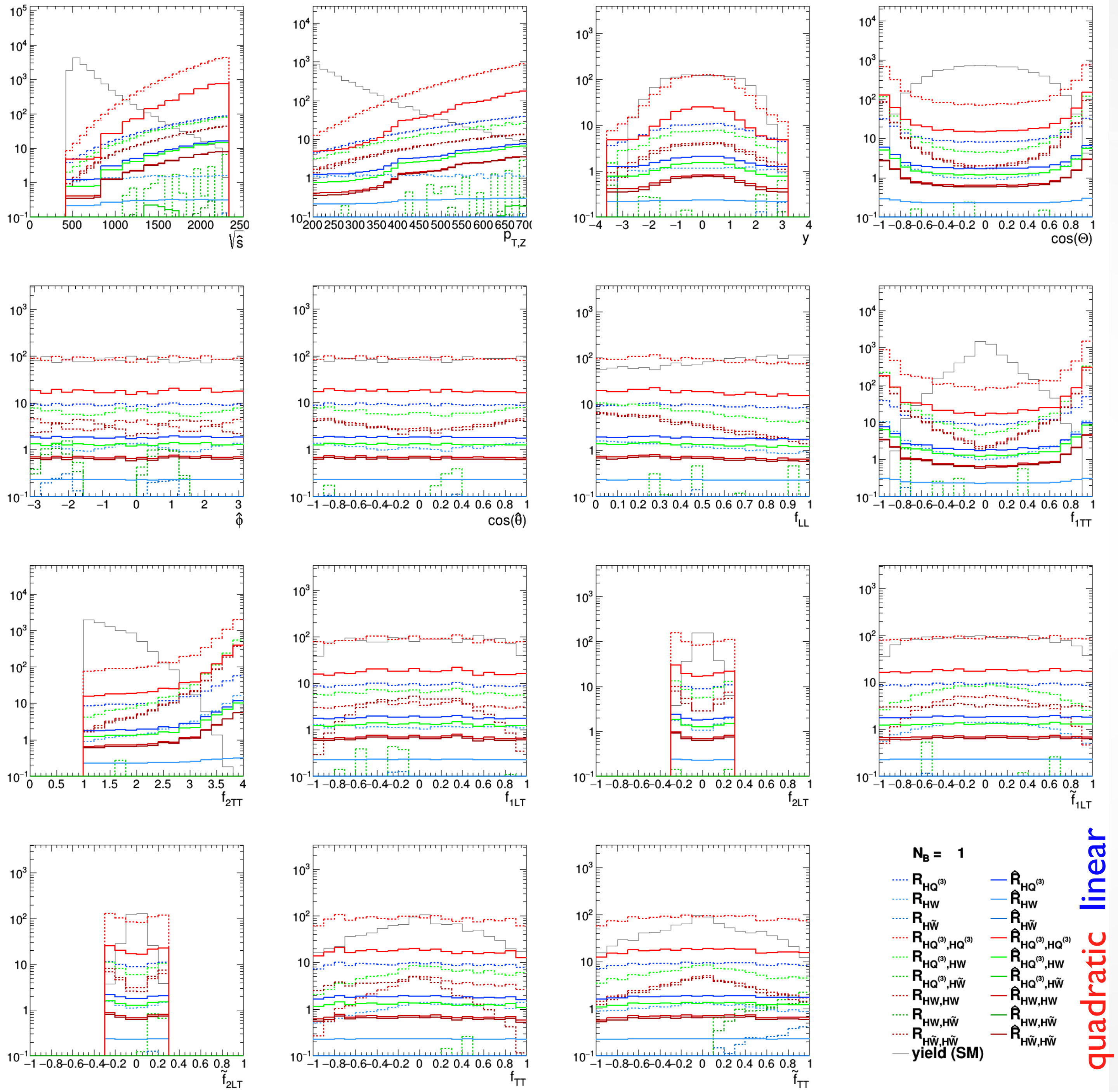
.... perform this iteratively
"Boosted Information Tree"

1D TOY EXAMPLE

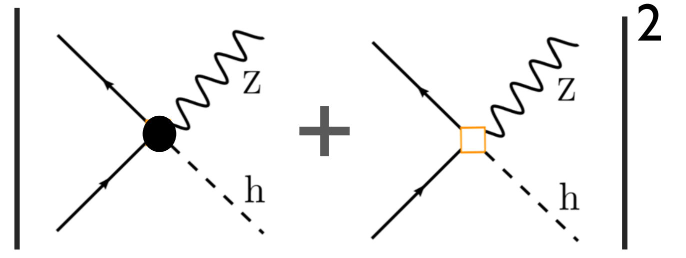
- pdf(x|$\boldsymbol{\theta}$) = 1/(2$\boldsymbol{\pi}$) - $\boldsymbol{\theta}$ Sin(x)
- We predict the first derivative of the pdf wrt. to the parameter



GIF animation not showing in pdf



- Realistic case: model of the ZH process
- "Boosted Information Tree (BIT)"
  - 3 WC, 9 DOF, 500k events, ZH
  - 200 trees, D=5, 9 minutes of training
  - also more realistic study, including backgrounds [2107.10859], [2205.12976]
- Learning coefficient functions to compute parametrized optimal oberables

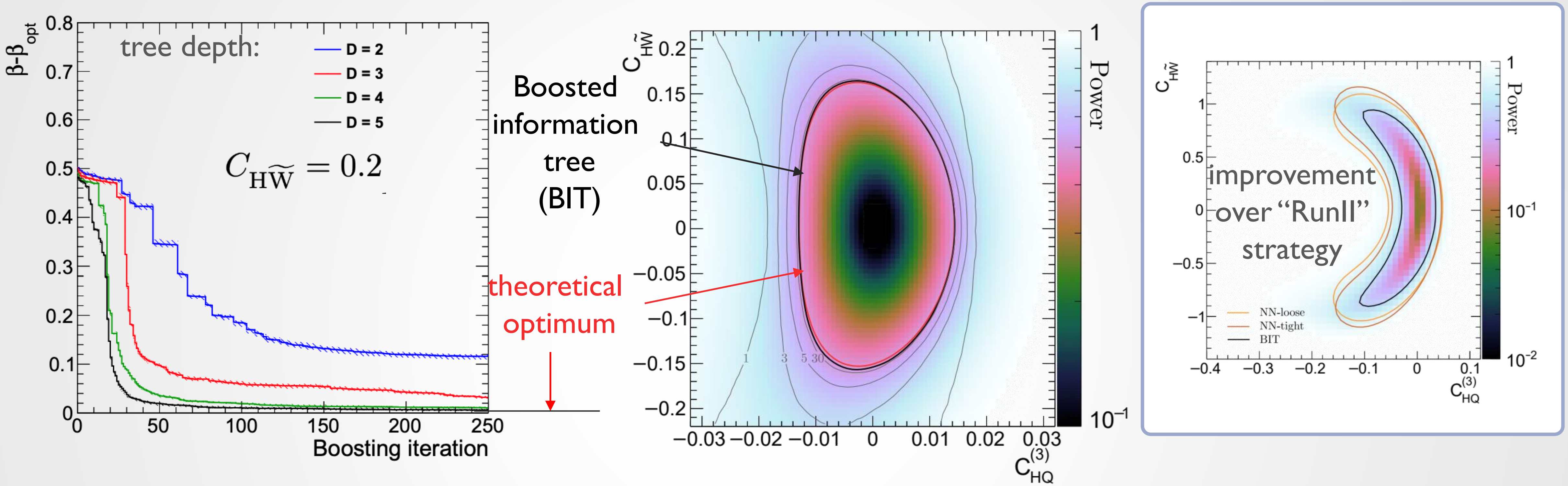


$$= \sigma^{\mathrm{SM}} + \frac{\theta}{\Lambda^2}\sigma^{\mathrm{int}} + \frac{\theta^2}{\Lambda^4}\sigma^{\mathrm{quad}}$$

$\rightarrow$ parametrized $q_{\boldsymbol{\theta}}(\mathcal{D})$

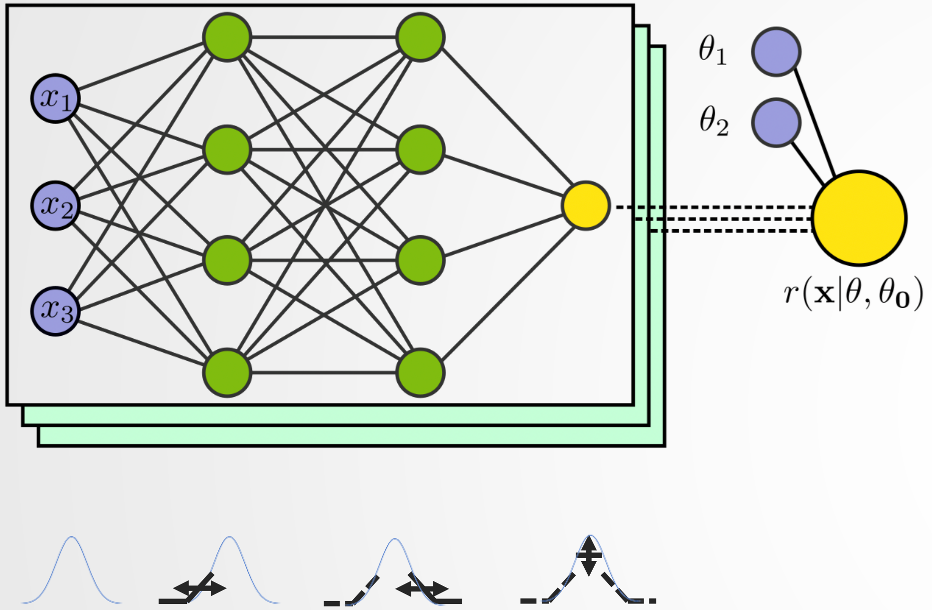
OPTIMALITY IN TEST CASES

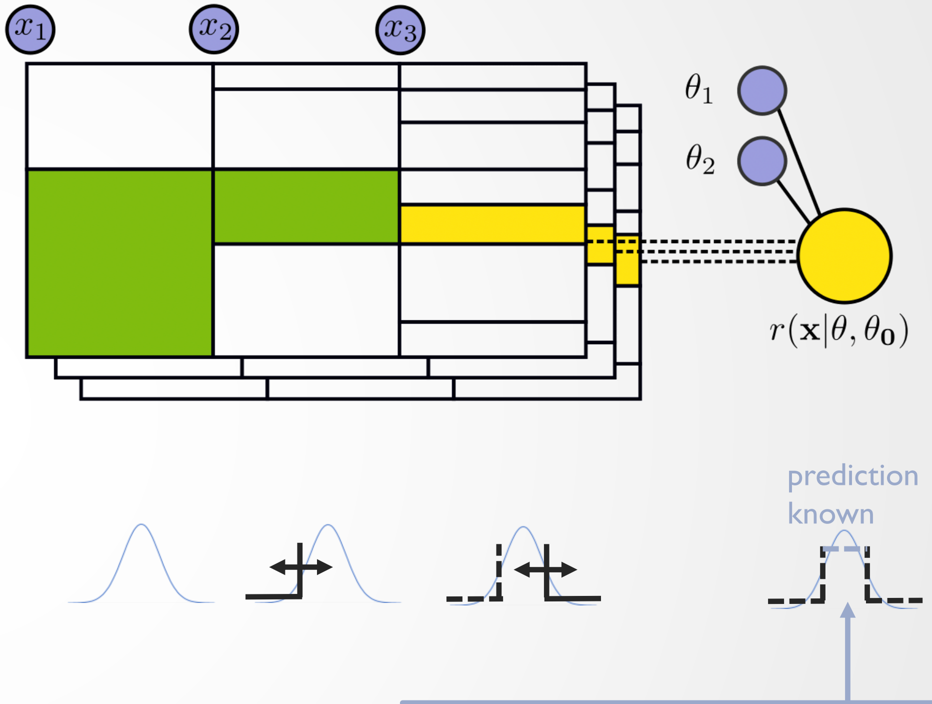
[arXiv:2107.10859, arXiv:2205:12976]



- Obtain parametrized classifiers with 20-40% improvements in 2D toy cases (NOT marginalized!)
- No free lunch – Analysis dependent choices are needed
  - Binned analysis: variable binning → background estimation is CPU intensive
  - Systematics treatment for unbinned analyses (beyond Higgs $M_{4\ell}$) less far developed
- Is it all worth it in higher dimensions? Yes! More examples: [ML4EFT]; full list of references in backup

NETWORKS VS. TREES – WHAT IS THE BIG DEAL?





$$F_j(\boldsymbol{\theta}) = \frac{\sum_{i\in j} w_i(\boldsymbol{\theta})}{\sum_{i\in j} w_i(\boldsymbol{\theta}_0)} = \frac{\int \mathrm{d}z \frac{\mathrm{d}\sigma_{\boldsymbol{\theta}}}{\mathrm{d}(x,z)}}{\int \mathrm{d}z \frac{\mathrm{d}\sigma_{\mathrm{SM}}}{\mathrm{d}(x,z)}}$$

- Given a phase space region with EFT dependence: NN must select & predict
- In the Boosted Information Tree, the weak learner only selects
  - The prediction (F_j) is computed from the boxed events → integrates latent space
  - The regression problem is solved with computational complexity of classification
    - Speed advantage at high operator dimensions!

SUMMARY!

- Various algorithms predict SMEFT dependence, mostly capitalizing on weighted simulation
  - Provide NP-optimal observables for hypothesis tests
- Trees eliminate the latent space integration and suitable to weight-based SMEFT predictions
- Parametrized classifiers "ask" for unbinned analysis

[SWAN project]

Get in touch with us for an in-person walkthrough

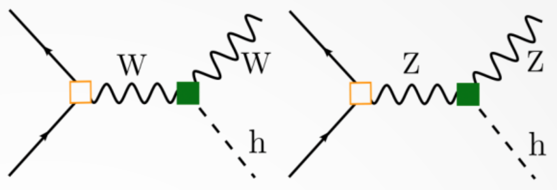
```
from MultiBoostedInformationTree import MultiBoostedInformationTree

bit = MultiBoostedInformationTree(
        training_features     = training_features,
        training_weights      = training_weights,
        base_points           = base_points,
        feature_names         = model.feature_names,
        **model.multi_bit_cfg
            )
bit.boost()
test_predictions = bit.vectorized_predict(test_features)
```

+ 500 lines for figs

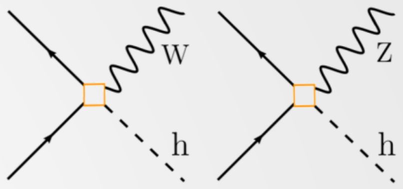
GOALS FOR MACHINE-LEARNING *OF* EFT

[EPJC 81 (2021) 178]

- SMEFT effects can be
  1. in the tails of the distributions because, e.g. 4-point functions grow with energy
  2. in angular observables & correlations, sometimes encoding CP-violating effects
     - "interference resurrection" PLB 2017 11 086
       "method of moments" JHEP 06 (2021) 031
     - Enhance / single out the linear term
       - Up to triple-angular correlations, x5-10 boost in sensitivity
  3. on top of "kinematically complex" backgrounds
     - Def: Usually amenable to classification MVAs
     - Unify the training target with classification



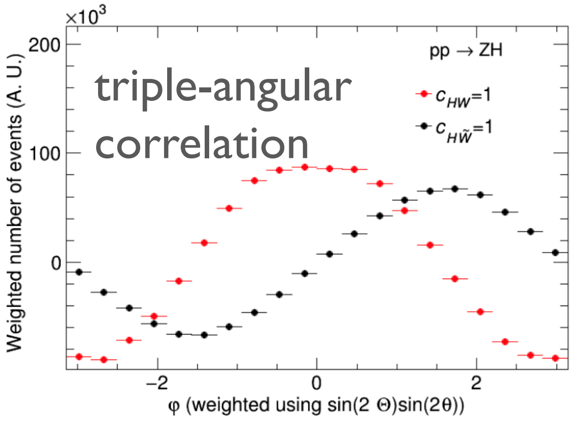
cHW/cHWtil

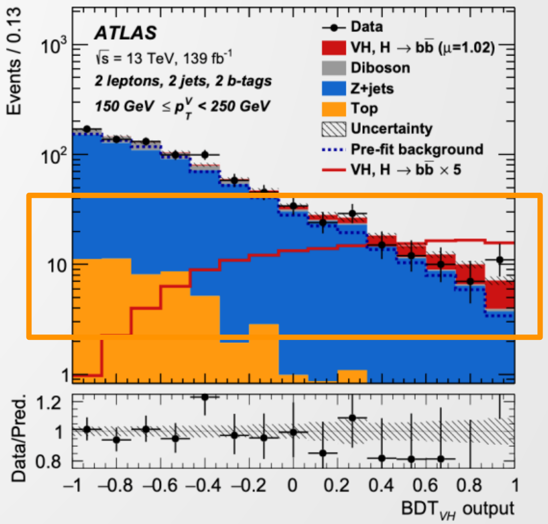


cHQ3 ↔ also in ttZ

Tree-level SMEFT amplitude of ZH (transverse polarisation):

$$\hat{\mathcal{M}}_\sigma^{\lambda=\pm} = g_Z m_Z \sqrt{\hat{s}}\left[\frac{g_{Z\sigma}}{\hat{s}-m_Z^2} + c_{\theta_W}\left(1+\frac{\hat{s}-m_h^2}{m_Z^2}\right)\left(\frac{g_{Z\sigma}c_{\theta_W}}{\hat{s}-m_Z^2}+\frac{Q_q e s_{\theta_W}}{\hat{s}}\right)\frac{v^2}{\Lambda^2}C_{HW} - \frac{2i\lambda k\sqrt{\hat{s}}}{m_Z^2}c_{\theta_W}\left(\frac{g_{Z\sigma}c_{\theta_W}}{\hat{s}-m_Z^2}+\frac{Q_q e s_{\theta_W}}{\hat{s}}\right)\frac{v^2}{\Lambda^2}C_{H\widetilde{W}}\right] + g_Z^2\frac{\sqrt{\hat{s}}}{m_Z}T_q^{(3)}\frac{v^2}{\Lambda^2}C_{HQ^{(3)}},$$





HOW TO PARAMETRIZE?

- Quantum field theory: Differential cross section predict polynomial SM-EFT dependence:

$$d\sigma(\boldsymbol{\theta}) \propto |\mathcal{M}_{\mathrm{SM}}(\boldsymbol{z}) + \boldsymbol{\theta}_a \mathcal{M}^a_{\mathrm{BSM}}(\boldsymbol{z})|^2 d\boldsymbol{z}$$

probability = wave function, squared

  - additivity of the matrix element → incur a simple (polynomial) dependence in $\boldsymbol{\theta}$ for fixed configuration z

$$\frac{d\sigma(\boldsymbol{x},\boldsymbol{\theta})}{d\boldsymbol{x}} = \frac{d\sigma_{\mathrm{SM}}(\boldsymbol{x})}{d\boldsymbol{x}} + \sum_a \theta_a \frac{d\sigma^a_{\mathrm{int.}}(\boldsymbol{x})}{d\boldsymbol{x}} + \frac{1}{2}\sum_{a,b}\theta_a\theta_b \frac{d\sigma^{ab}_{\mathrm{BSM}}(\boldsymbol{x})}{d\boldsymbol{x}}$$

- Neyman-Pearson: $q(\mathcal{D}) = \dfrac{L(\mathcal{D}|\boldsymbol{\theta}\;\;)}{L(\mathcal{D}|\mathrm{SM})}$ where $L(\mathcal{D}|\boldsymbol{\theta}) = \underbrace{\mathrm{P}_{\mathcal{L}\sigma(\boldsymbol{\theta})}(N)}_{\text{“normalization”}} \times \underbrace{\prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta})}_{\text{“shape”}}$

$$q_{\boldsymbol{\theta}}(\mathcal{D}) = \underbrace{\mathcal{L}(\sigma_{\boldsymbol{\theta}} - \sigma_{\mathrm{SM}})}_{\mathrm{const.}} - \sum_{\boldsymbol{x}_i\in\mathcal{D}} \log R(\boldsymbol{x}_i|\boldsymbol{\theta},\mathrm{SM})$$

Optimality can be achieved with cross-section ratio R or its **universal** coefficient functions R_a, R_{ab}

$$R(\boldsymbol{x}|\boldsymbol{\theta},\mathrm{SM}) = \frac{d\sigma(\boldsymbol{x},\boldsymbol{\theta})/d\boldsymbol{x}}{d\sigma(\boldsymbol{x},\mathrm{SM})/d\boldsymbol{x}} = 1 + \sum_a \theta_a R_a(\boldsymbol{x}) + \frac{1}{2}\sum_{a,b}\theta_a\theta_b R_{ab}(\boldsymbol{x})$$

$$\hat{=} \left(1 + \sum_a \theta_a \hat{n}_a(\boldsymbol{x})\right)^2 + \sum_a \left(\sum_{b\geq a} \theta_b \hat{n}_{ab}(\boldsymbol{x})\right)^2$$

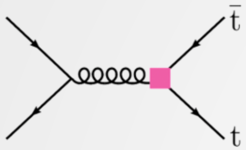
NB #1 Curse of dimensionality is lifted!!
15 operators → 136 coefficients

NB #2: R is positive: Fit universal dependence using the most general quadratic polynomial

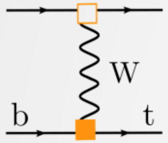
TOP QUARK PAIR + Z BOSON

JHEP 12 (2021) 083

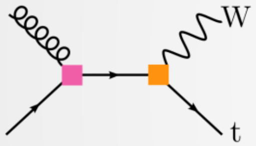
tt
811 pb



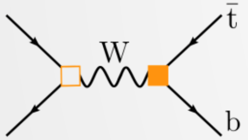
t (t-channel)
217 pb



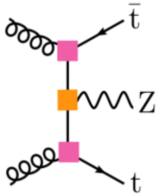
tW
72 pb



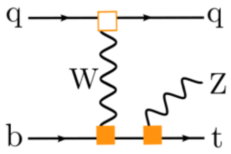
t (s-channel)
10 pb



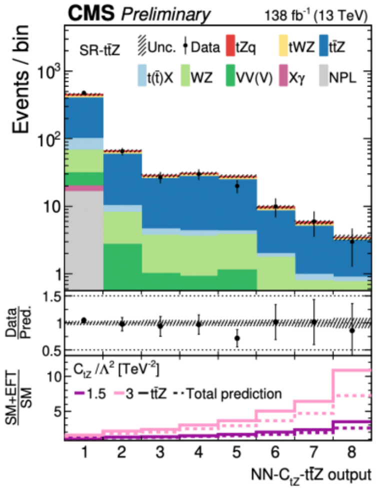
ttZ
1 pb

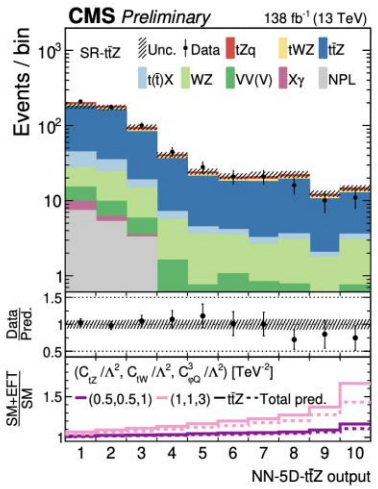


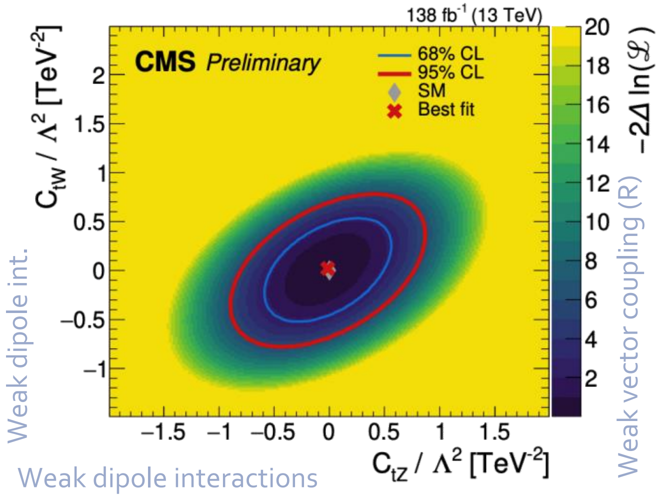
tZq
0.088 pb

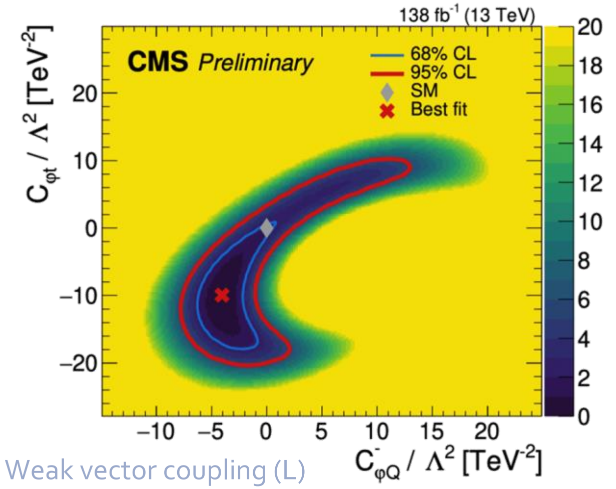


- Measure the top quark – Z boson coupling
- Train separate "SM vs. EFT" classifiers
  - Single operator O_{tZ}, O_{tW}, $O^3_{\phi Q}$
  - different trainings for different limits (!)
  - "likelihood trick" for SMEFT effects
- signal extraction with 1D, 2D, and 5D LL fit
  - Sampling of parameter space in the training
  - Targeted signals differ kinematically, but no parametrized training is used
  - Signal mix
  - no large linear terms → OK
- Best current limits





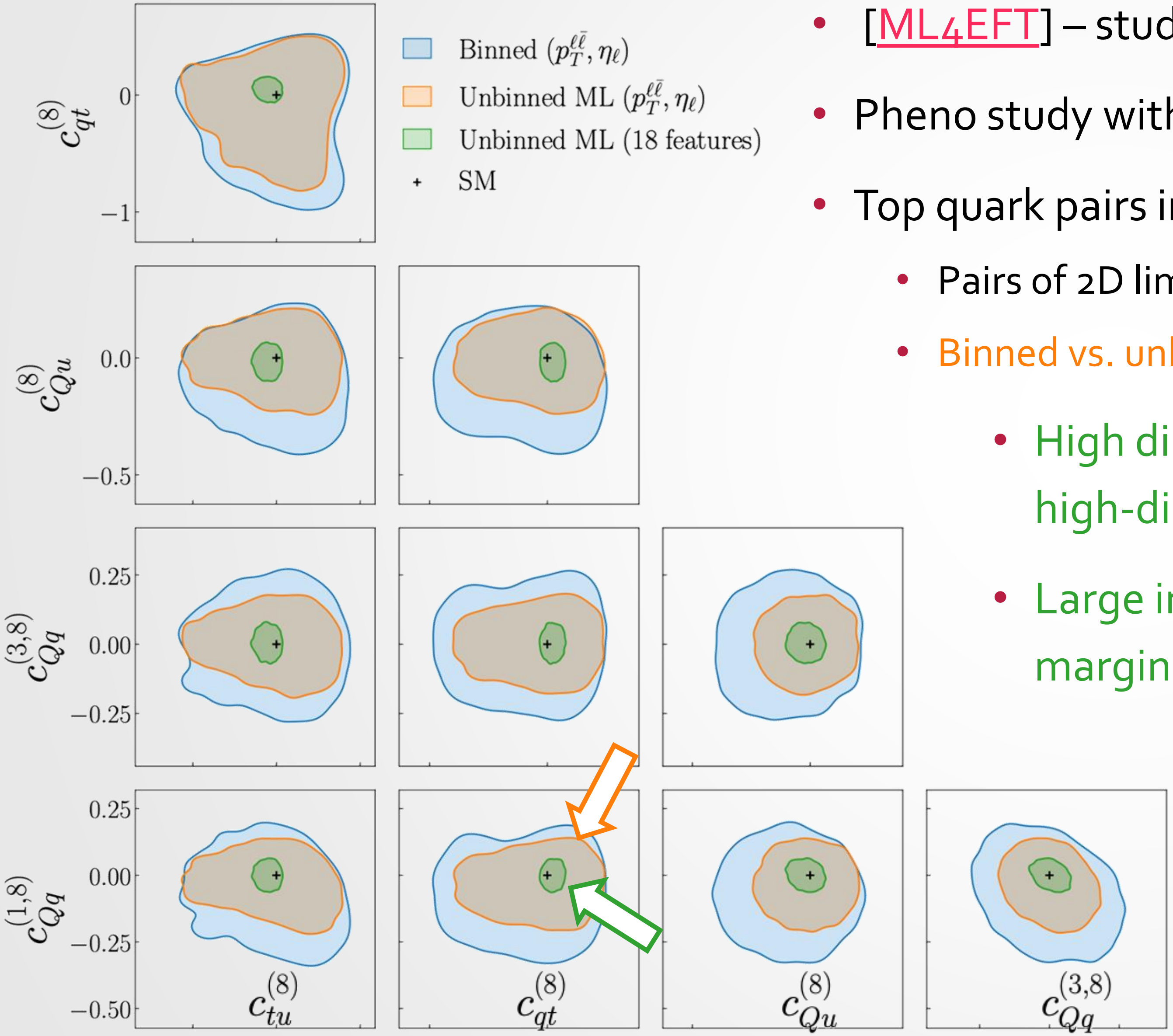




ML4EFT R. Ambrosio, J. Hoeve, M. Madigan, J. Rojo, V. Sanz [2211.02058]

[CMS-TOP-PAS-20-006]

IMPROVING HIGH DIMENSIONAL LIMITS



- [ML4EFT] – study ZH and top quark pairs
- Pheno study with parametrized NN classifiers
- Top quark pairs in low (N_f=2) and high feature dimension N_f=18
  - Pairs of 2D limits with 6 more ops marginalized
  - Binned vs. unbinned: Some gain w/ unbinned when using 2 features
    - High dimensional observation (N_f=18) constraining a high-dimensional (N_{coef}=8) model using an SM candle
    - Large improvement for N_f=18– mostly in the marginalized limits
- Take seriously constraining power from SM candle
- Whether the sensitivity gain survives systematics in an unbinned detector-level analysis is an open question

TOWARDS UNBINNED ANALYSIS

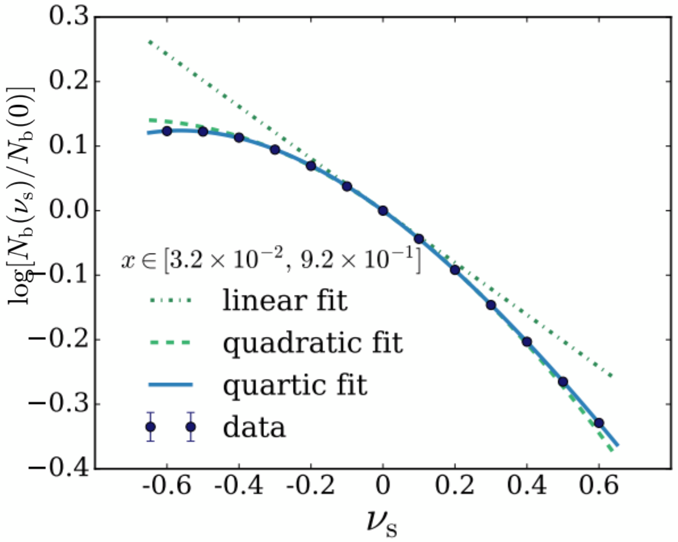
- Binned parametrized classifiers are impractical for high SMEFT parameter dimension
- What's missing to go all-in? Systematics.
  -
  - ²

$L = \sum$
$\theta,$

ematic
ations
cheap!

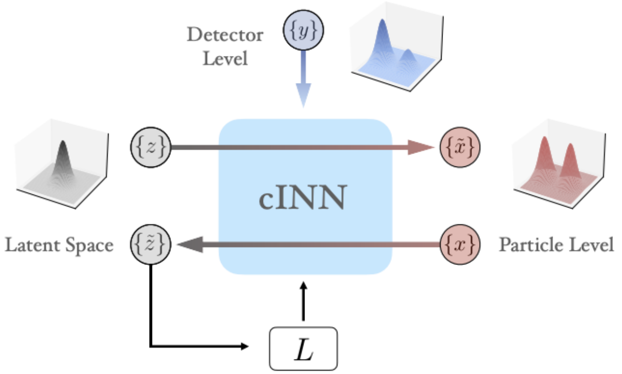
M,ν=0

[Pierini et.al 2111.13633]



- A ch h event counts in the profiling
- Divide & conquer #1: Experiments begun machine-learning certain nuisances: h_{damp}, b-fragmentation

- Divide & conquer #2: Unbinned unfolding for high dimensions
- Consider on the conditional pdf $p(x_{det}|z_{ptl})$ which can be evaluated in the forward mode
- Unfolding algorithms use Bayes' theorem
$$p(x_{det}|z_{ptl})p(z_{ptl}) = p(z_{ptl}|x_{det})p(x_{det})$$
to learn $p(z_{ptl}|x_{det})$; GAN & other generative versions
  - Mostly iterative, to remove simulated prior



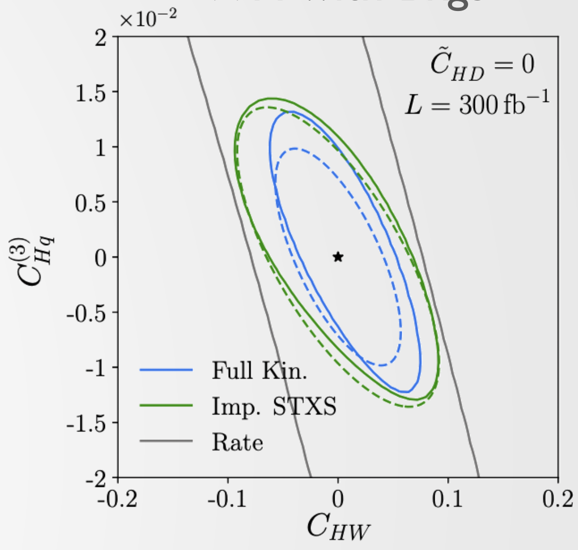
[community paper]
e.g. [OmniFold]
[cINN], [all]

- Report unbinned unfolded data; then SMEFT analysis

REFERENCES

- Madminer: Neural networks based likelihood-free inference & related techniques
    - K. Cranmer , J. Pavez , and G. Louppe [1506.02169]
      J. Brehmer, K. Cranmer, G. Louppe, J. Pavez [1805.00013] [1805.00020] [1805.12244]
      J. Brehmer, F. Kling, I. Espejo, K. Cranmer [1907.10621]
    - J. Brehmer, S. Dawson, S. Homiller, F. Kling, T. Plehn [1908.06980]
    - A. Butter, T. Plehn, N. Soybelman, J. Brehmer [2109.10414]
    - established many of the *main ideas* & *statistical interpretation* in various *NN applications*
- Weight derivative regression (A.Valassi) [2003.12853]
- Parametrized classifiers for SM-EFT: NN with quadratic structure
- S. Chen, A. Glioti, G. Panico, A. Wulzer [*JHEP* 05 (2021) 247] [arXiv:2308.05704]
- Boosted Information Trees: Tree algorithms & boosting
    - S. Chatterjee, S. Rohshap, N. Frohner, R.S., D. Schwarz [2107.10859], [2205.12976]
- ML4EFT R. Ambrosio, J. Hoeve, M. Madigan, J. Rojo, V. Sanz [2211.02058]
- All approaches are "SMEFT-specific ML" with differences mostly on the practical side

WH with Bkgs



my practical experience

→ talk later today