Theory and Machine Learning

Tilman Plehn

Universität Heidelberg

HKUST IAS, January 2022

First-principle simulations and back

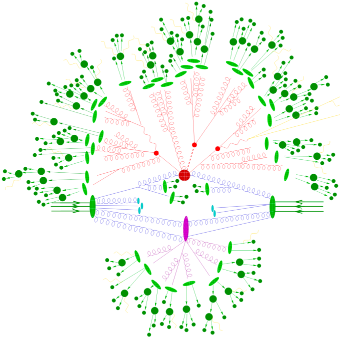
Fundamental questions — why we are running LHC

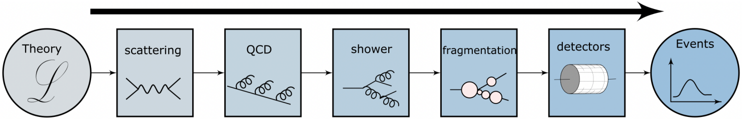
- particle nature of dark matter?
- origin of the Higgs mechanism? [hierarchy problem?]
- matter-antimatter asymmetry? [CP-symmetry]
- Standard Model all there is?

Simulation-based inference

- start with Lagrangian, perturbative QFT
- simulate events [Sherpa, Madgraph, Pythia, Powheg]
- simulate detectors

⇒ Predict LHC events in virtual worlds





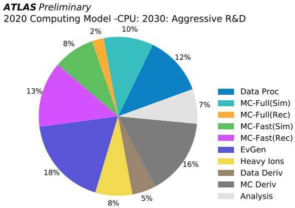
First-principle simulations and back

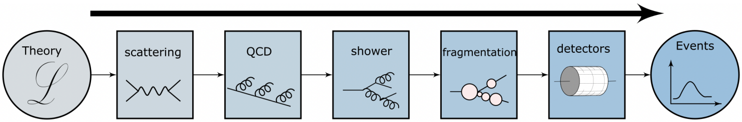
Fundamental questions — why we are running LHC

– particle nature of dark matter?
– origin of the Higgs mechanism? [hierarchy problem?]
– matter-antimatter asymmetry? [CP-symmetry]
– Standard Model all there is?

1– Forward LHC simulations

– HL-LHC: preparing for 25-fold data set
– simulated event numbers $\sim$ expected events
– statistics requiring 1%-2% uncertainty [NNLO/N^{3}LO]
– flexible signal hypotheses [time-dependent]
– low-rate high-multiplicity backgrounds
$\Rightarrow$ Event generation limiting factor





First-principle simulations and back

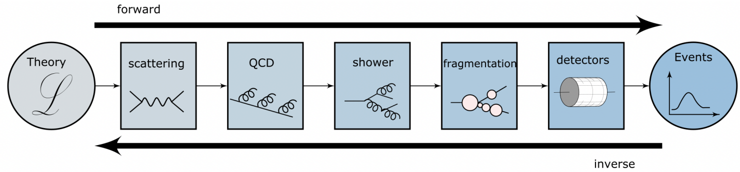
Fundamental questions — why we are running LHC

– particle nature of dark matter?
– origin of the Higgs mechanism? [hierarchy problem?]
– matter-antimatter asymmetry? [CP-symmetry]
– Standard Model all there is?

2– Inverted LHC simulations

– unfolding QCD-shower to hard parton standard [jet algorithm]
unfolding detector common
unfolding top-quark decays useful
matrix element method complete unfolding

⇒ Maybe benefit from NN-concepts [Omnifold, cINN]



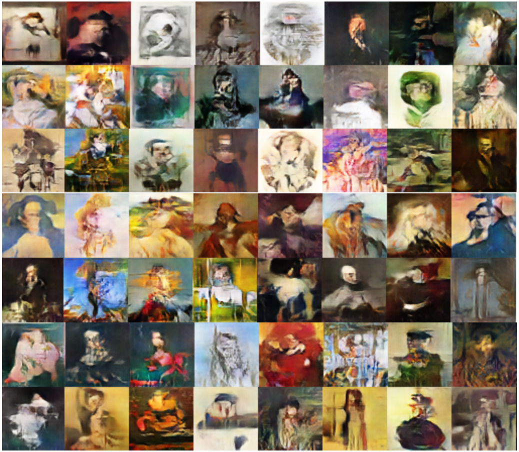
Generative networks

GANGogh [Bonafilia, Jones, Danyluk (2017)]

– can networks create new pieces of art?
  map random numbers to image pixels

– train on 80,000 pictures [organized by style and genre]

– generate portraits



Generative networks

GANGogh [Bonafilia, Jones, Danyluk (2017)]

- can networks create new pieces of art?
  map random numbers to image pixels
- train on 80,000 pictures [organized by style and genre]
- generate portraits

Edmond de Belamy [Caselles-Dupre, Fautrel, Vernier (2018)]

- trained on 15,000 portraits
- sold for $432.500

⇒ ML about marketing and sales

Generative networks

GANGogh [Bonafilia, Jones, Danyluk (2017)]

– can networks create new pieces of art?
map random numbers to image pixels

– train on 80,000 pictures [organized by style and genre]

– generate portraits

Edmond de Belamy [Caselles-Dupre, Fautrel, Vernier (2018)]

– trained on 15,000 portraits

– sold for $432.500

⇒ ML about marketing and sales

LHC applications

– jets [Nachman (2017), Carrazza-Dreyer (2019)...]

– LHC events [Butter (2019), Review (2020)...]

– inversion/unfolding [Omnifold, cGAN, cINN (2019/2020)]

– inference [QCD splittings (2020)...]

– parton density compression [Rabemananjara (2021)]

...

Precision generator

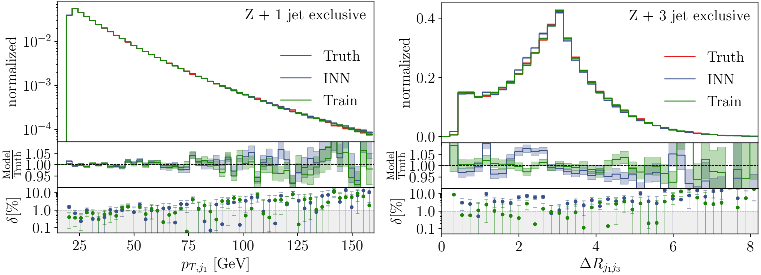
Challenging ML-event generators [my favorite playground]

– training from event samples
  no energy-momentum conservation
  no detector effects [sharper structures]

– benchmark processes
  $t\bar{t} \rightarrow 6$ jets [resonance structure]
  $Z_{\mu\mu} + \{1, 2, 3\}$ jets [Z-peak, variable jet number, jet-jet topology]

– from GANs to normalizing flows/INNs [Butter, Heimel, Hummerich, Krebs, TP, Rousselot, Vent]

$\Rightarrow$ Precision-wise getting there...

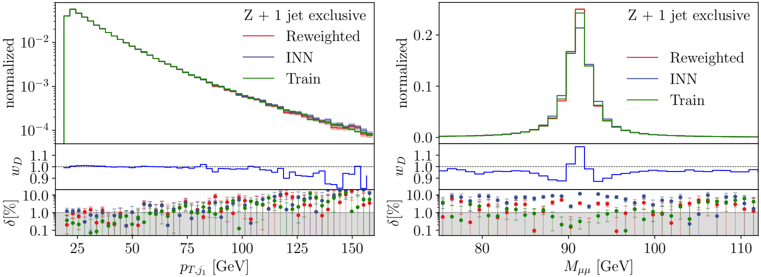


Controlled precision generator

Additional discriminator: training vs generated

- input $\{p_T, \eta, \phi, M, M_{\mu\mu}, \Delta R\}$
  output $D = 0(\text{generator}), 1(\text{truth})$
- decent generator training $D \approx 0.5$
- additional event weight $w_D = D/(1-D) \to 1$

$\Rightarrow$ Dual purpose: control and reweight



Controlled precision generator

Additional discriminator: training vs generated

– input $\{p_T, \eta, \phi, M, M_{\mu\mu}, \Delta R\}$
  output $D = 0(\text{generator}), 1(\text{truth})$
– decent generator training $D \approx 0.5$
– additional event weight $w_D = D/(1-D) \to 1$

$\Rightarrow$ Dual purpose: control and reweight

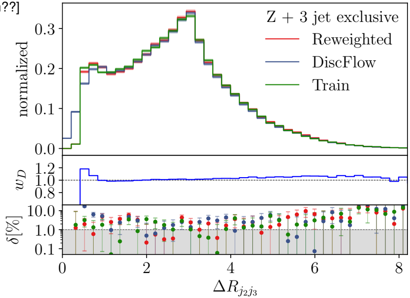
Joint DiscFlow training [GAN inspiration]

– GAN-like training unstable [Nash equilibrium??]
– coupling through weights

$$L_{\text{DiscFlow}} = -\sum_{i=1}^{B} w_D(x_i)^\alpha \; \log \frac{P(x_i)}{P_{\text{ref}}(x_i)}$$

$$\approx -\int dx \, \frac{P_{\text{ref}}^{\alpha+1}(x)}{P^\alpha(x)} \; \log \frac{P(x)}{P_{\text{ref}}(x)}$$

$\Rightarrow$ Un- and re-weighted controlled events

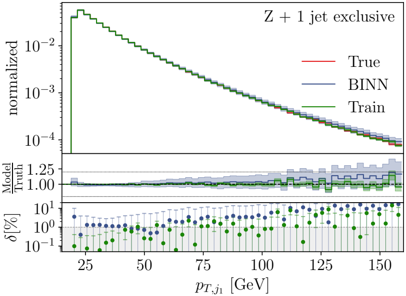


Precision generator with uncertainties

BINN generator

- Bayesian precision generator
- uncertainty over phase space
- training statistics leading source

$\Rightarrow$ Training-related error bars



Precision generator with uncertainties

BINN generator

- Bayesian precision generator
- uncertainty over phase space
- training statistics leading source

$\Rightarrow$ Training-related error bars

Theory uncertainties

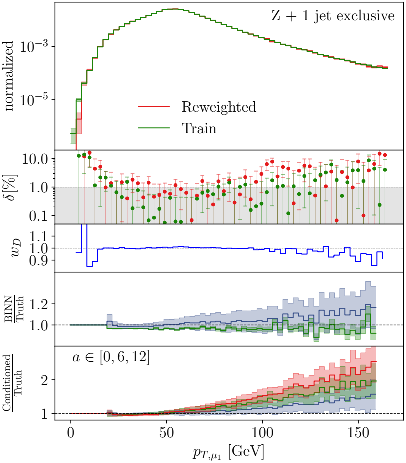
- BNN regression/classification:
  systematics from data augmentation
- systematic uncertainties in tails

$$w = 1 + a \left(\frac{p_{T,j_1} - 15 \;\; \text{GeV}}{100 \;\; \text{GeV}} \right)^2$$

augment training data $[a = 0 \dots 30]$

- train conditionally on smeared a
  error from sampling a

$\Rightarrow$ Systematic/theory error bars



Optimal observables

Measure model parameter θ optimally [Atwood-Soni, Diehl-Nachtmann, Davier etal]

– single-event likelihood [from Monte Carlo]

$$p(x|\theta) = \frac{1}{\sigma_{\text{tot}}(\theta)} \frac{d^d\sigma(x|\theta)}{dx^d}$$

– expanded locally in θ, define score [just taylor log]

$$\log \frac{p(x|\theta)}{p(x|\theta_0)} \approx (\theta - \theta_0)\, \nabla_\theta \log p(x|\theta)\Big|_{\theta_0} \equiv (\theta - \theta_0)\, t(x|\theta_0) \equiv (\theta - \theta_0)\, \mathcal{O}^{\text{opt}}(x)$$

– parton level, as used in ATLAS [CPV, Schumacher]

$$p(x|\theta) \approx |\mathcal{M}|_0^2 + \theta|\mathcal{M}|_{\text{int}}^2 \qquad \Rightarrow \qquad t(x|\theta_0) \sim \frac{|\mathcal{M}|_{\text{int}}^2}{|\mathcal{M}|_0^2} \,,$$

$\Rightarrow$ Easy at parton level, LEP physics...

Optimal observables

Measure model parameter θ optimally [Atwood-Soni, Diehl-Nachtmann, Davier etal]

– single-event likelihood [from Monte Carlo]

$$p(x|\theta) = \frac{1}{\sigma_{\text{tot}}(\theta)} \frac{d^d\sigma(x|\theta)}{dx^d}$$

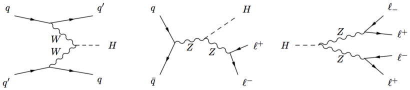
– expanded locally in θ, define score [just taylor log]

$$\log \frac{p(x|\theta)}{p(x|\theta_0)} \approx (\theta - \theta_0)\, \nabla_\theta \log p(x|\theta)\Big|_{\theta_0} \equiv (\theta - \theta_0)\; t(x|\theta_0) \equiv (\theta - \theta_0)\; \mathcal{O}^{\text{opt}}(x)$$

– parton level, as used in ATLAS [CPV, Schumacher]

$$p(x|\theta) \approx |\mathcal{M}|_0^2 + \theta |\mathcal{M}|^2_{\text{int}} \qquad \Rightarrow \qquad t(x|\theta_0) \sim \frac{|\mathcal{M}|^2_{\text{int}}}{|\mathcal{M}|^2_0} \,,$$

$\Rightarrow$ Easy at parton level, LEP physics...



Discrete symmetry [Brehmer, Kling, TP, Tait]

– CPV at dimension-6

– unique CP-observable [C-even, P-odd, $\hat{T}$-odd]

$$t \propto \epsilon_{\mu\nu\rho\sigma}\; k_1^\mu\; k_2^\nu\; q_1^\rho\; q_2^\sigma\; \text{sign}\left[(k_1 - k_2)\cdot(q_1 - q_2)\right] \xrightarrow{\text{lab frame}} \sin \Delta\phi_{jj}$$

$\Rightarrow$ Computable, modulo prefactor from D6-operator

Optimal observables

Measure model parameter θ optimally [Atwood-Soni, Diehl-Nachtmann, Davier etal]

– single-event likelihood [from Monte Carlo]

$$p(x|\theta) = \frac{1}{\sigma_{\text{tot}}(\theta)} \frac{d^d\sigma(x|\theta)}{dx^d}$$

– expanded locally in θ, define score [just taylor log]

$$\log \frac{p(x|\theta)}{p(x|\theta_0)} \approx (\theta - \theta_0)\, \nabla_\theta \log p(x|\theta)\Big|_{\theta_0} \equiv (\theta - \theta_0)\, t(x|\theta_0) \equiv (\theta - \theta_0)\, \mathscr{O}^{\text{opt}}(x)$$

– parton level, as used in ATLAS [CPV, Schumacher]

$$p(x|\theta) \approx |\mathcal{M}|_0^2 + \theta |\mathcal{M}|_{\text{int}}^2 \qquad \Rightarrow \qquad t(x|\theta_0) \sim \frac{|\mathcal{M}|_{\text{int}}^2}{|\mathcal{M}|_0^2} \,,$$

$\Rightarrow$ Easy at parton level, LEP physics...

To ML and back to formulas [Brehmer, Butter, TP, Soybelman]

– detector-level score from MadMiner

– parton-level score analytically

– good enough formula for controlled use?

$\Rightarrow$ Symbolic regression

PySR (backup slide, really)

Analytic formula for score [M Cranmer (2020)]

- function to approximate $t(x|\theta)$
- order-one phase space parameters $x_p = p_T/m_H, \Delta\eta, \Delta\phi$ [node]
- operators $\sin x, x^2, x^3, x+y, x-y, x*y, x/y$ [node]
- represent formula as tree [complexity = number of nodes]

$\Rightarrow$ figures of merit

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\Big[g_i(x) - t(x,z|\theta)\Big]^2$$

$$\text{score} \approx \text{MSE} + \text{parsimony} \cdot \text{complexity}$$

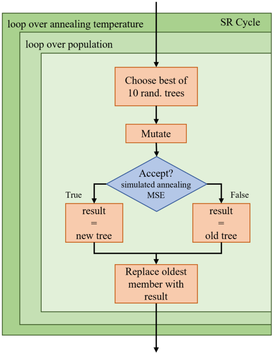
Simulated annealing

- combine trees to populations
- mutate trees exchange, add, delete nodes
- acceptance probability

$$p = \exp\left(-\frac{\text{score}_\text{new} - \text{score}_\text{old}}{\alpha T\ \text{score}_\text{old}}\right)$$

- added: proper fit of pre-factors

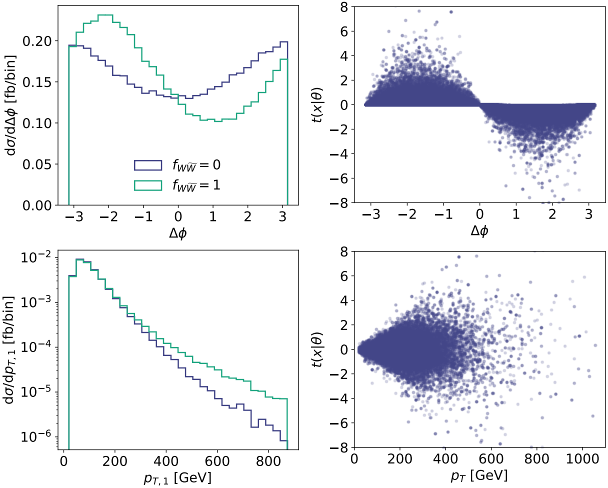
$\Rightarrow$ Hall of Fame: best equation per complexity



Score around Standard Model

Score around Standard Model

– shift in distributions, reflected in score [parton level]
CP-effect in $\Delta\phi_{jj}$
D6-effect in $p_{T,j}$



Score around Standard Model

Score around Standard Model

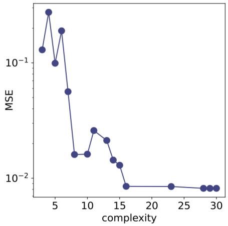
- shift in distributions, reflected in score [parton level]
  CP-effect in $\Delta\phi_{jj}$
  D6-effect in $p_{T,j}$
- best 4-parameter formula including $\Delta\eta$ [without/with detector]

$$t = -x_{p,1}\,\left(x_{p,2} + c\right)\,(a - b\Delta\eta)\,\sin(\Delta\phi + d)$$

with $a = 1.086(11)$ $b = 0.10241(19)$ $c = 0.24165(8)$ $d = 0.00662(32)$
$a = 0.926(2)$ $b = 0.08387(35)$ $c = 0.3542(20)$ $d = 0.00911(67)$

$\Rightarrow$ Mostly expected formula

compl	dof	function	MSE
3	1	$a\,\Delta\phi$	$1.30 \cdot 10^{-1}$
4	1	$\sin(a\Delta\phi)$	$2.75 \cdot 10^{-1}$
5	1	$a\Delta\phi x_{p,1}$	$9.93 \cdot 10^{-2}$
6	1	$-x_{p,1}\sin(\Delta\phi + a)$	$1.90 \cdot 10^{-1}$
7	1	$(-x_{p,1} - a)\sin(\sin(\Delta\phi))$	$5.63 \cdot 10^{-2}$
8	1	$(a - x_{p,1})x_{p,2}\sin(\Delta\phi)$	$1.61 \cdot 10^{-2}$
14	2	$x_{p,1}(a\Delta\phi - \sin(\sin(\Delta\phi)))(x_{p,2} + b)$	$1.44 \cdot 10^{-2}$
15	3	$-(x_{p,2}(a\Delta\eta^2 + x_{p,1}) + b)\sin(\Delta\phi + c)$	$1.30 \cdot 10^{-2}$
16	4	$-x_{p,1}(a - b\Delta\eta)(x_{p,2} + c)\sin(\Delta\phi + d)$	$8.50 \cdot 10^{-3}$
28	7	$(x_{p,2} + a)(bx_{p,1}(c - \Delta\phi) - x_{p,1}(d\Delta\eta + ex_{p,2} + f)\sin(\Delta\phi + g))$	$8.18 \cdot 10^{-3}$



ML for LHC Theory

ML-applications in LHC theory

- just another numerical tool for a numerical field
- driven by money from data industry, medical research
- goals are...

  ...improve established tasks
  ...develop new tools for straightforward tasks
  ...come up with and enable new ideas

1- example: controlled forward/backward simulation with uncertainties

2- example: recovering formulas from numerics

$\Rightarrow$ Opportunity for young people to make a difference!