Machine Learning for Top Quarks

Matthias Komm

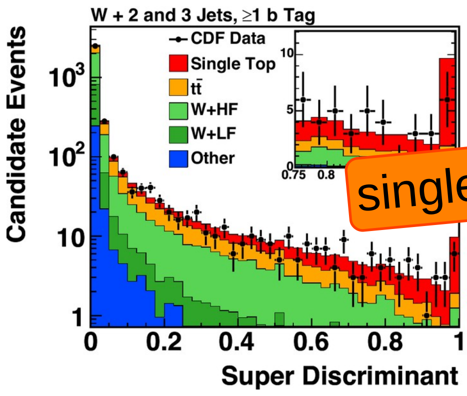
for the ATLAS & CMS collaborations

Introduction

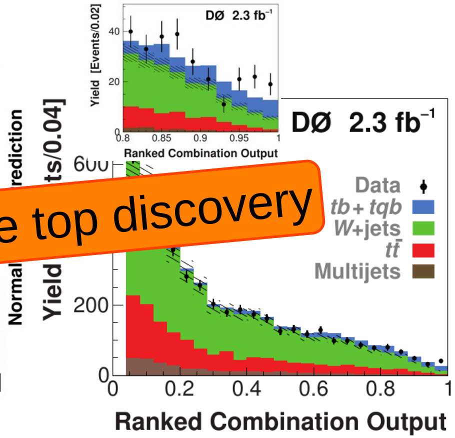
- Top quark research; driven by ML!
  - best customers of b tagging since $\mathcal{B}(\mathrm{t} \to \mathrm{bW}) \approx 1$
  - top quark measurements used NNs & BDTs (+MEM) already at the Tevatron

23 years ago

PRD 82 (2010) 112005



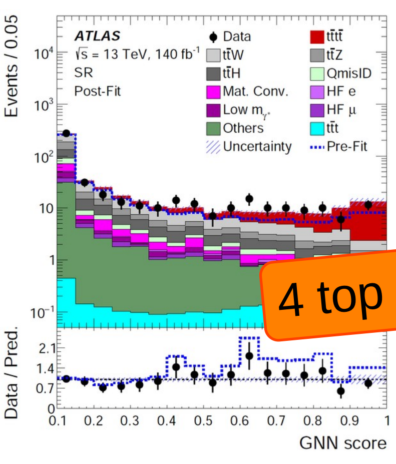
PRL 103 (2009) 092001



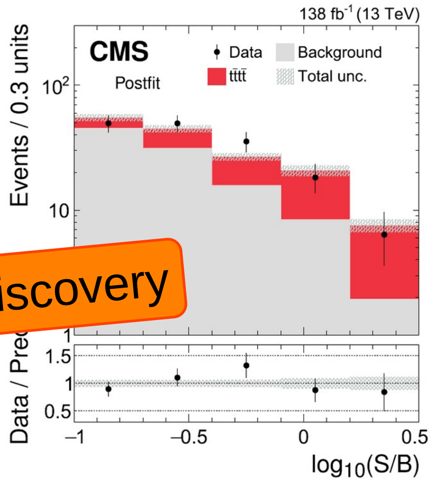
single top discovery

- ML today
  - hunting for very rare processes t(t) + vector bosons; 4 top quarks; etc.
  - rapidly evolving field

EPJC 83 (2023) 496



PLB 847 (2023) 138290

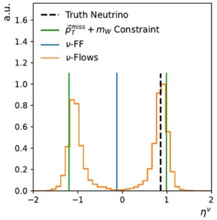


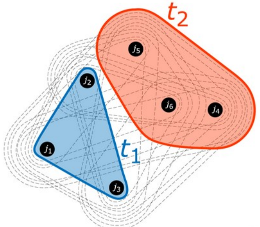
4 top discovery

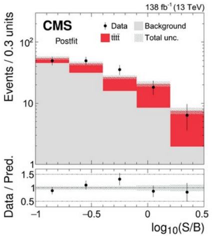
Outline

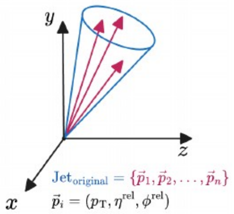
Disclaimer: will focus on (some) techniques & ideas rather than analyses → big field!

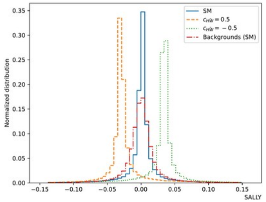
- ML for analyses
  - event reconstruction
  - background estimation
  - signal vs. background classification
- ML for interpretation
  - likelihood-free inference
  - reweighting
  - unfolding
- ML for HL LHC

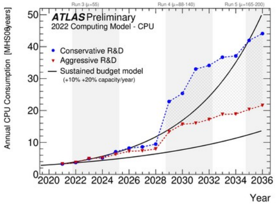


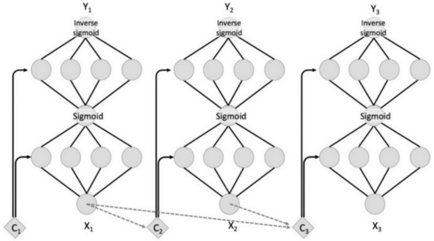


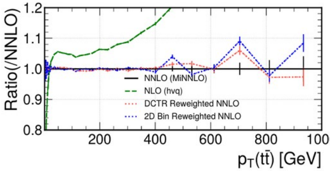


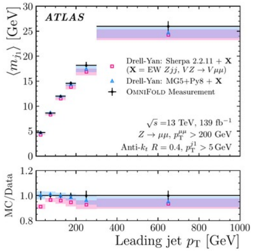






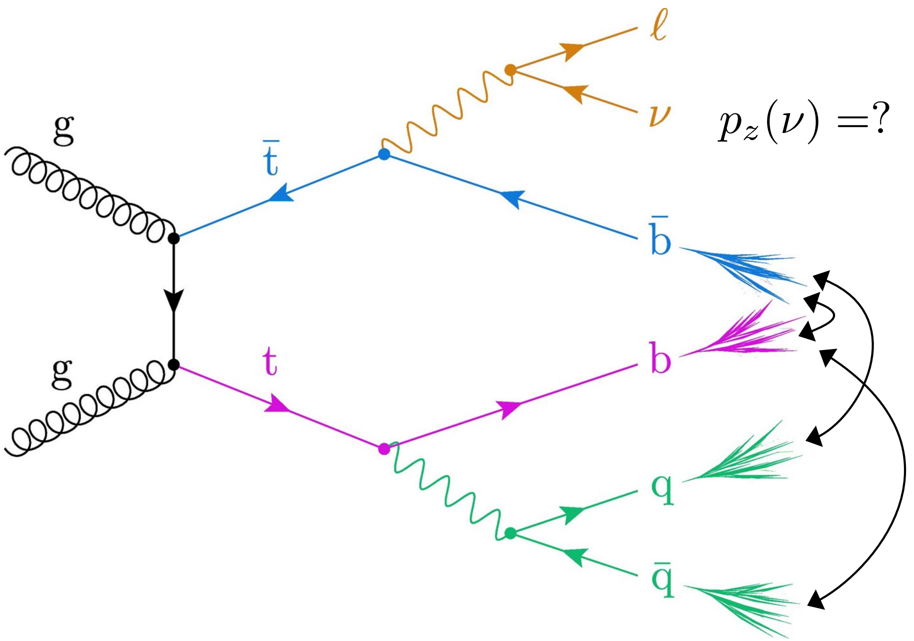






ML for top quark **analysis**

Reconstructing top quarks *traditionally*



- find unknown momentum of neutrino(s)
  - W boson mass constraint still leaves ambiguities
    - 2 real solutions
    - complex solutions
  - even more complicated for dilepton $t\bar{t}$

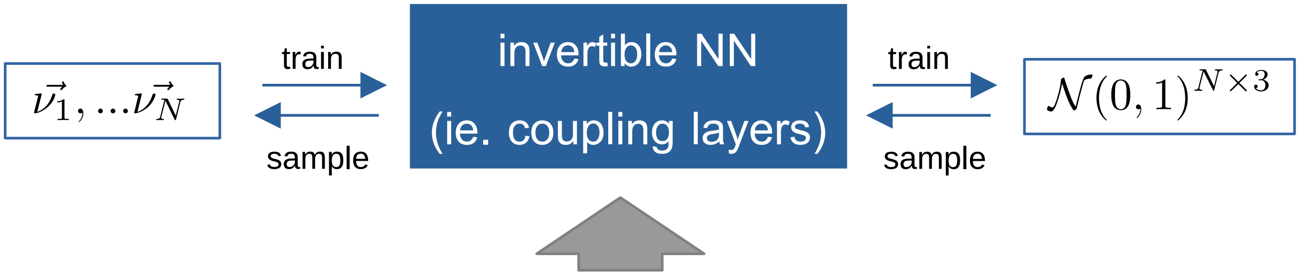
- jet-parton assignments
  - large combinatorial problem; eg. 2520 for $t\bar{t}{+}2j$
  - brute force approaches: χ^2 & kinematic fitting
    → need to iterate through all combinations

- ML can improve both tasks!

Normalizing flows for ν solution

SciPost Phys. 14 (2023) 159; PRD 109 (2024) 1, 012005

- idea
  - learn full likelihood of solutions instead of single point



neutrino momenta
from generator truth

Nx3 dim. normal
distribution

$\vec{\nu}_1, ...\vec{\nu}_N$

train

sample

invertible NN

(ie. coupling layers)

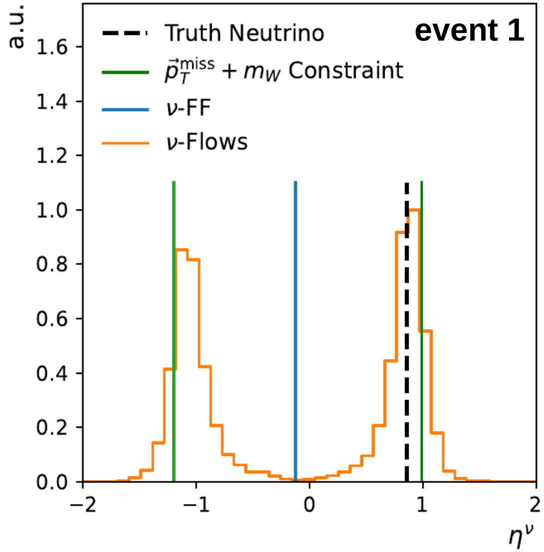
train

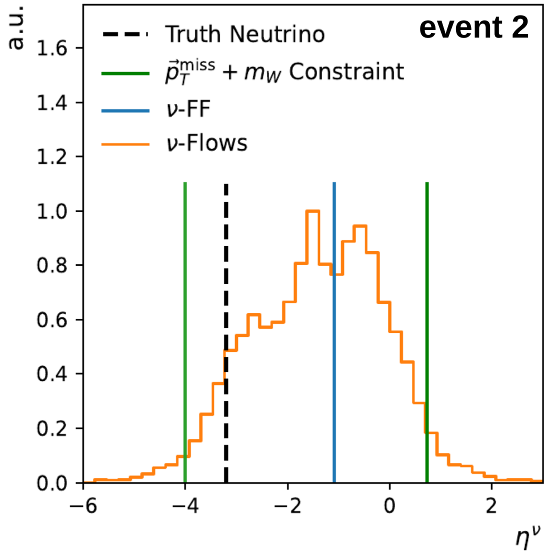
sample

$\mathcal{N}(0,1)^{N\times 3}$

conditioned on reconstructed observables:

$\vec{p}_T^{\,\mathrm{miss}}, \vec{p}(\ell), \vec{p}(j), N_{\mathrm{jets}}, N_{\mathrm{b\text{-}tags}},$ lepton/jet properties





Attention for jet assignment

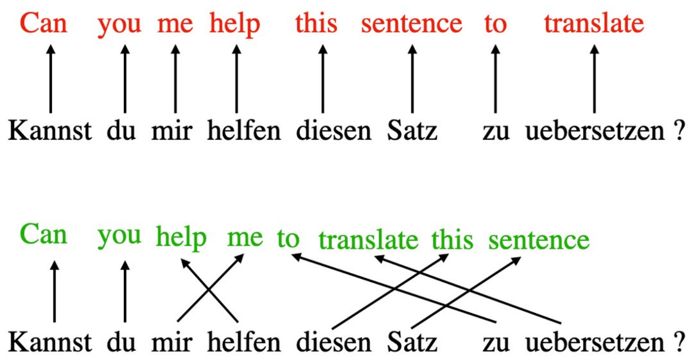
- sequence mapping with transformer
  - learn **Q**uery, **K**ey, **V**alue

$$\vec{y}_i = \underbrace{\text{softmax}\left(\vec{Q}\cdot\vec{K}^T\right)}_{\text{attention}} \cdot\vec{V}$$

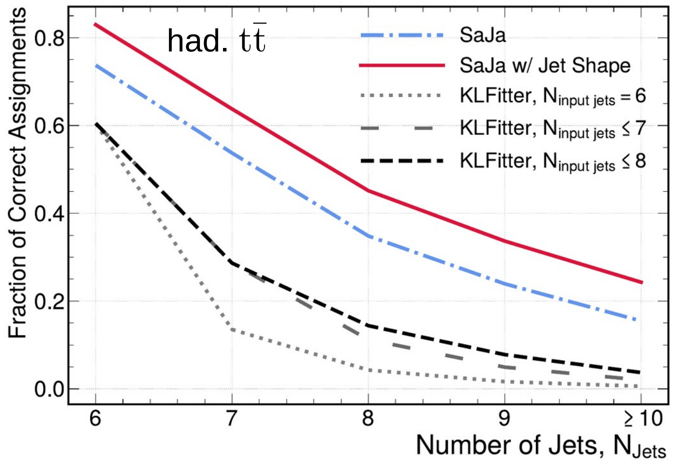
  → outputs value if query matches key
  - more powerful than LSTM

- Self-Attention for Jet Assignment (SAJA)
  J. Korean Phys. Soc. 84 (2024) 427
  - Q,K,V are learned from each element → self-attention
  - assignment uncertainty estimated using Shannon entropy $\mathbb{H}[\hat{Y}] = \frac{1}{N}\sum_{j=1}^{N}\left(-\sum_{c\in\text{classes}} \hat{y}_c^{(j)}\log\hat{y}_c^{(j)}\right)$

translation with attention



SAJA

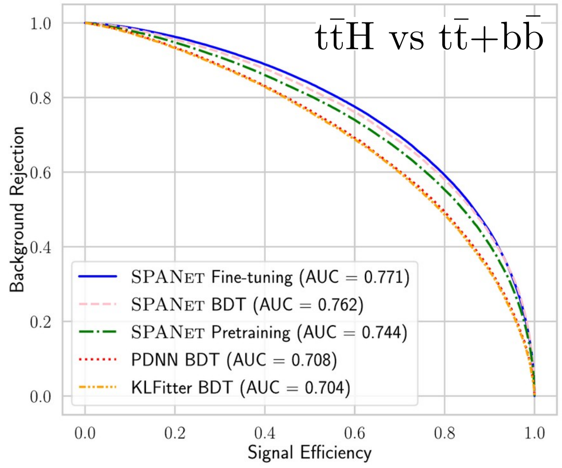


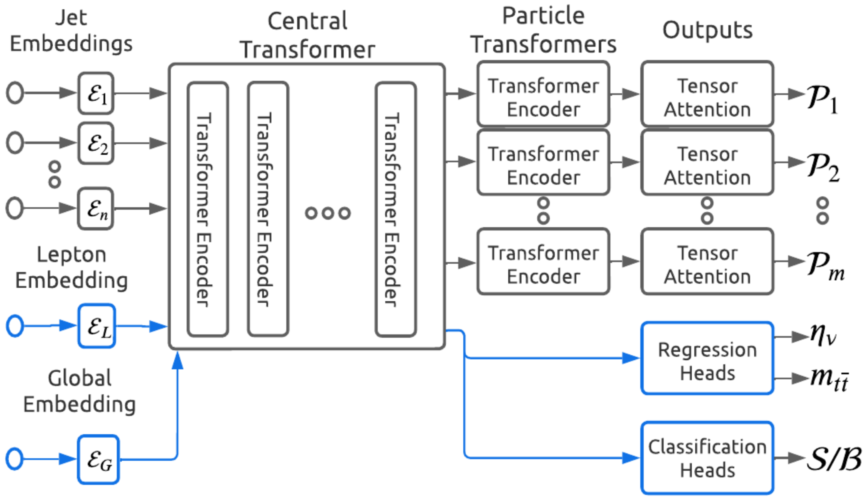
Attention for jet assignment (2)

- SPA NET

  SciPost Phys. 12, 178 (2022), Commun Phys 7, 139 (2024)

  - transformers + symmetry-aware attention
  - complexity $\mathcal{O}(N_{jets}^{\#\text{daughters}})$ instead of $\mathcal{O}(N!)$
  - **76%** of semileptonic $t\bar{t}$ events correctly reco'ed! (only 42**%** using KLFitter)
  - regression of auxiliary targets possible, eg. $\eta(\nu)$





Jet Embeddings

Central Transformer

Particle Transformers

Outputs

$\mathcal{E}_1$

$\mathcal{E}_2$

$\mathcal{E}_n$

Lepton Embedding

$\mathcal{E}_L$

Global Embedding

$\mathcal{E}_G$

Transformer Encoder

Transformer Encoder

Transformer Encoder

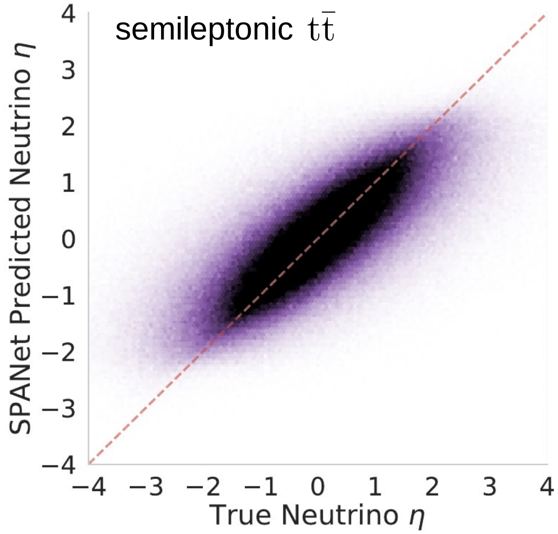
Transformer Encoder → Tensor Attention → $\mathcal{P}_1$

Transformer Encoder → Tensor Attention → $\mathcal{P}_2$

Transformer Encoder → Tensor Attention → $\mathcal{P}_m$

Regression Heads → η_ν, $m_{t\bar{t}}$

Classification Heads → $\mathcal{S}/\mathcal{B}$

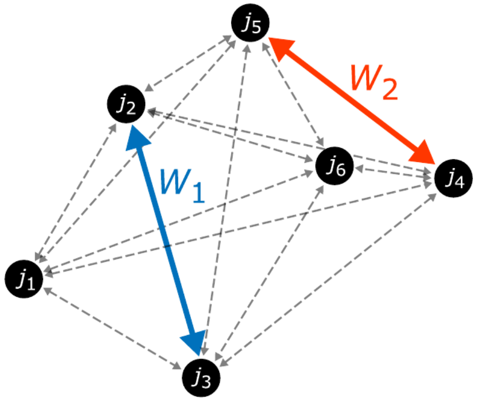


Hypergraphs

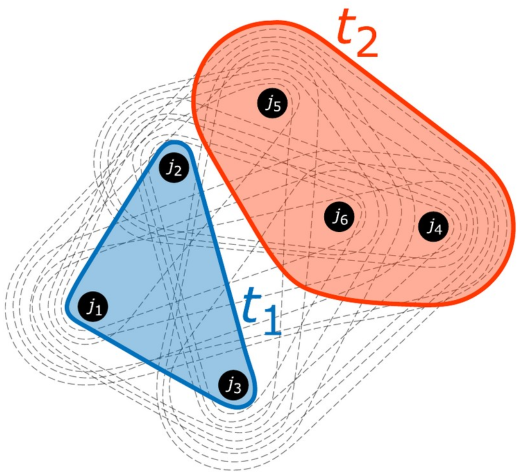
arXiv:2402.10149

- idea
  - message passing in normal GNNs only between pairs of nodes
    → ok for finding $\mathrm{W} \to \mathrm{q\bar{q}}$
  - hypergraphs defines multiple nodes per edge; can represent $\mathrm{t} \to \mathrm{bq\bar{q}}$

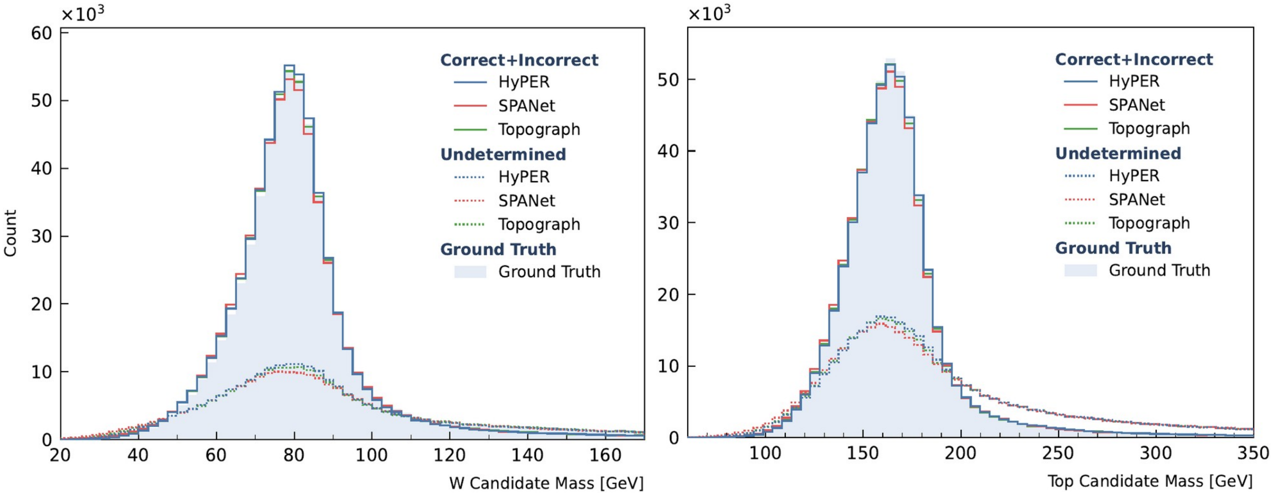
classical graph



hypergraph



- results
  - able to assign 67% of had. $\mathrm{t\bar{t}}$ (SPA NET: 65%)
  - HyPER: only 345k parameters! (SPA NET: 10.7M)

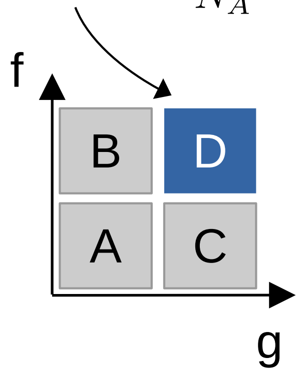


Background estimations

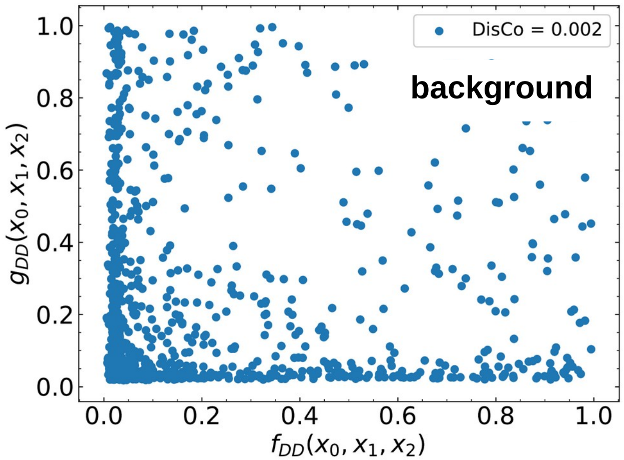
- ABCDisCo Phys. Rev. D 103, 035021 (2021)
  - idea: NN outputs 2 independent variables to discriminate against signal
  - loss: $\mathcal{L}[f,g] = \mathcal{L}_{\text{classifier}}[f(X),y] + \mathcal{L}_{\text{classifier}}[g(X),y] + \lambda \boxed{\text{dCorr}^2_{y=0}[f(X),g(X)]}$

    → use f & g to construct ABCD estimate of bkg. **distance correlation**
  - NB: also useful to decorrelate against auxiliary observable; eg. unfolding observable
  - “easier” to train than adversarial approaches; comparable performance

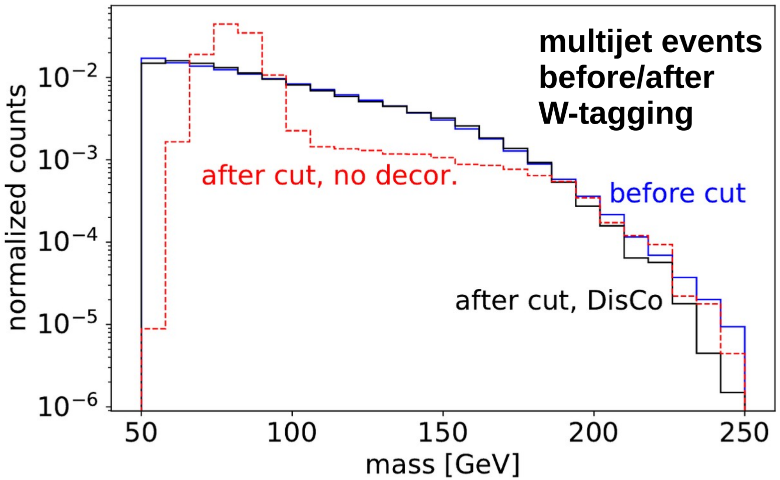
$$N_D = \frac{N_B N_C}{N_A}$$



toy model



decorrelation example



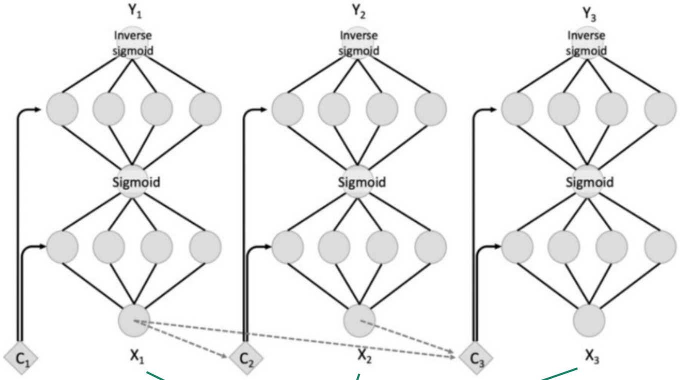
Background estimations (2)

- Normalizing flow

  Phys. Lett. B 844 (2023) 138076

  - idea: use autoregressive normalizing flow to learn a mapping of background distribution CR → SR
  - applied in full hadronic 4t analyses by CMS
  - trained on $t\bar{t}$ MC
  - 2D transform of $(H_\mathrm{T}, \mathrm{BDT\ score})$

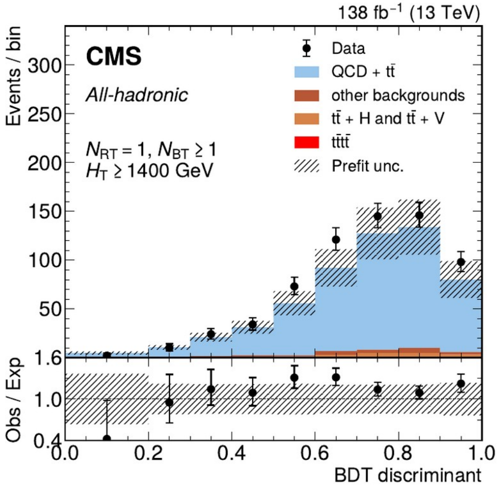
target



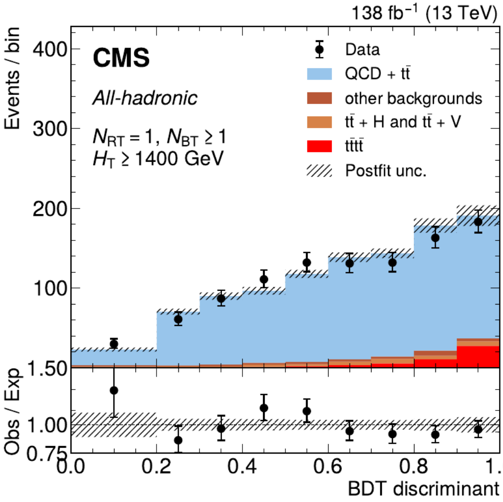
conditions

source

validation



estimated bkg. in SR



Signal vs bkg. **classification**

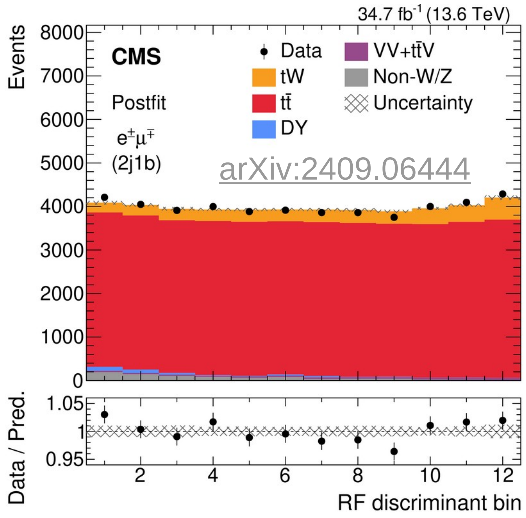
(some thoughts on something supposedly simple)

- optimal binning scheme of classifier output score?
  → might leave performance on the table

- classification = not the best training target?
  - optimize for discovery significance
  - systematic (& profiling)-aware training
    → CMS-PAS-MLG-23-005, INFERNO

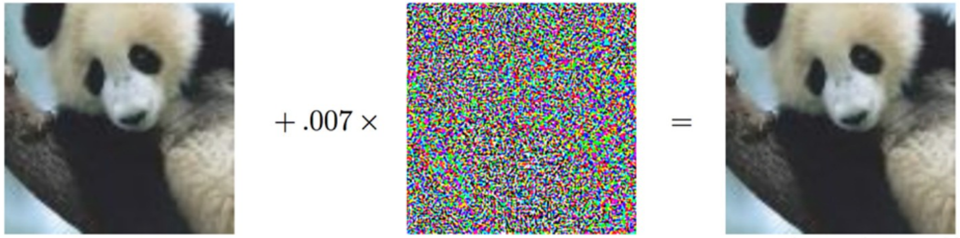
- provocative: are we reinforcing the SM by training only on SM?
  - augmenting signal (& bkg) training samples
  - explainable & robust ML
  - train on data? un-/semi-supervised

flat bkg. binning scheme



arXiv:2409.06444

adversarial attack



"panda"
57.7% confidence

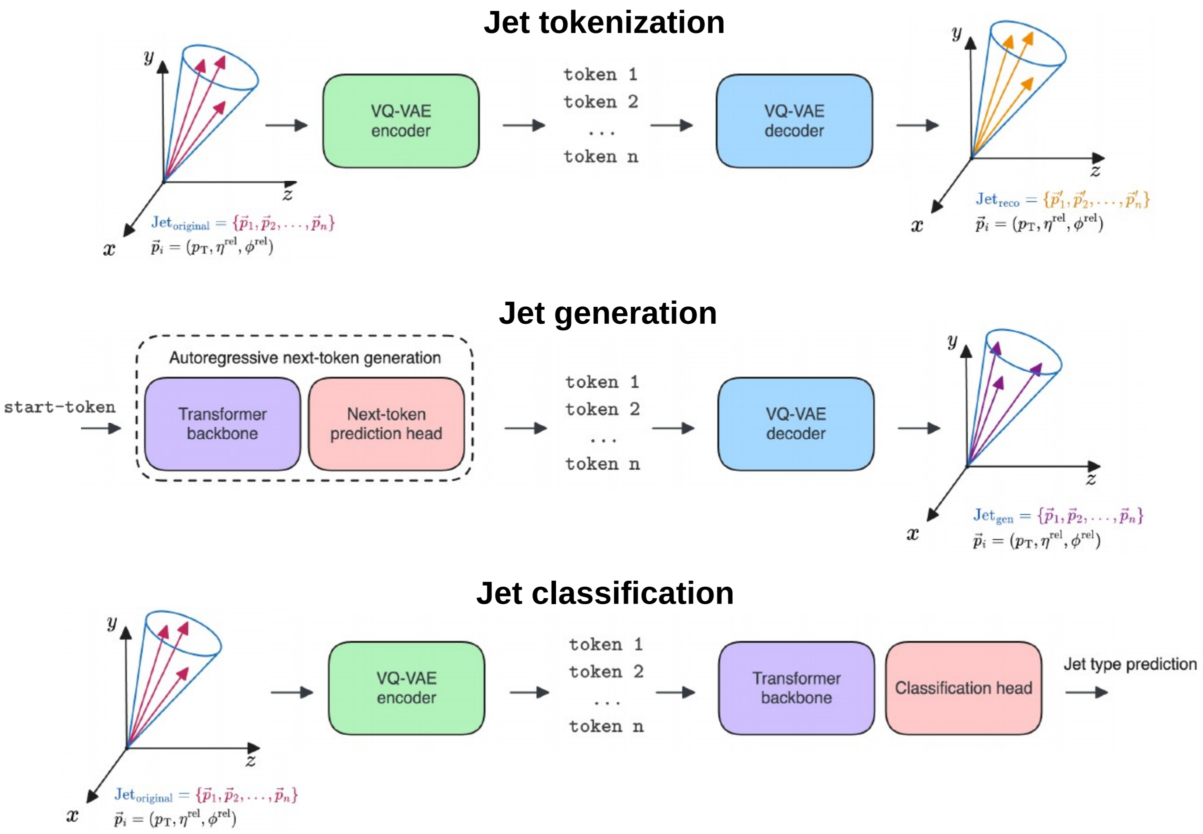
"nematode"
8.2% confidence

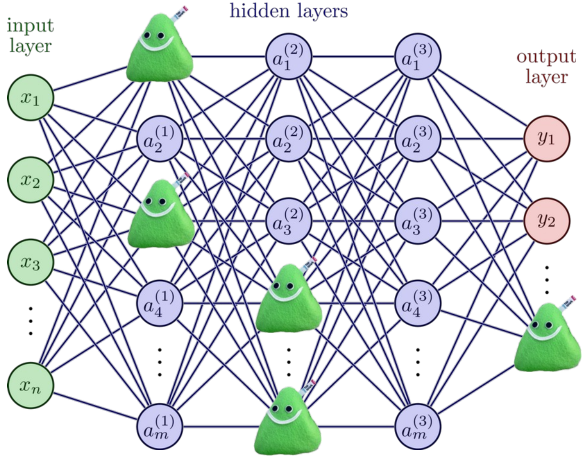
"gibbon"
99.3 % confidence

A **foundation model** for top quarks?

- foundation models (eg. OpenAI's GPT)
  - pretrained on big datasets; basis for many applications
  - reduces computational costs & sharing of common tasks
  - in HEP: first attempt with jets
- ideas for top quarks
  - can harmonize/share: reconstruction techniques, selection & classification, unfolding
  - cross experiment/theory: common tokens from custom tokenization step

OmniJet-alpha MLST 5 (2024) 3, 035031



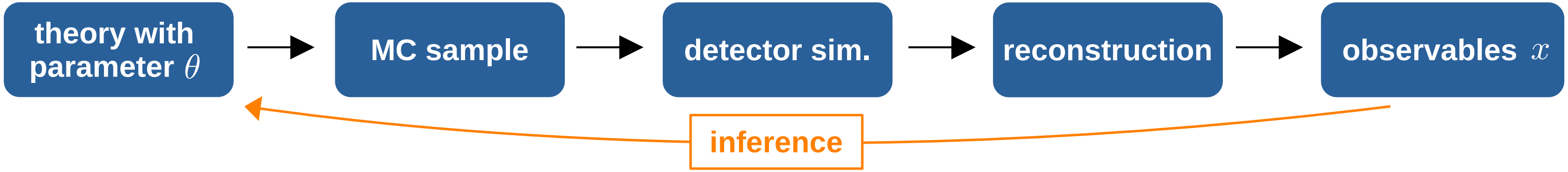


ML for top quark **interpretation**

Likelihood-free **inference**

arXiv:2010.06439

also called *simulation-based inference*



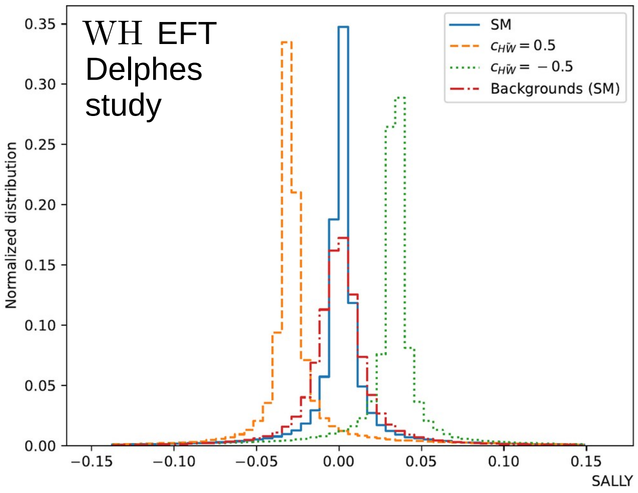
- “easy” to run simulator & generate samples $x \sim p(x|\theta)$
- likelihood $p(x|\theta)$ intractable; typical solution: use a test statistic (ie. histograms) $x' \Rightarrow p(x'|\theta)$

➢ idea: approximate likelihood ratio with ML classifier score

$$s(x) = \frac{S}{S+B} \Rightarrow \frac{H_1}{H_0} = \frac{s(x)}{1-s(x)}$$

- only few ideas/applications in HEP so far:
  SALLY, INFERNO, ... but many more in other fields

JHEP 04 (2024) 014



> “Simulation-based inference is the next step in the methodological evolution of statistical practice in the sciences”
>
> https://simulation-based-inference.org/

Learn to **reweight**

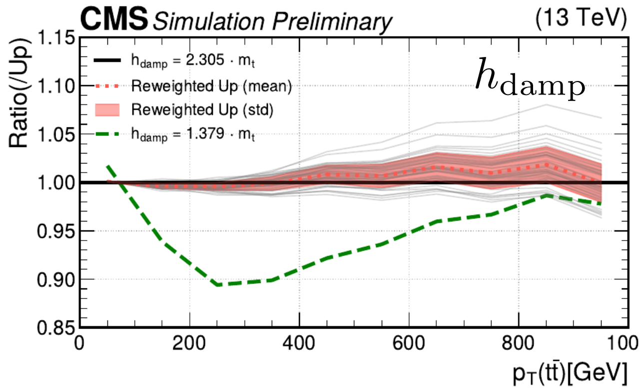
CMS-PAS-MLG-24-001, PRD 101, 091901 (2020)

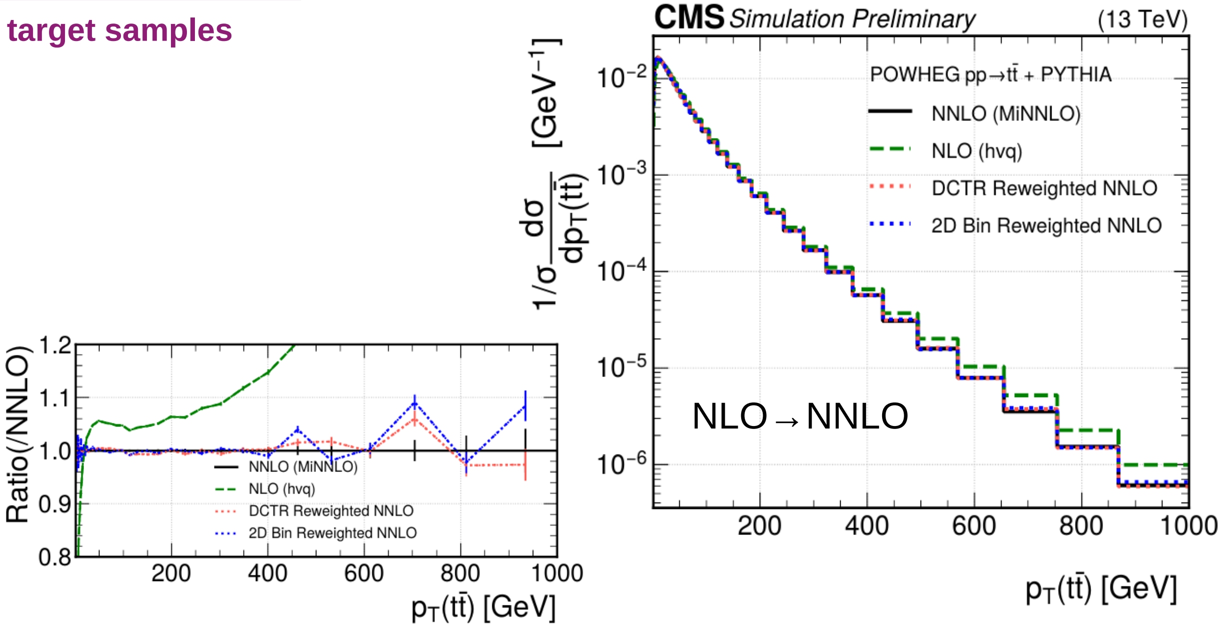
- typically need dedicated samples to assess MC modeling uncertainties (eg. hdamp, b fragmentation, etc.)
- DTCR method: use NN classifier to derive event weights

$$\text{loss}(f(x,\underbrace{\boxed{\theta}}_{\text{parameter}})) = -\underbrace{\boxed{\sum_{i\in\theta_0}\log f(x_i,\theta)}}_{\text{source sample}} - \underbrace{\boxed{\sum_{i\in\boldsymbol{\theta}}\log(1-f(x_i,\theta))}}_{\text{target samples}}$$

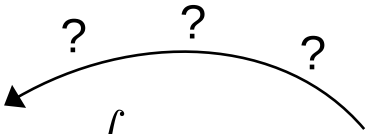
→ enables continuous reweighting

- particle flow network (deep-set like)
- reweighting to incorporate higher order correction also possible!
- integrated into central CMS software using ONNX

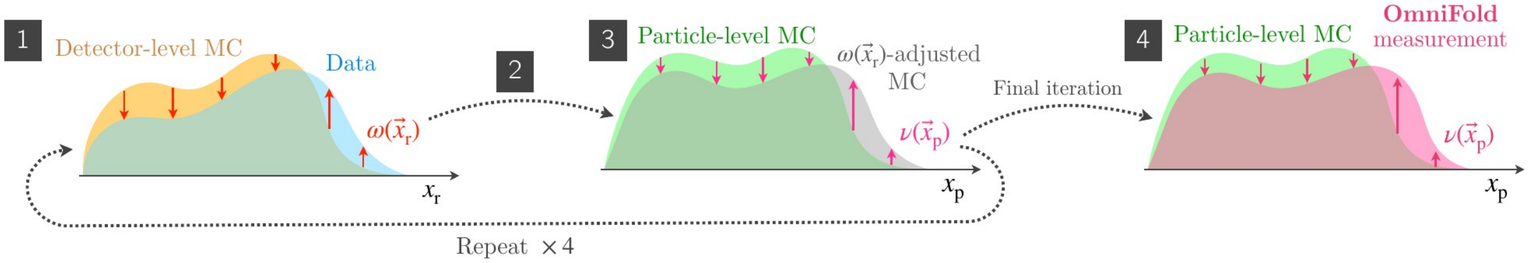




Unfolding

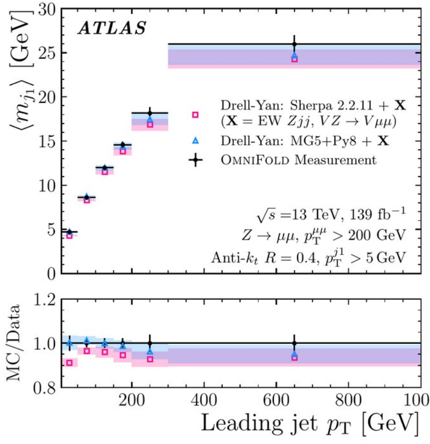


- an ill-posed inverse problem: $p_{\text{gen}}(y) = \int \underset{\text{efficiency}}{\epsilon(y)}\, \underset{\text{response}}{R(y,x)}\, p_{\text{reco}}(x)\,\mathrm{d}x$
  - ML cannot get around this problem; but it can …
  - improve reconstruction → less migrations
  - estimate n-dim. (unbinned) response matrix
- Omnifold: NN-based multidim. unbinned unfolding
  - inspired by d'Agostini iterative unfolding
  - NN trained to reweight simulation to data
  - iteratively propagate the learned weights to particle/parton level
  - can unfold quantities beyond cross sections, eg. $\langle m_{j1} \rangle$ vs p_T



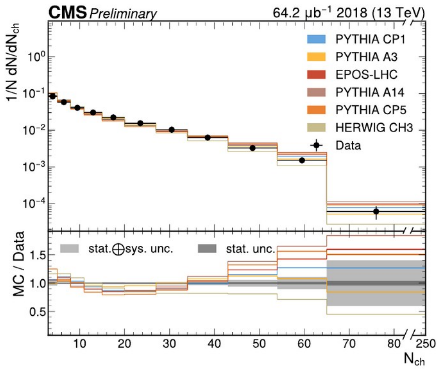
Ominfold applied to Z+jets

arXiv:2405.20041



Ominfold applied to min. bias

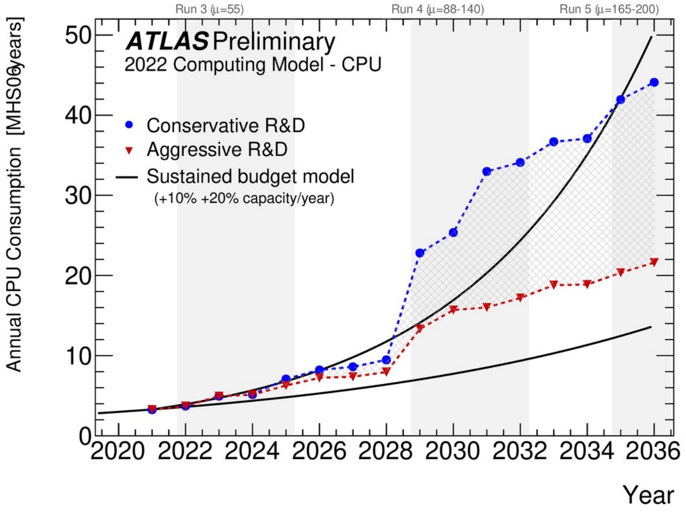
CMS-PAS-SMP-23-008



HL LHC **challenges**

- data sets will increase significantly; do we need it all?
  → ML can identify clean events with low uncertainties
- analyses will require lots of MC samples
  → speed up simulation/systematic evaluation
  Comput Softw Big Sci 5, 15 (2021), SciPost Phys. 14, 079 (2023)
- reusability and reinterpretation → LHC legacy
  - Les Houches guide to reusable ML models arXiv:2312.14575
  - classifier surrogates for analysis reinterpretations arXiv:2402.15558
- how to evaluate ML models fast?
  - direct, ie. within the process; restricted often to CPU only
  - indirect, deferred to dedicated machine/REST API endpoint
    → NVIDIA Triton, SONIC by CMS Comput. Softw. Big Sci. 8 (2024) 17
  - edge, on special hardware (eg. FPGA)
    → hls4ml JINST 13 P07027 (2018), arXiv:2308.05170

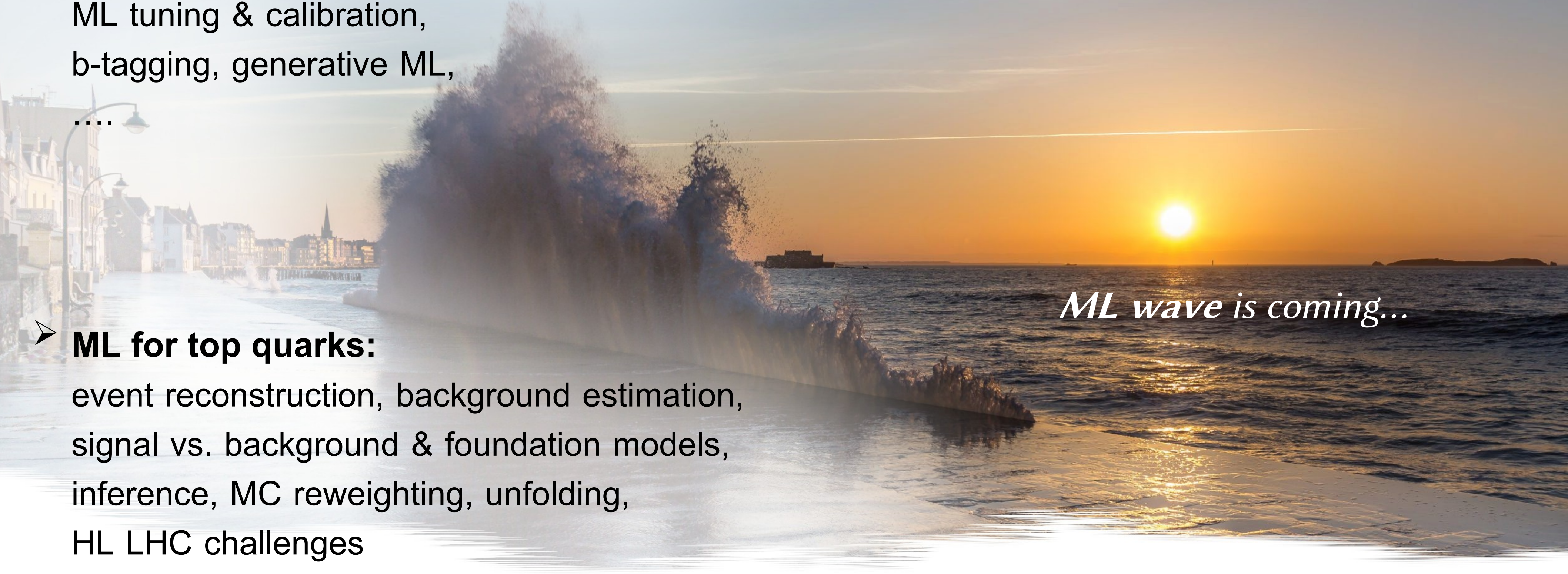
CERN-LHCC-2022-005



Summary

- ML is a big field! I’ve not talked about …
  unsupervised learning/anomaly detection,
  ML tuning & calibration,
  b-tagging, generative ML,
  ….

- **ML for top quarks:**
  event reconstruction, background estimation,
  signal vs. background & foundation models,
  inference, MC reweighting, unfolding,
  HL LHC challenges



ML wave *is coming…*