Interpretable graphical models for collider studies

Machine Learning at GGI

Manuel Szewc, Jozef Stefan Institute

In this talk

I'll introduce a little bit of collider physics and motivate the need for better modelling in a few areas

Detail **one** possible avenue: probabilistic graphical models or Bayesian networks

I'll show one very detailed example detailed on arxiv:2107.00668, you can ask me also about arxiv:2112.11352 where we apply graphical models to quark/gluon jets to get a truly unsupervised tagger

Motivation

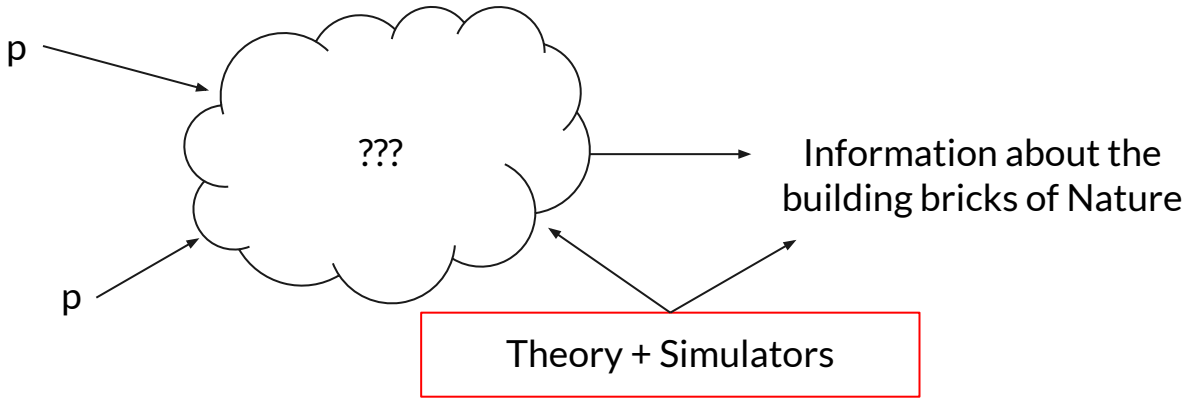
The Standard Model of particle physics represents our understanding of (some of) the most fundamental aspects of Nature.

However, experimental observations (dark matter, neutrino masses and gravity,...) and theoretical issues (θ_{QCD}problem, the Higgs hierarchy problem,...) make clear the limitations of the SM.

How do we explore the SM and beyond?

Motivation

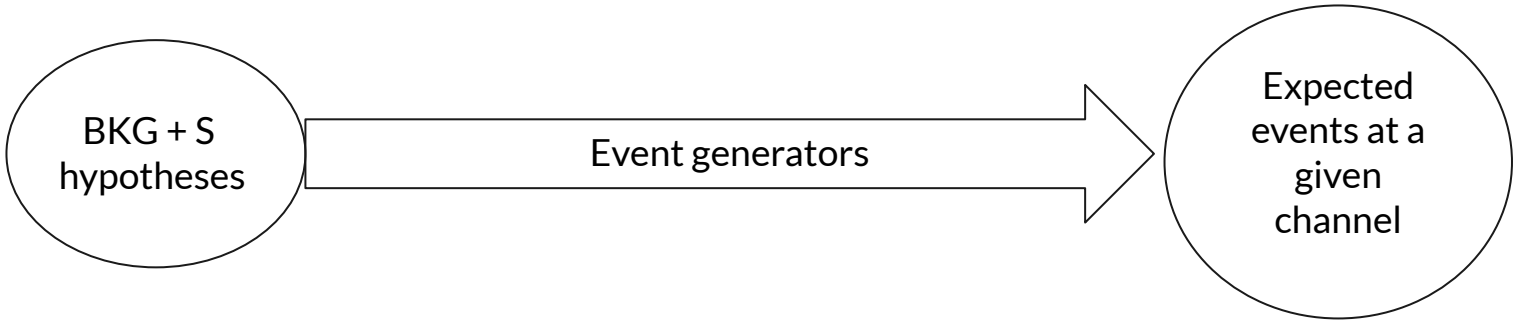
Colliders are among the most powerful tools we have to test and expand our knowledge



Monte Carlo based strategies

We perform dedicated analyses with specific topologies in mind.

Dedicated searches rely on good modelling: the simulation pipeline is fundamental.



Monte Carlo based strategies

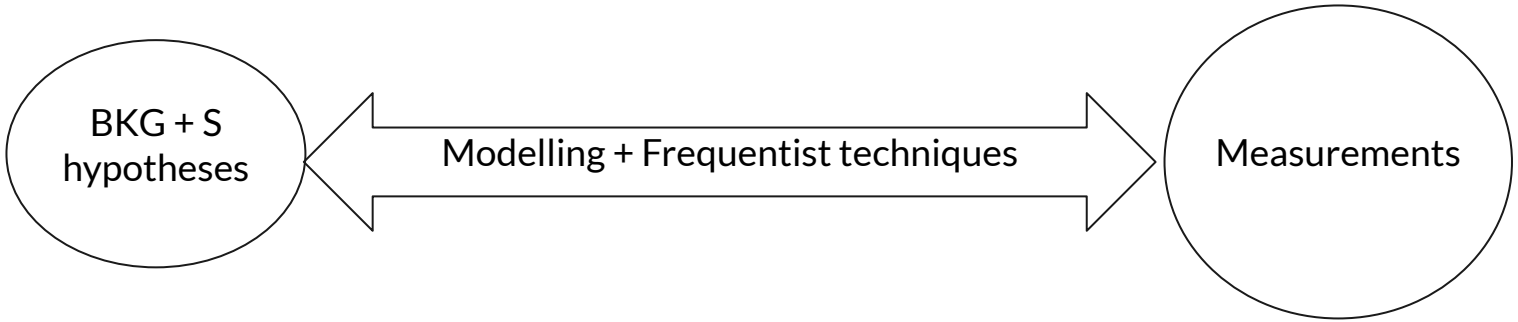
With the expected events, we can turn from estimation to inference. In collider studies, there is a primacy of frequentist methods to estimate or set limits to the parameters of interest or even for model selection.

MC based strategies for detecting a given signal

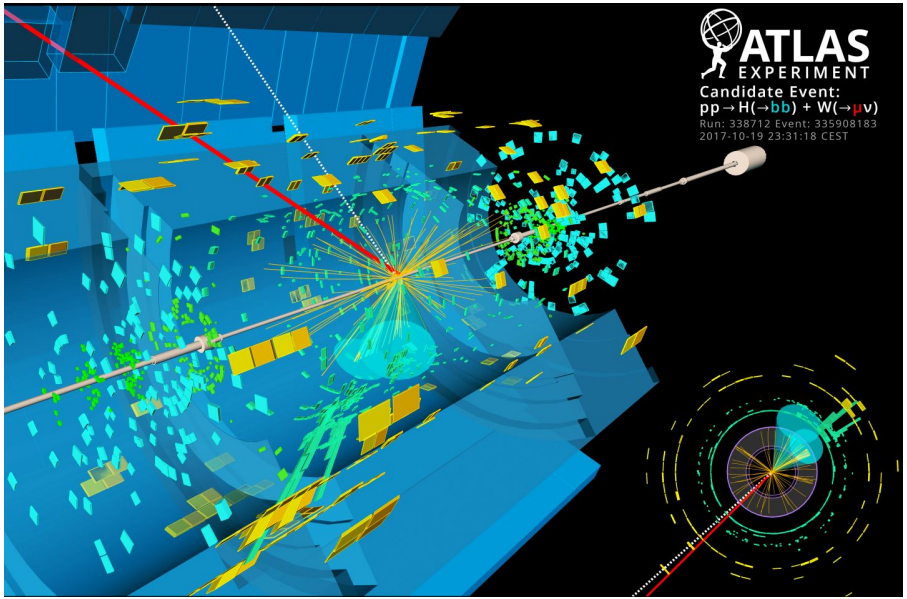
BKG-only hypothesis discarded → Discovery (similar to Higgs discovery)

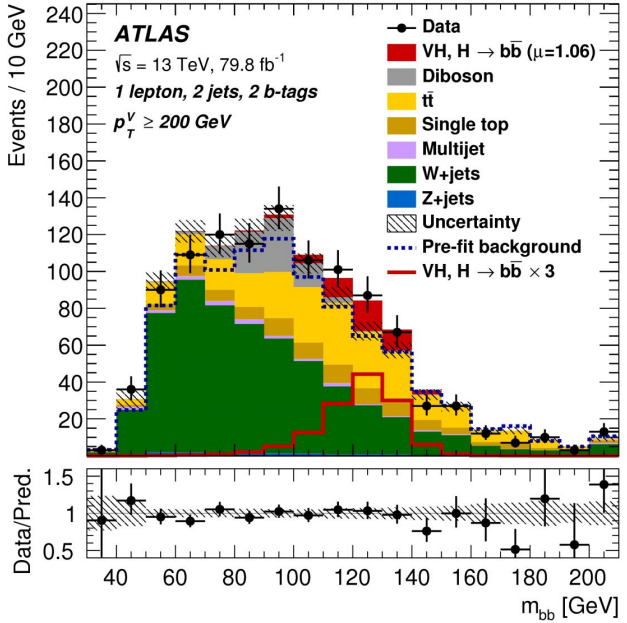
BKG + S hypothesis discarded → Exclusion regions on the specific model parameter space (with possibility of recasting)

BKG + S measured → Parameter estimation



Example from an ATLAS search





Motivation

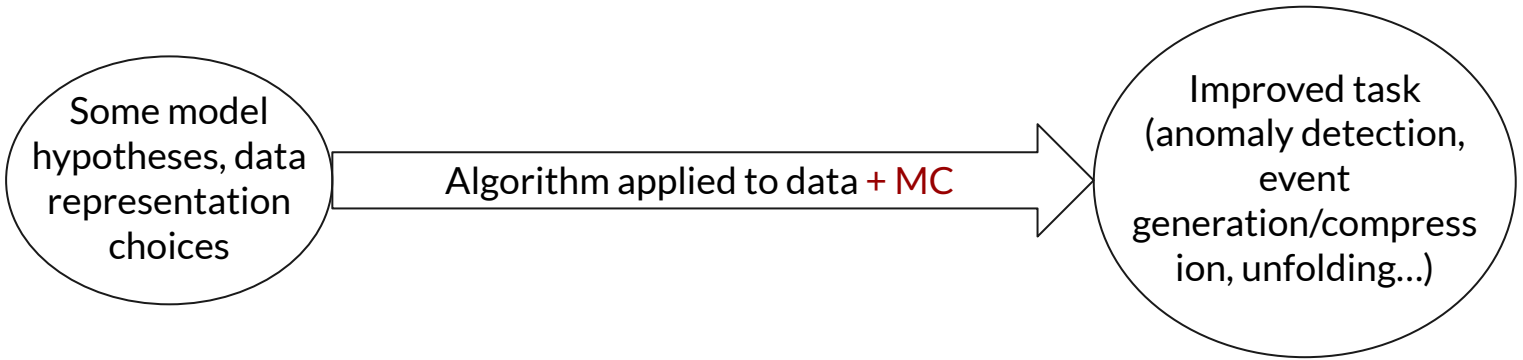
With colliders, we have been able to test our predictions to astounding precision. The discovery of the Higgs for instance, and the incredible precision in measuring its properties.

However, one may fear we're getting to the end of the line due to the energies/luminosities achievable and the systematic limitations of our current analysis methodology.

Namely, the need for precision in a high statistics environment can drive the computational costs. And there is no guarantee that the modelling can be precise enough!

Machine Learning based

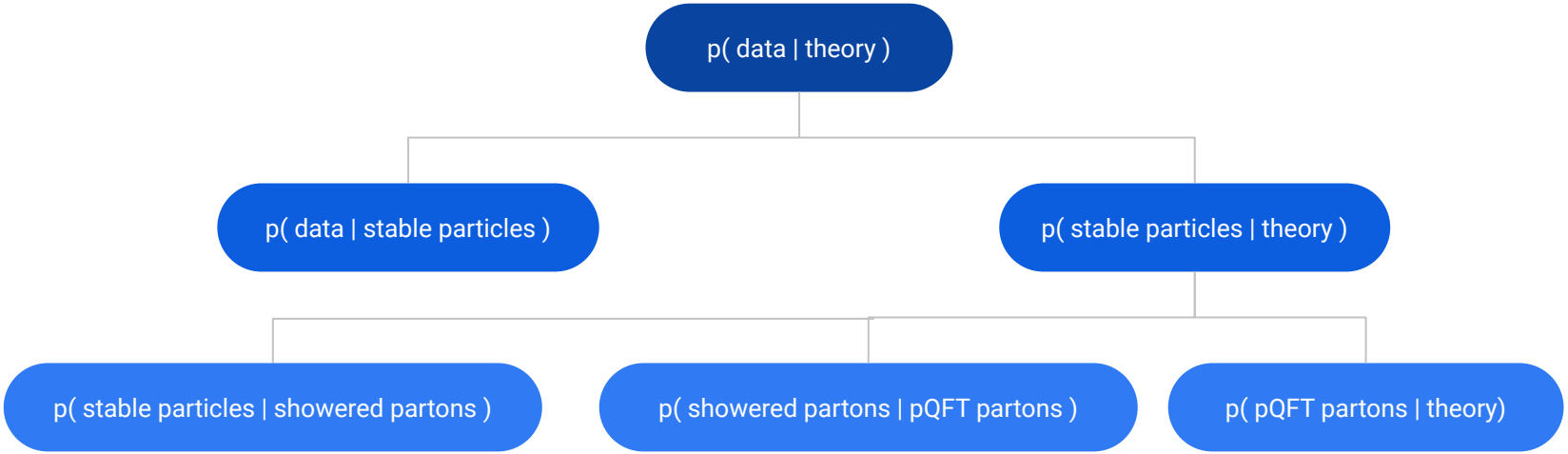
Semi-supervised and unsupervised algorithms could ease the dependency on Monte Carlo. Although there is no such thing as a free lunch.



Surrogate models

Surrogate models emulate a given process, and can be preferable to the true process probability density by a variety of reasons: accessibility, speed, storage size.

We already use empirical models for non-perturbative MC simulations (mainly, hadronization) and there are known pitfalls due to additional systematic uncertainties and unphysical behavior.


p(data | theory)
p(data | stable particles)
p(stable particles | theory)
p(stable particles | showered partons)
p(showered partons | pQFT partons)
p(pQFT partons | theory)

Bayesian techniques

They already permeate event generators and their surrogate models extensions: NNPDFs, Bayesian Neural Networks, cINN surrogate models.

Bayesian inference allows for a better evaluation of uncertainties and for improved unfolding (as long as one can make probabilistic statements).

Intuitively, Bayesian always sounds good… Now it is becoming more possible. Differentiable programming is a key development. So are different approaches to inference like Black Box Variational Inference and simulation-based inference where the intractabilities are somewhat sidestepped.

Bayesian techniques for modelling

There is a vast array of literature about Bayesian model building we can take advantage of. Already a lot of examples in HEP (e.g. GANs, VAEs, Shower Deconstruction, LDA, Topic models in general …)

Always need to keep in mind the same requirements as for event generators:

- Physically meaningful: harder to achieve than with event generators. Requires physical bias to be baked into the model.
- Speed + size: training should not be too hard nor require so much data as to render simulations preferable

Probabilistic Graphical Models

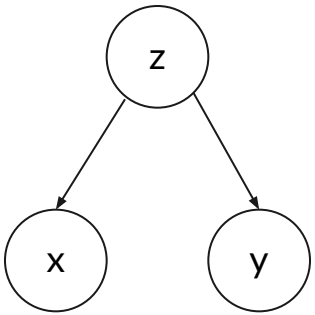
Clear, interpretable, testable way to state our modelling assumptions and incorporate physical bias.

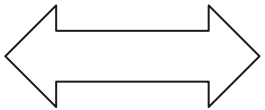
Vast literature on implementations and inference available.

Has already found many applications in collider physics (see Topic models which has been applied to quark/gluon tagging at high p_T arxiv: 2205.04459, 4-top searches arxiv:1911.09699 and quark/jet modifications Heavy Ions collisions arxiv:2008.08596).

Probabilistic Graphical Models

Association between a Graph (we'll deal with Directed Acyclical Graphs or DAGs) and a joint distribution or model. We call a DAG with an associated distribution a Bayesian Network.





$$p(x, y, z) = p(x|z)p(y|z)p(z)$$

Probabilistic Graphical Models

Powerful visualization technique to express assumptions. Useful for designing and motivating models.

Economical representation of the joint distribution that also provides insights into properties of the model, like **conditional independence** from graphical criteria like **d-separation.**

Inference can be expressed efficiently in terms of graphical manipulations.

What I'm interested in...

We can think of Bayesian Networks for unsupervised learning.

We do not learn with labels. Instead, we want to model the generative process of the data and infer the appropriate underlying classes and hopefully match them with physical processes (non-guaranteed!).

We can then run statistical tests using the learned probability densities (frequentists tests with the Likelihood, but specially Bayesian tests to qualify the learned data distributions).

A more involved problem: Four top quarks

Based on arxiv:2107.00668 by E. Alvarez, B. M. Dillon, D. A. Faroughy, J. F. Kamenik, F. Lamagna and MS.

Selecting key observables and with an appropriate modelling assumption (Conditional Independence), we are able to disentangle 4-top quark production from its main background process.

It’s not there yet, but it is an step towards surrogate models for learnable, interpretable models with verifiable assumptions.

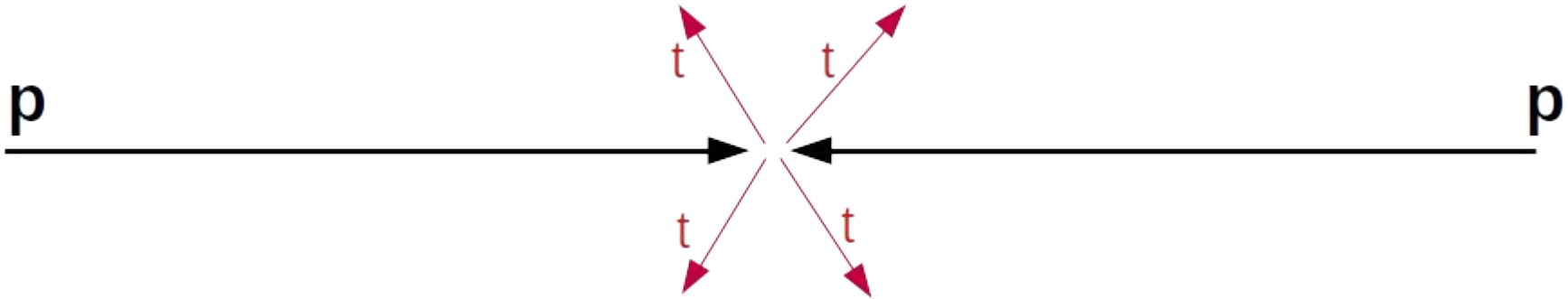
Four tops at the LHC

An increasingly sensitive SM benchmark to be explored at the LHC, with huge experimental effort and impressive results. Recent results in monolepton + 2LOS (ATLAS coll., ATLAS-CONF-2021-013, CMS coll., arxiv:1906.02805) and 2LSS + multilepton (ATLAS coll., arxiv:2007.14858. and CMS coll., arxiv:1908.06463).

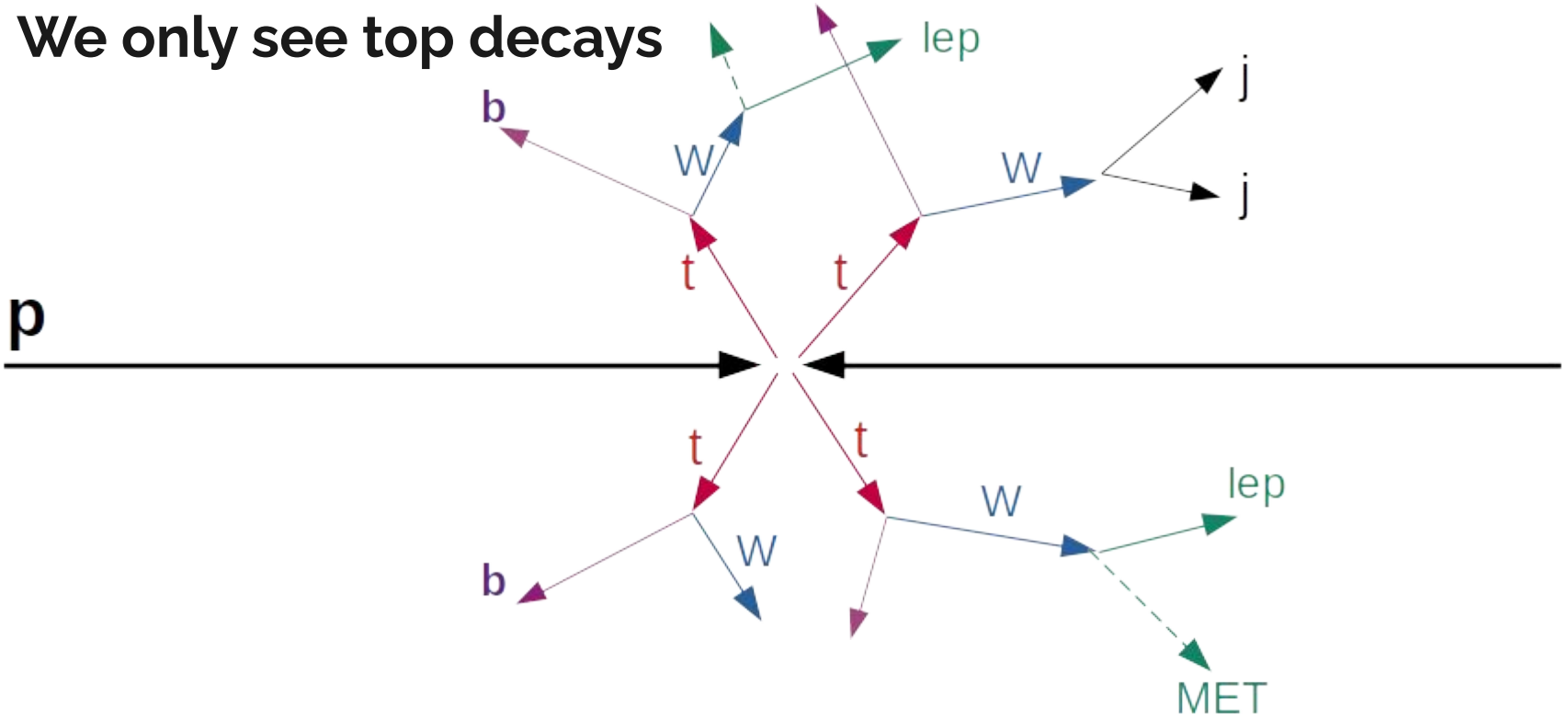
Correspondingly, state of the art calculations and improvement of SM predictions (see R. Frederix, D. Pagani, M. Zaro, arxiv:1711.02116).

A theoretically well motivated (but perhaps more importantly, still allowed) window to BSM effects (see e.g. G. Banelli, E. Salvioni, J. Serra, T. Theil, A. Weiler, arxiv:2010.05915, E. Alvarez, A. Juste, M. S., T. Vazquez Schroeder, arxiv:2011.06514 and Luc Darmé, Benjamin Fuks, Fabio Maltoni, arxiv:2104.09512).

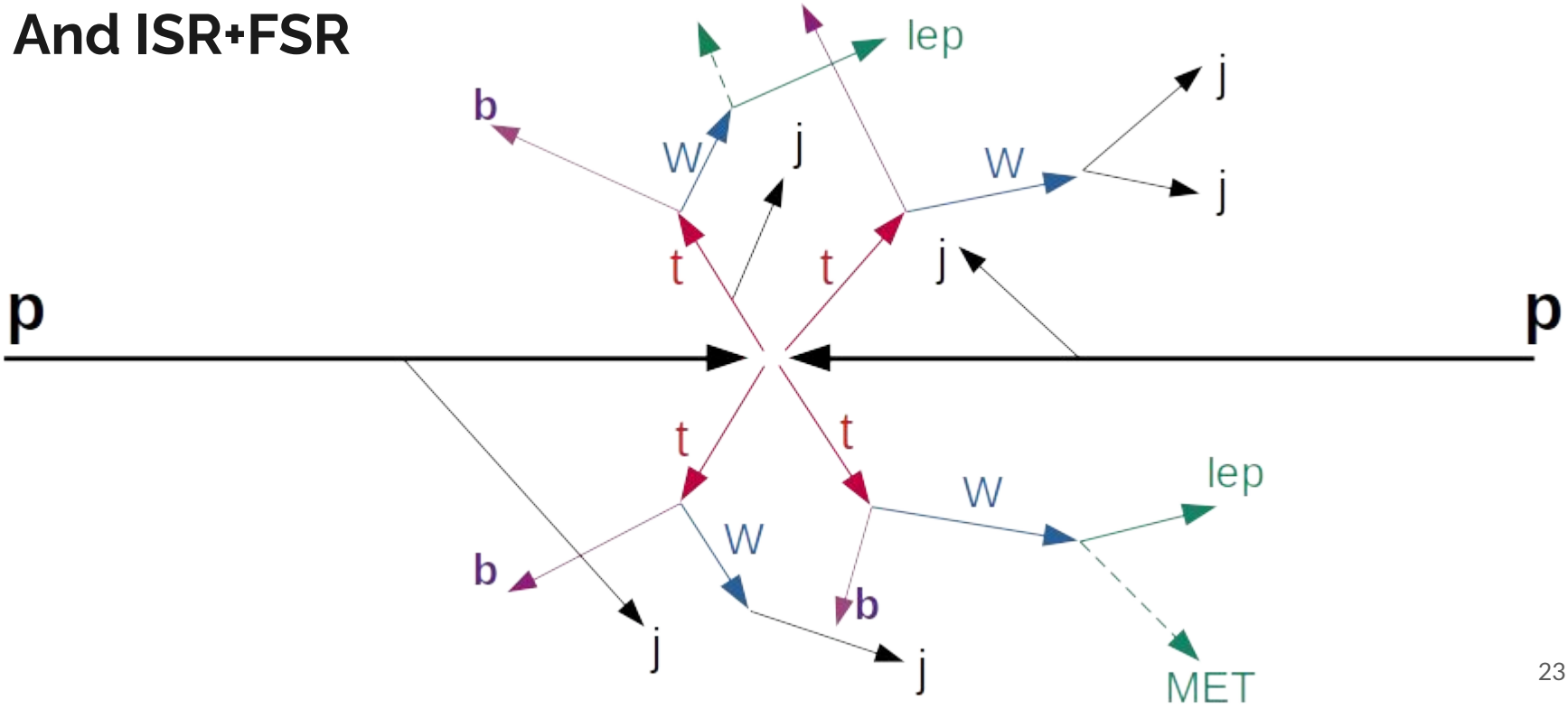
Already a very populated partonic final state



We only see top decays



And ISR+FSR



Theoretical calculations are really challenging

From R. Frederix, D. Pagani, M. Zaro, arxiv:1711.02116:

NLO corrections are large + Mixed corrections are comparable to pure QCD.

→ Theoretical calculations are expensive and necessary.

Large, accidental cancelations of (N)LO terms which involve QCD+EW couplings. Clear scale dependence of the accidental cancellations.

→ Hard to assert how BSM would change these cancellations. Need for very expensive simulations for each BSM model?

Four tops in the 2LSS+multilepton channel

Signal and irreducible backgrounds (mainly $t\bar{t}Z$, $t\bar{t}H$ and specially $\mathbf{t\bar{t}W^{\pm}}$ + heavy-flavour) are of the same order of magnitude.

Reduced complexity of the multijet final state compared to single lepton. Added complexity of other source of MET.

Still very challenging!

Four tops in the 2LSS+multilepton channel

CMS measures something along the lines of the SM cross-section: $12.6^{+5.8}_{-5.2}$ fb

While ATLAS measures 24^{+7}_{-6} fb

It's still compatible with SM predictions but the central value is roughly twice as expected

Four tops in the 2LSS+multilepton channel

The main sources of uncertainty are:

High multiplicities final states are challenging to simulate.

Misaligned charge + Fake/non-prompt leptons need to be estimated with data-driven techniques.

Charge asymmetric + High b-jet multiplicity discrepancies are addressed by using a Normalization Factor for $t\bar{t}W^{\pm}$: $NF_{ttW}=1.6\pm0.3$ for ATLAS and 1.3 ± 0.2 for CMS.

Also… not a whole lot of events (~ 300 events for $140fb^{-1}$).

Four tops in the 2LSS+multilepton channel

What can we do?

Improving on our knowledge of 4-top and its irreducible backgrounds would be ideal. However, obtaining the $t\bar{t}W^{\pm}$ +HF and 4-top cross-sections, and kinematical distributions, to a higher precision is a daunting task.

Maybe reduce Monte Carlo dependency by learning from the Signal Region directly using semi- or unsupervised techniques.

Learning four tops

We will focus on the 2LSS++ channel and consider only four tops and $t\bar{t}W^{\pm}$. $t\bar{t}Z$ and $t\bar{t}H$ could be included easily.

In this channel, we have roughly 1/1 events.

Let’s focus on the technique more than in the simulation dataset.

Learning four tops

We can resume our goal as the following:

Consider our imperfect simulations of the (known) physical processes as **prior knowledge** and **update** our knowledge using the **measured data** → Bayesian framework

Learning four tops

As seen in E. Alvarez, D. A. Faroughy, J. F. Kamenik, R. Morales, A. Szynkman in arxiv:1611.05032, **N_b and N_j** are the low level observables that drive the discriminatory power.

Let's focus on those! **Keep things simple.**

We consider a probabilistic mixture model → each event is generated by one of underlying the physical processes.

Because we cannot observe this assignment, it is a **latent parameter** z.

Modelling four tops

For event n, we measure $N_j = j_n$ and $N_b = b_n$.

We want to model $p(j_n, b_n)$ as a mixture of two processes: background ($t\bar{t}W^{\pm}$) and signal (four tops). This is achieved by writing the likelihood as:

$$p(j_n, b_n) = \sum_t p(j_n, b_n | z_n = t) p(z_n = t)$$

Modelling four tops

$p(z_n = \text{four tops})$ is the probability of an event originating from a four tops hard process

$\rightarrow S/(S+B) = p(z_n = 1) = \pi_1$

$p(j_n, b_n | z_n = t)$ are the probability mass functions we would **ideally** obtain from simulation

$\rightarrow (j, b) \sim \text{Multinomial}(\gamma_{t,(j,b)})$ where $\gamma_{t,(j,b)}$ is a matrix of dimension $2 \times (d_j \times d_b - 1)$

Modelling four tops

However, we do not really know any of these parameters.

(Roughly) Frequentist approach: obtain the best parameters using a Max Likelihood fit.

(Roughly) Bayesian approach: treat these parameters as random variables with prior probability distributions that need to be updated with the data as encoded in the likelihood (Bayes theorem).

Modelling four tops

Posterior = Likelihood x Prior / Evidence

Modelling four tops

$$p(\{\gamma_{t,(j,b)}, \pi_t\}|j,b) = [\sum_m p(j,b|\gamma_{m,(j,b)}, z=m)\pi_m] p(\{\gamma_{t,(j,b)}, \pi_t\})/p(j,b)$$

Modelling four tops

But there is a problem: We have $2x(d_j x d_b - 1)+1$ parameters to infer. But we have one measurement per event → We do not have enough information to disentangle!

We can solve this by assuming conditional independence between N_j,N_b. That is:

$$p(j, b|z = t) = p(j|z = t)p(b|z = t) = \text{Multi}(\alpha_t)\text{Multi}(\beta_t)$$

Modelling four tops

We are assuming that the correlations between N_j and N_b come from the fact that there is a mixture of processes. We need to learn this mixture to learn the correlations. If supervised, it would be Naive Bayes.

This is very different from:

$$p(j,b|z=t)p(t) = p(j|z=t)p(b|z=t)p(t) \neq p(j)p(b)$$

Modelling four tops

The likelihood is now:

$$p(j,b) = \sum_t p(j|z=t)p(b|z=t)p(t) = \sum_t \pi_t \alpha_{t,j} \beta_{t,b}$$

Which implies an specific covariance matrix between j and b:

$$C_{j,b} = \sum_{t,t'} (\pi_t \delta_{t,t'} - \pi_t \pi_{t'}) \alpha_{t,j} \beta_{t',b}$$

Modelling four tops

We go from $2x(d_j x d_b - 1)+1 \rightarrow 2x(d_j + d_b - 2)+1$ parameters.

So now Bayes theorem looks like...

$$p(\{\alpha_{t,j}, \beta_{t,b}, \pi_t\}|j,b) = [\sum_m \pi_m \alpha_{m,j} \beta_{m,b}] \prod_{m'} p(\pi_{m'}) p(\alpha_{m',j}) p(\beta_{m',b}) / p(j,b)$$

Modelling four tops

We need to specify the priors. We consider the conjugate prior of the Multinomial: the Dirichlet.

$$\mathrm{Dir}(\theta|\eta) = \prod_{\nu} \theta_{\nu}^{\eta_\nu - 1} / \mathrm{B}(\eta)$$

We map our prior knowledge to the Dirichlet hyperparameters.

The MC simulations yield estimations on the parameters θ = π, α, and β $\rightarrow$ Expected values under the prior distribution

Modelling four tops

We can fix the expected values given by our MC by parameterizing $\eta_k = \Sigma\, p_k$ where p_k is the parameter value estimated through MC simulations and Σ a total scaling factor which encodes our confidence in the prior estimations.

Looking at the mean and variance of a given possible outcome θ_k:

$$\mathrm{E}[\theta_\nu]_\eta = p_\nu;\ \mathrm{Var}[\theta_\nu]_\eta = \frac{p_\nu(1-p_\nu)}{\Sigma}$$

Putting it all together: Generative process

We can understand it using a **plate diagram** whic encodes the **generative process** of the data:

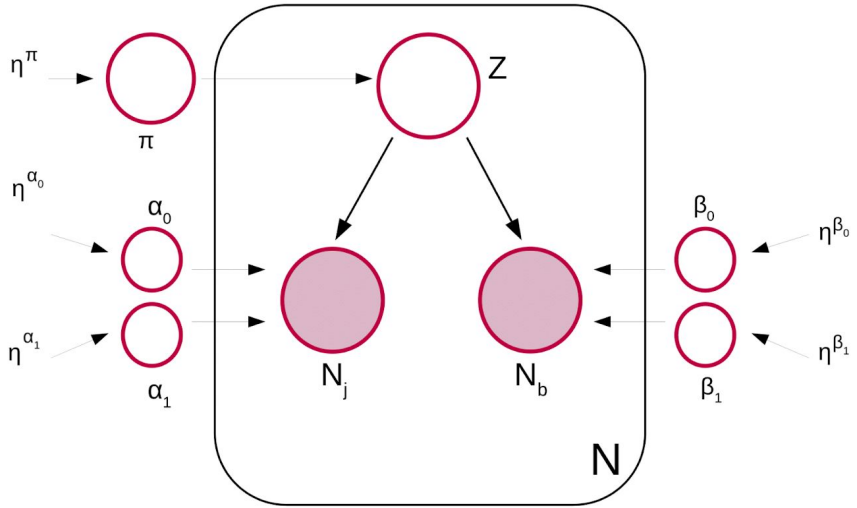
Sample fractions $\pi_0, \pi_1 \sim \text{Dir}(\eta^{\pi})$

For t=1,2:

- Sample light jet multinomials $\alpha_t \sim \text{Dir}(\eta^{\alpha t})$
- Sample b-jet $\beta_t \sim \text{Dir}(\eta^{\beta t})$

For event n=1,..,N:

- Sample event assignment $z_n \sim \text{Multi}(\pi_0, \pi_1$
- Sample $j_n \sim \text{Multi}(\alpha_{zn})$
- Sample $b_n \sim \text{Multi}(\beta_{zn})$



Learning four tops

The thing is: we cannot do the inference procedure exactly. The evidence is intractable: a lot of possible assignments and thus updates of the prior.

Luckily, there is a vast literature on the subject and a lot of techniques. EM+priors for finding the MAP, Variational Inference for approximate, fast and analytical inference and Markov Chain Monte Carlos for “exact” numerical inference.

Our model is so simple (everything is either multinomial or dirichlet!) that we are able to write the latter in python without the need to resort to dedicated software (such as pymc, emcee or pyro).

A simple benchmark

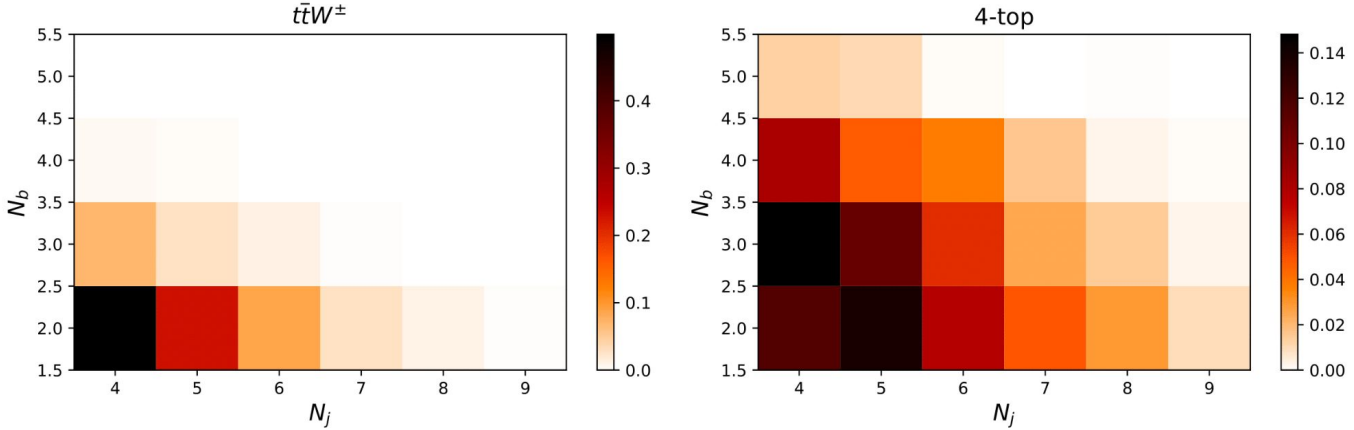
We consider the case with N=500, f_1=0.30 (roughly a Luminosity of 800 fb^{-1}).

To study our algorithm we use MC samples as “true data” and a smeared version as our “MC” which plays the role of prior.

We make an exception for the signal fractions, for which we assume no prior knowledge.

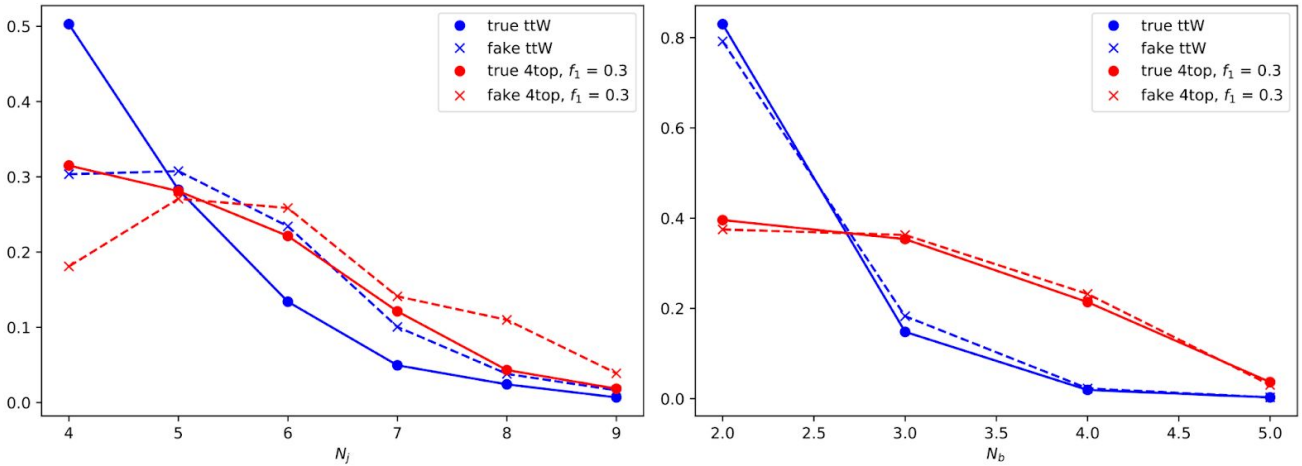
A simple benchmark

These are the two dimensional distributions



A simple benchmark

And here are the 1d projections we use.



Inference: Gibbs sampler

We obtain T “independent” samples of parameters drawn from the posterior.

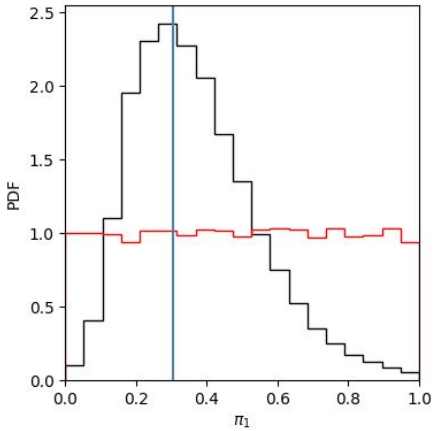
Any expected value $E_{z,\pi,\alpha,\beta}[f(z,\pi,\alpha,\beta)]$ can be approximated by the mean over the samples $\Sigma_i f(z_i,\pi_i,\alpha_i,\beta_i)/T$

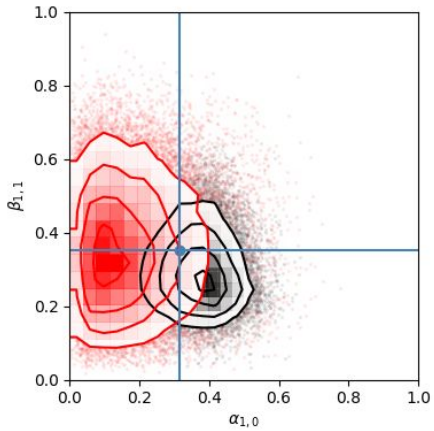
We can plot the marginalized distributions simply by drawing histograms on the relevant parameters.

We need the conditional distributions of each parameter conditioned on the others $p(\theta_v|\theta_{\setminus v})$ → Really simple because we have only Multinomials and Dirichlets.

Corner plots

For each parameter, we show its 1d histogram and its 2d correlations with every other parameter

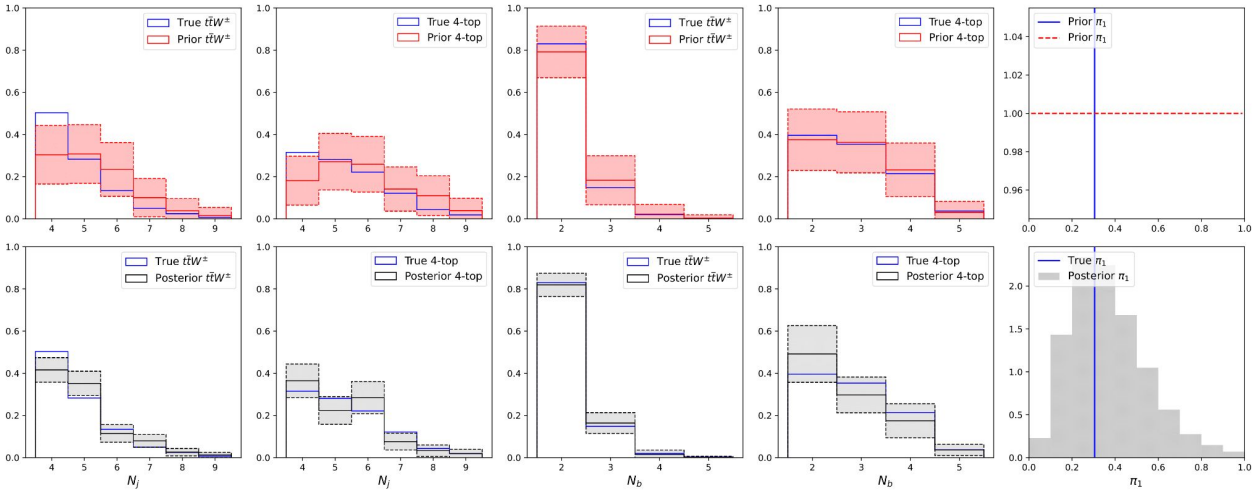




This is a particularly good example where we capture correlations

Grouping parameters together

Now we can compare prior vs posterior



General observations

From this and other benchmarks:

Good convergence + uncertainty reduction!

N_j is easier to fix. N_b is harder and π is the hardest.

N_j has a lot of very populated bins. This is not the case for N_b.

The limitations in π probably reflect the limitations of our modelling (and the use of a non-informative prior). However, once we have N_j and N_b simulations we can trust, we can obtain π in the usual manner.

As the main problem is obtaining N_j, this algorithm seems relevant.

Evaluating our results

To analyse the results of the inference, we consider the log-likelihood ratio of the correct parameters given the prior and the posterior probabilities.

In a real application, we do not know the true values. However, this is where the Bayesian framework is useful. There is a vast literature on model selection techniques!

Because we have a generative model, sanity checks and interpretability are straightforward.

Validating our model through Mutual Information

$$I(N_j, N_b) = D_{\mathrm{KL}}(p(j,b)||p(j)p(b)) = \sum_{j=4}^{9}\sum_{b=2}^{5} p(j,b)\ \mathrm{Ln}\ \frac{p(j,b)}{p(j)p(b)}$$

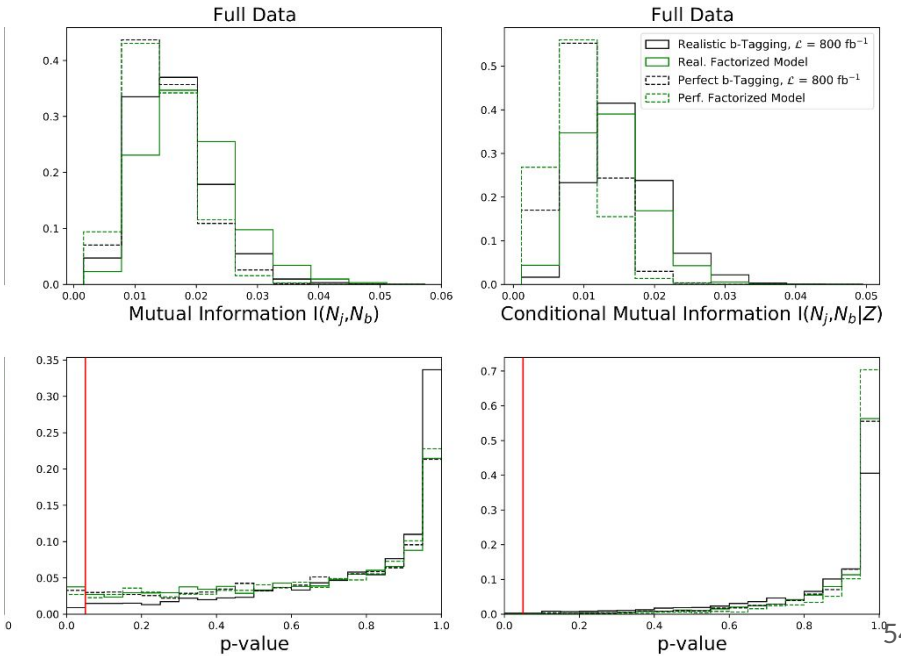
$$I(N_j, N_b|z) = D_{\mathrm{KL}}(p(j,b|z)||p(j|z)p(b|z)) = \sum_{j=4}^{9}\sum_{b=2}^{5} p(j,b|z)\ \mathrm{Ln}\ \frac{p(j,b|z)}{p(j|z)p(b|z)}$$

$$I(N_j, N_b|Z) = \sum_{z=0,1} p(z) I(N_j, N_b|z)$$

Validating our model through Mutual Information

We see how we cannot rule out $I(N_j,N_b) = 0$ or $I(N_j,N_b|Z) = 0$, but the latter is way more pronounced.

Perfect b-tagging yields better results but does not really change the picture. For limited statistics, the model works.



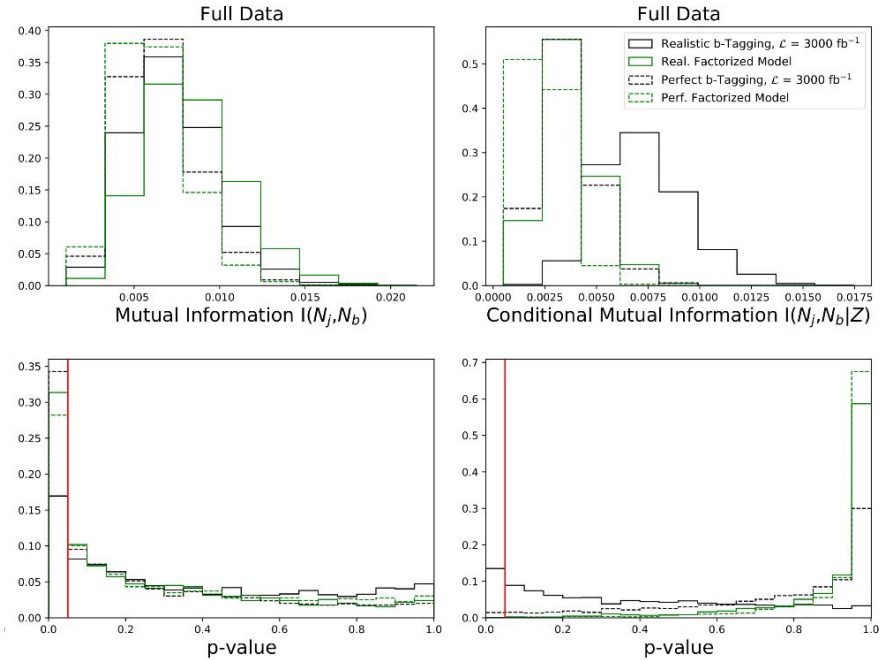
Validating our model through Mutual Information

If we increase the Luminosity to HL-LHC:

For perfect b-tagging $I(N_j,N_b) = 0$ is mostly ruled out but $I(N_j,N_b|Z) = 0$ is not.

For realistic b-tagging, $I(N_j,N_b|Z) = 0$ can become false because of induced dependence between N_j and N_b that is especially noticeable for 4-top.

For increased statistics, we should incorporate b-tagging into our model. And we can!



From this work

We are able to model the data with a generative process. Conditional independence (which is already slightly broken by imperfect b-tagging!) is a key modelling assumption that is **explicitly verifiable**.

This allows us to improve on our Monte Carlo estimations by treating them as prior knowledge which is to be updated through event measurements.

We are able to correct the N_j distributions properly.

What could we do next?

Increase the expressivity of our model by including further physical effects such as b-tagging.

Once we that we can try to:

- Tune our Monte Carlo generators on signal regions
- Measure the signal cross-section with reduced systematics
- Test for NP effects
- Adapt to other channels

What did we learn?

Two states of the art collide! It's very interesting to see what can happen.

ML can enhance statistical analyses either by taking full advantage of Monte Carlo distributions or reducing dependence on them.

I tend to value simpler models where we can incorporate domain knowledge and where unknowns are easier to catch.

However, it's hard to generalise to not so nice distributions… Deep Exponential Families and Black Box Variational Inference could provide a trade-off between power and interpretability.

Conclusions

Collider experiments are providing us with incredible amounts of measurements.

To take advantage of it, simulations and analyses techniques need to step-up.

Bayesian graphical models provide clear modelling and fast inference.

A possible tool among many, whenever it is convenient to apply them.


Thank you!

Backup slides

Complexity of collider experiments

Many different physical phenomena, with different scales and different tools:

- Partons originating from colliding hadrons sampled through PDFs
- Hard scattering amplitudes calculation and phase space integration
- ISR/FSR
- Hadronization
- Final state interactions
- Underlying event effects
- Detector effects

Making use of factorization theorems, different dedicated softwares have achieved incredible sophistication but still face difficulties

Main issues:

High-dimensional parameter space that models empirically different effects plus assume exact factorization theorems:

- Expensive tunes
- Expensive and difficult treatment of uncertainties
- Additional systematics due to modelling
- Computational bottlenecks
- Numerical instabilities
- Cross-cutting from factorization theorems' breakdowns
- Self-consistency of parameter tunes

Surrogate models

This is usually where Machine Learning can be really helpful as it can learn very precise surrogate models for different modules.

See for example (there are many, many others):

- VAEs for Jet surrogate models
- MLHAD / HADML for Pythia8 / Herwig surrogate models
- CaloGAN, CaloFlow for Geant4 calorimeter surrogate models
- OTUS, DijetGAN for End-to-end surrogate models

Surrogate models for inference

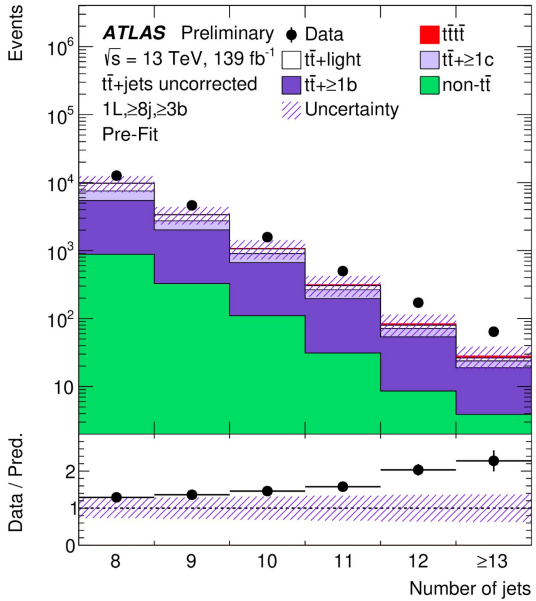
Surrogate models can make for a much easier parameter inference and unfolding.

See for example:

- MLPF for particle reconstruction from calorimeter and trackers
- OmniFold and cINN for unfolding
- MadMiner and the Matrix Element Method for parameter inference
- A. Wulzer and G. Grosso's talks

From ATLAS 1L+2LOS search

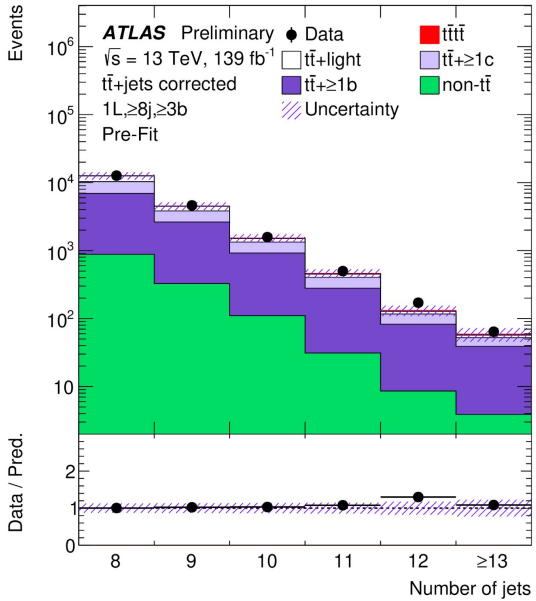
Uncertainties related to the background simulations at large N_j



From ATLAS 1L+2LOS search

Corrected through various clever techniques that need to trust that the MC extrapolates between different regions

They introduce additional systematics



Four tops in the 2LSS+multilepton channel

Let's look under the hood!

From E. Alvarez, A. Juste, M.S. and T. Vazquez Schroeder arxiv:2011.06514

Very similar strategies… but not exactly the same.

four-top-quarks	ATLAS		CMS	
	2LSS	$\geq$ 3L	2LSS	$\geq$ 3L
Total lepton charge	$\pm$2	-	$\pm$2	-
Lepton p_T [GeV]	28 (all ℓ)		25/20	25/20/20(/20)
Number of jets and b-jets	$\geq$ 6j $\geq$ 2bj (77% eff.)		$\geq$ 6j $\geq$ 2bj OR 5j $\geq$ 3bj (55-70% eff.)	$\geq$ 5j $\geq$ 2bj OR 4j $\geq$ 3bj (55-70% eff.)
H_T [GeV]	$>$ 500		$>$ 300	
$\lvert m_{e^\pm e^\pm}\rvert$ (2LSS) or $\lvert m_{OSSF}\rvert$ (3L) [GeV]	$>$ 15	-	$>$ 12	
$\lvert m_{e^\pm e^\pm} - m_Z\rvert$ (2LSS) or $\lvert m_{OSSF} - m_Z\rvert$ (3L) [GeV]	$>$ 10		-	$>$ 15
Other	-		Missing transverse momentum cuts	

Table 3: Comparison of event selections between the ATLAS [5] and CMS [6] four-top-quarks analyses. H_T is the scalar p_T sum of jets, leptons and b-jets.

Inference: Gibbs sampler

To start a given iteration t, we only need the sufficient statistics $N^{(t)}_{kjb}$.

$N^{(t)}_{k,j,b}$ is the number of measurements of j and b assigned to class k. It can be obtained merely by having the event class assignments.

From there we can obtain $N = \Sigma_k N_k = \Sigma_{k,j} N^{(t)}_{k,j} = \Sigma_{k,b} N^{(t)}_{k,b} = \Sigma_{k,j,b} N^{(t)}_{k,j,b}$

So we start the Gibbs sampler with an initial random event class assignment $Z^{(0)}$ which we'll then forget about later

Gibbs sampler

So, for iteration t:

$$\pi^{(t)} \sim \mathrm{Dir}(\{\eta_k^{\pi} + N_k^{(t-1)}, k = 1, .., K\})$$

for $k = 1, \ldots, K$

$$\alpha_k^{(t)} \sim \mathrm{Dir}(\{\eta_j^{\alpha_k} + N_{kj}^{(t-1)}, j = 1, .., d_j\})$$

$$\beta_k^{(t)} \sim \mathrm{Dir}(\{\eta_b^{\beta_k} + N_{kb}^{(t-1)}, b = 1, .., d_b\})$$

and then for n=1,...,N

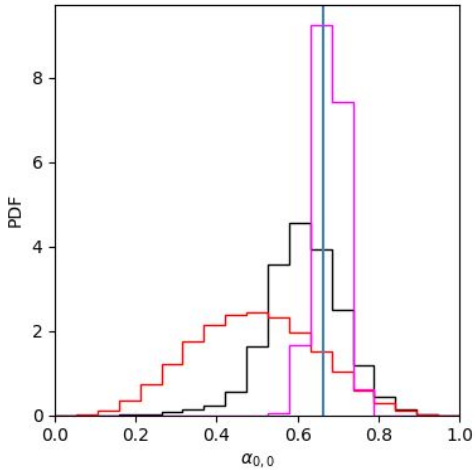
$$z_n^{(t)} \sim \mathrm{Multinomial}(\{\frac{\pi_k^{(t)} \alpha_{kj_n}^{(t)} \beta_{kb_n}^{(t)}}{\sum_{l=1}^{K} \pi_l^{(t)} \alpha_{lj_n}^{(t)} \beta_{lb_n}^{(t)}}, k = 1, \ldots, K\})$$

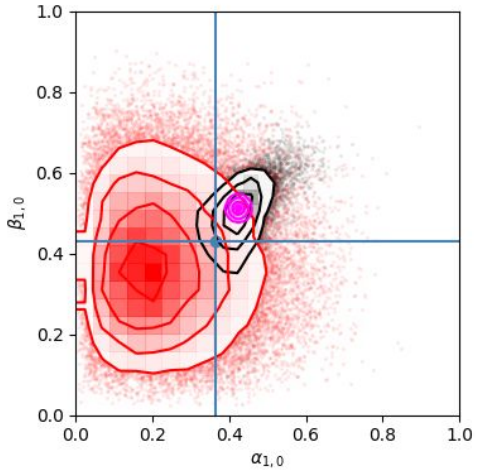
In practice you need to burn-in and thin the samples to get more or less independent samples.

Afterwards, you can get a lot of information: corrected distributions p(π,α,β|X), an event-by-event probabilistic tagger p(z_n|x_n) where we marginalize over π,α,β, etc.

Inference: Variational Inference

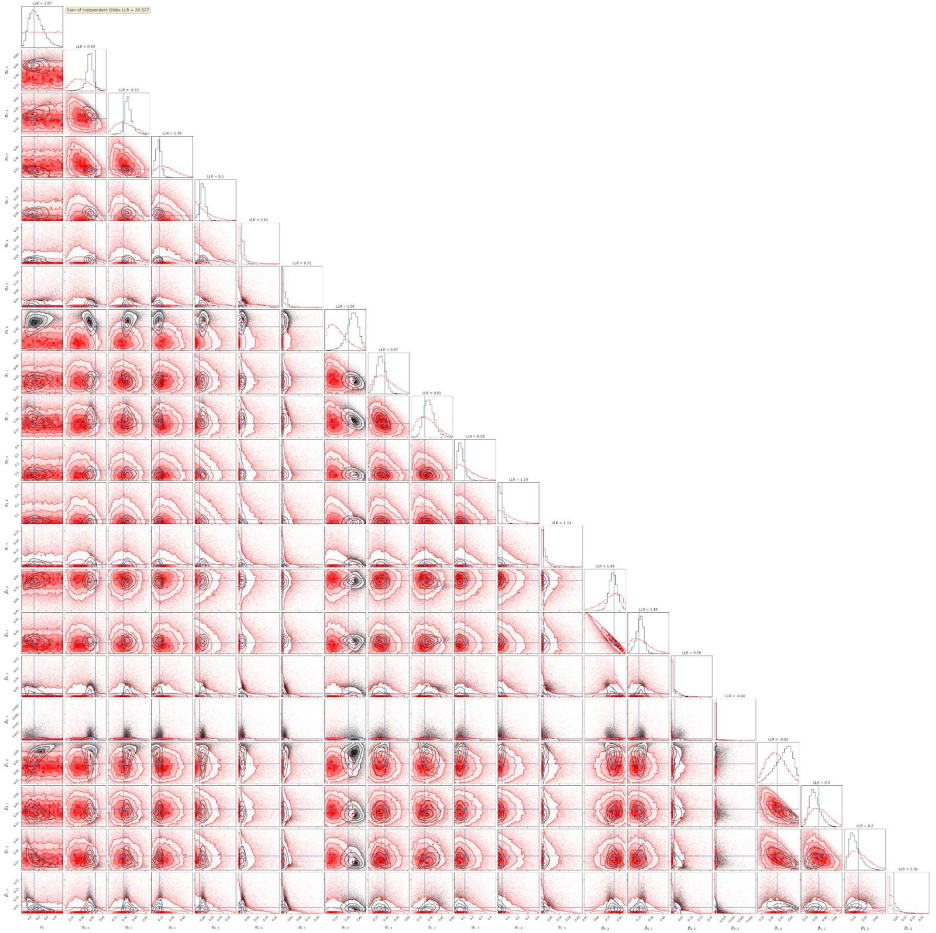
Variational Inference is an approximated inference technique which assumes certain factorizations





It is inherently limited to find what we need it to find! Distributions are too narrow

There are a lot of them...



Another example: A Quark/Gluon classifier

Unsupervised quark/gluon jet tagging with Poissonian Mixture Models

Based on E. Alvarez, M. Spannowsky and MS, arxiv:2112.11352

- Not exactly BSM search, but highly important task for a lot of analyses
- Usually quark/gluon classifiers rely on supervised datasets with approximately well-known observables
- But Monte Carlo bring uncertainties (and also the definition of quark and gluon jets can be problematic!)

Quark vs Gluon jets

Let's focus on a well-known observable, the iterative SoftDrop multiplicity with hyperparameters , and (C. Frye, A. J. Larkoski, J. Thaler, and K. Zhou, arxiv:1704.06266)

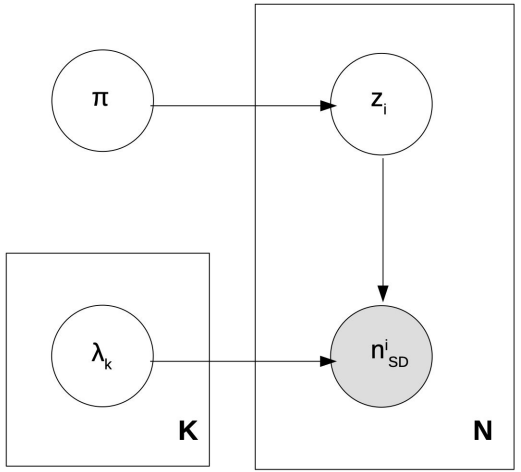
At LL order, it's Poissonean

Let's consider a mixture model that can be trained on unlabeled data! Ideally, we would match the classes to quark and gluon jets.

As a benchmark, we consider the Quark/Gluon dataset provided by P. T. Komiske, E. M. Metodiev and J. Thaler, arxiv:1810.05165

The mixture model

$$p(X) = \prod_{i=1}^{N} \sum_{k=\{q,g\}} \pi_k \text{Poisson}(n_{SD}^i; \lambda_k)$$

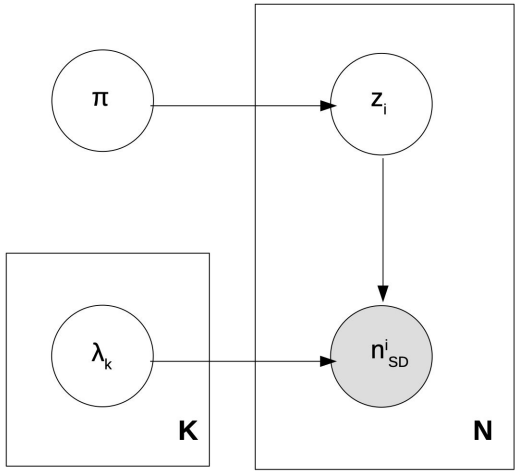


The mixture model

Here we don’t include any priors but we consider uniform priors on a limited range

Mode degeneracy is avoided by identifying gluon jets with the largest rate found

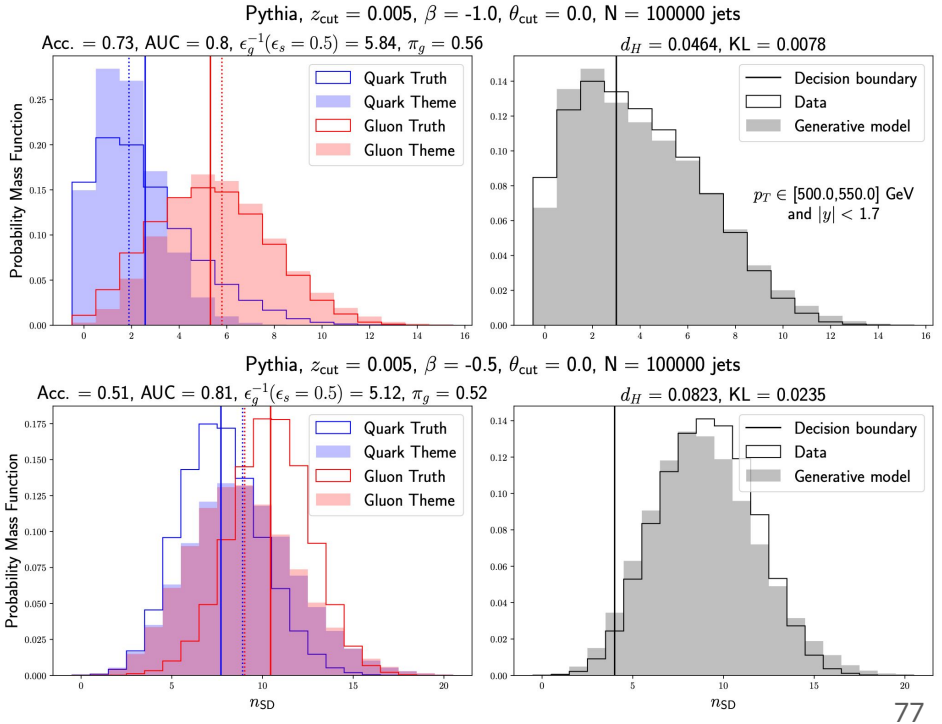
Mode collapse is (mostly) avoided because there are only two classes and the problem is rich enough



Good and bad examples

We show learned distributions for the underlying processes and for the data

Supervised metrics are for validation, but unsupervised metrics are the ones we care about.



Supervised and Unsupervised Metrics

We can define a quark/gluon tagger from the learned model

$$p(z = \text{quark}|n_{SD}, \pi^{\text{MLE}}, \lambda^{\text{MLE}}) = \frac{\pi_q^{\text{MLE}} \text{Poisson}(n_{SD}, \lambda_q^{\text{MLE}})}{\sum_{k=\{q,g\}} \pi_k^{\text{MLE}} \text{Poisson}(n_{SD}, \lambda_k^{\text{MLE}})}$$

We can also obtain the learned data probability density

$$p(n_{\text{SD}}|\text{data}) = \pi_g^{\text{MLE}} \, \text{Poisson}(\lambda_g^{\text{MLE}}) + (1.0 - \pi_g^{\text{MLE}}) \, \text{Poisson}(\lambda_q^{\text{MLE}})$$

Supervised and Unsupervised Metrics

With a tagger we can compute the usual supervised metrics (accuracy, AUC, etc)

But more interestingly, we can compare the learned data density with the measured density to obtain unsupervised metrics

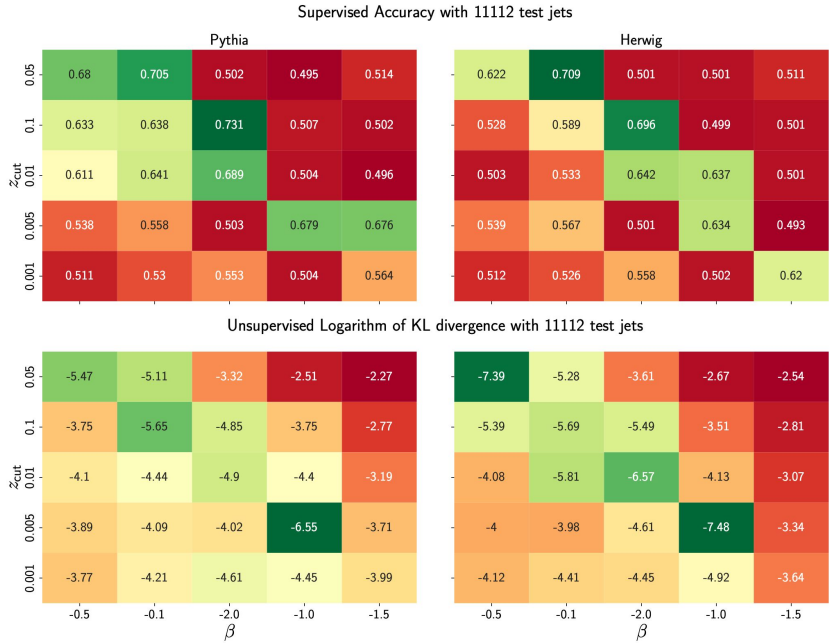
$$d_H(p,q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{n_{SD}=0}^{\infty}(\sqrt{p(n_{SD})} - \sqrt{q(n_{SD})})^2}$$

$$\mathrm{KL}(p||q) = -\sum_{n_{SD}=0}^{\infty} p(n_{SD})\mathrm{Ln}\left(\frac{q(n_{SD})}{p(n_{SD})}\right)$$

Supervised vs Unsupervised metrics

There is a good correlation between supervised and unsupervised metrics

They allow us to select hyperparameters where the model is good at explaining the data AND at distinguishing quarks from gluons

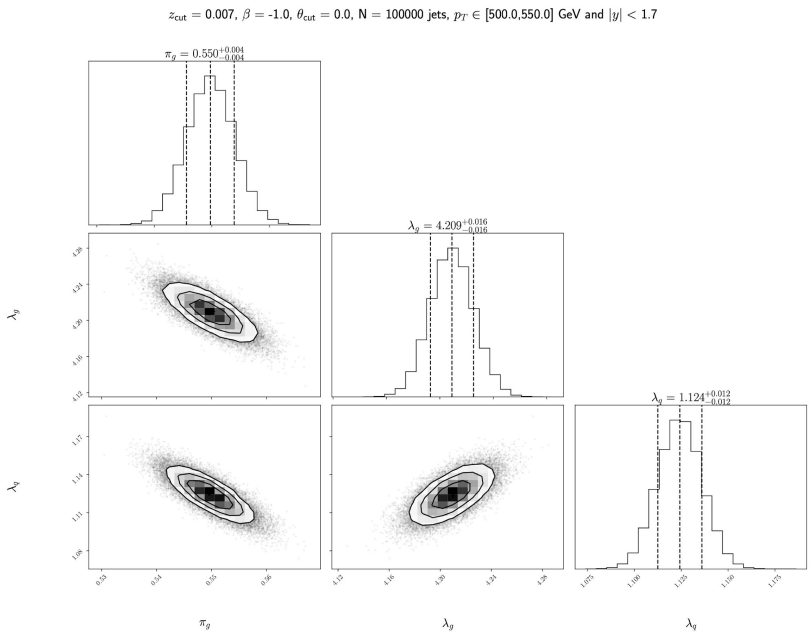


Full Bayesian analysis

We can go from MLE to full Bayesian analysis (with uniform priors)

We observe the correlations between random variables

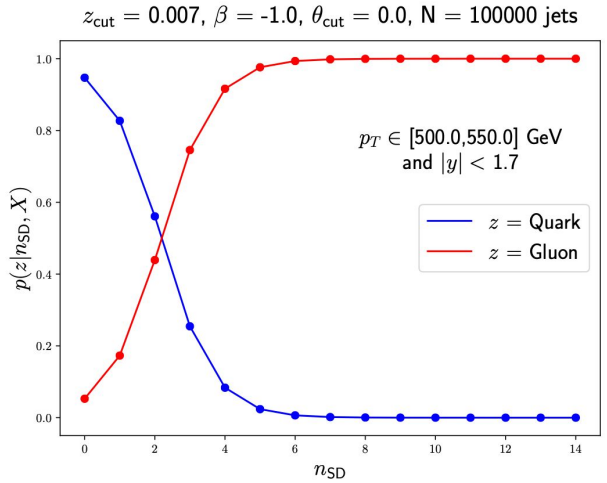
But very narrow distributions!



We can also marginalize to get a better tagger

This is a tagger that considers the full posterior distribution

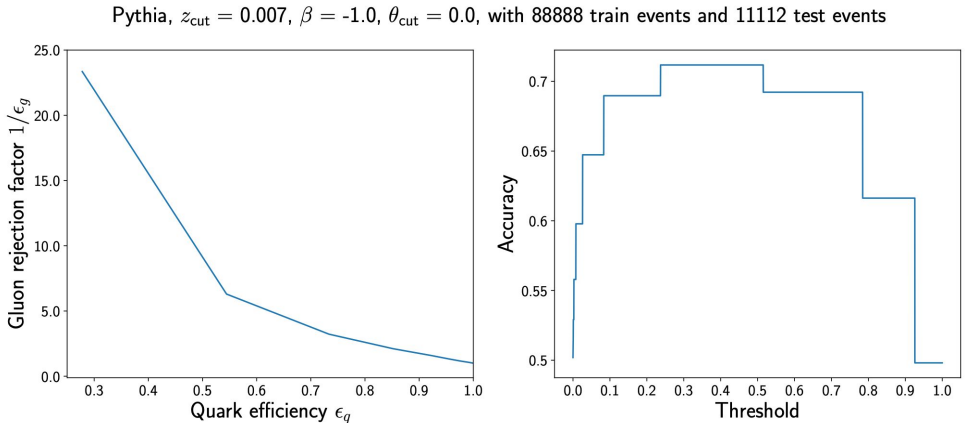
$$p(z|n_{\mathrm{SD}}, X) \approx \frac{1}{T}\sum_{t=1}^{T} p(z|n_{\mathrm{SD}}, \pi_g^t, \lambda_g^t, \lambda_q^t)$$



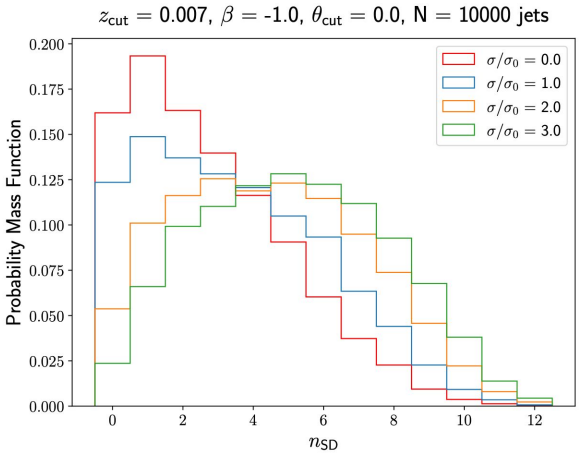
The classifier performance

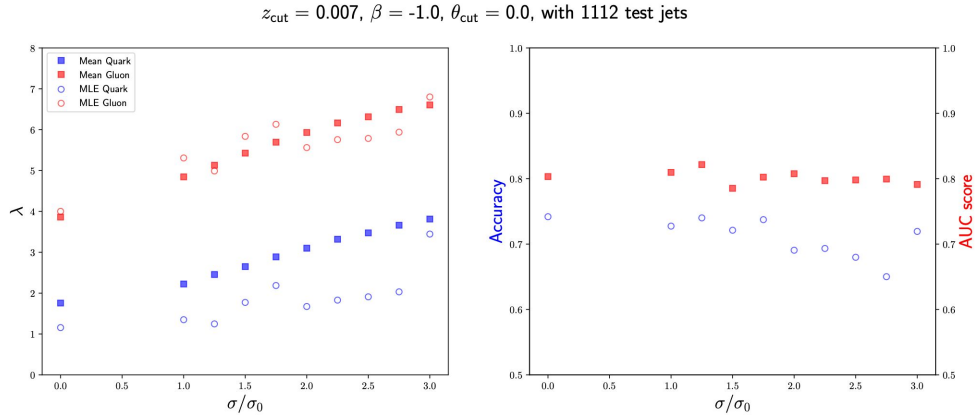
We observe how the performance focuses on the probabilistic threshold

Remember, this is unsupervised learning.

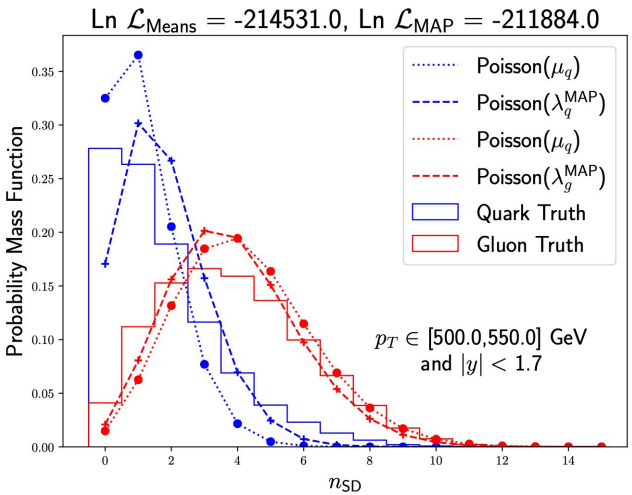


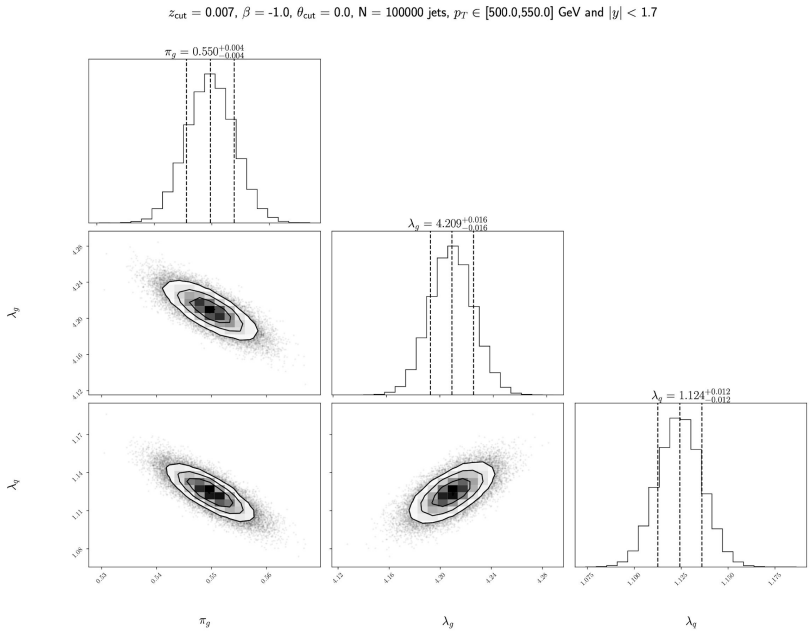
Detector Effects





Full Bayesian Analysis





So... what did we learn?

We can obtain a quark/gluon classifier directly from data assuming a Poisson mixture model.

This classifier can be optimized with data-driven metrics, resulting in accuracy in the 0.65-0.70 range.

This classifier is robust against detector effects.

We could incorporate these unsupervised methods to traditional analyses (either at the Likelihood level or by computing Bayesian tests)

Main issues

The dataset provides a narrow bin. For a more realistic implementation, should be included either as a latent variable or by binning the dataset into different subsets.

The Poisson hypothesis is only approximately true… How do we deal with deviations? It is not that important for tagging but it is for Monte Carlo tuning.

All in all

A good, solid, unsupervised classifier which is really easy and cheap to implement

It can be extended with other observables provided we have some good understanding on how to model them

Can be part of a functional definition of quarks and gluons

We have not implemented the learned distributions in statistical tests as we are dealing with a classification problem, but dealing with BSM searches would be a different matter