Machine Learning — The Future of LHC Theory

Tilman Plehn

Universität Heidelberg

MCNet 6/2020

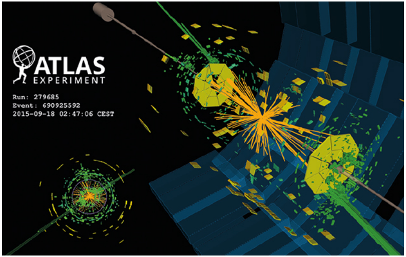
Why LHC?

Data from ATLAS & CMS

– HL-LHC = 2000 $\times$ Tevatron

– jet production $\sigma_{pp \to jj} \times \mathcal{L} \approx 10^8 \text{fb} \times 1000/\text{fb} \approx 10^{11}$ events

$\Rightarrow$ It's proper big data



Why LHC?

Data from ATLAS & CMS

– HL-LHC = 2000 $\times$ Tevatron

– jet production $\sigma_{pp\to jj} \times \mathcal{L} \approx 10^8\text{fb} \times 1000/\text{fb} \approx 10^{11}$ events

$\Rightarrow$ It's proper big data

Physics with jets

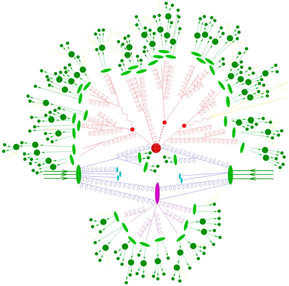
– re-summed perturbative QFT prediction from QCD

– jets as decay products

$67\%\ W \to jj \quad 70\%\ Z \to jj \quad 60\%\ H \to jj \quad 67\%\ t \to jjj \quad 60\%\ \tau \to j$ ...

– new physics in 'dark showers'

$\Rightarrow$ It's interesting



Why LHC?

Data from ATLAS & CMS

– HL-LHC = 2000 $\times$ Tevatron

– jet production $\quad \sigma_{pp \to jj} \times \mathcal{L} \approx 10^8 \text{fb} \times 1000/\text{fb} \approx 10^{11}$ events

$\Rightarrow$ It's proper big data

Physics with jets

– re-summed perturbative QFT prediction from QCD

– jets as decay products

67% $W \to jj$ 70% $Z \to jj$ 60% $H \to jj$ 67% $t \to jjj$ 60% $\tau \to j$ ...

– new physics in 'dark showers'

$\Rightarrow$ It's interesting

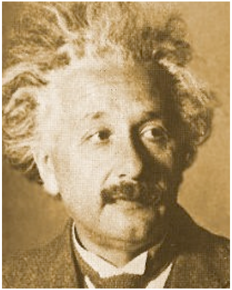
Monte Carlo data

– generators: Sherpa, Herwig, Pythia, Madgraph

– based on QFT-Lagrangian

– data-to-data comparison: MC vs LHC

$\Rightarrow$ It's properly understood



Why LHC?

Data from ATLAS & CMS

– HL-LHC = 2000 $\times$ Tevatron

– jet production $\sigma_{pp\to jj} \times \mathcal{L} \approx 10^8 \text{fb} \times 1000/\text{fb} \approx 10^{11}$ events

$\Rightarrow$ It's proper big data

Physics with jets

– re-summed perturbative QFT prediction from QCD

– jets as decay products

67% $W \to jj$ 70% $Z \to jj$ 60% $H \to jj$ 67% $t \to jjj$ 60% $\tau \to j$ ...

– new physics in 'dark showers'

$\Rightarrow$ It's interesting

Monte Carlo data

– generators: Sherpa, Herwig, Pythia, Madgraph

– based on QFT-Lagrangian

– data-to-data comparison: MC vs LHC

$\Rightarrow$ It's properly understood

Why not LHC?

ATLAS & CMS

– 3000 know-it-alls per experiment

– many just interested in detector

– top-down organized analysis groups

⇒ Shockingly little innovation

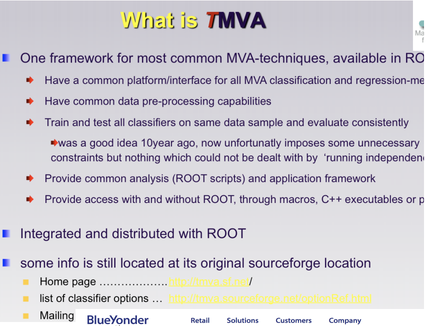
Expertize

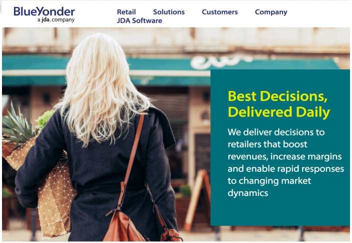
– LHC data format: ROOT

– multi-variate analyses tool: TMVA

– Tensorflow from TMVA/ROOT

⇒ Limited sense of ML-urgency





Experiment vs theory

– theorists linked to lack of team compatibility

– simulated data as good as actual data

– excellent personal ex-th connections

⇒ Someone has to drive developments...

1– Jet classification: Nothing is ever new

LHC visionaries

– 1991: NN-based quark-gluon tagger [Lönnblad, Peterson, Rögnvaldsson]

USING NEURAL NETWORKS TO IDENTIFY JETS

Leif LÖNNBLAD*, Carsten PETERSON and Thorsteinn RÖGNVALDSSON*****

Department of Theoretical Physics, University of Lund, Sölvegatan 14A, S-22362 Lund, Sweden

Received 29 June 1990

A neural network method for identifying the ancestor of a hadron jet is presented. The idea is to find an efficient mapping between certain observed hadronic kinematical variables and the quark-gluon identity. This is done with a neuronic expansion in terms of a network of sigmoidal functions using a gradient descent procedure, where the errors are back-propagated through the network. With this method we are able to separate gluon from quark jets originating from Monte Carlo generated e^+e^- events with ~85% approach. The result is independent of the MC model used. This approach for isolating the gluon jet is then used to study the so-called string effect.

In addition, heavy quarks (b and c) in e^+e^- reactions can be identified on the 50% level by just observing the hadrons. In particular we are able to separate b-quarks with an efficiency and purity, which is comparable with what is expected from vertex detectors. We also speculate on how the neural network method can be used to disentangle different hadronization schemes by compressing the dimensionality of the state space of hadrons.

1– Jet classification: Nothing is ever new

LHC visionaries

– 1991: NN-based quark-gluon tagger [Lönnblad, Peterson, Rögnvaldsson]

– 1994: jet algorithm for W, top... [Seymour]

USING NEURAL NETWORKS TO IDENTIFY JETS

Leif LÖNNBLAD *, Carsten PETERSON ** and Thorsteinn RÖGNVALDSSON ***

Department of Theoretical Physics, University of Lund, Sölvegatan 14A, S-22362 Lund, Sweden

Received 29 June 1990

A neural network method for identifying the ancestor of a hadron jet is presented. The idea is to find an efficient mapping between certain observed hadronic kinematical variables and the quark-gluon identity. This is done with a neuronic expansion in terms of a network of sigmoidal functions using a gradient descen network. With this method we ar Carlo generated e^+e^- events v model used. This approach for i effect.

In addition, heavy quarks (b a just observing the hadrons. In pa purity, which is comparable with how the neural network method compressing the dimensionality c

Searches for new particles using cone and cluster jet algorithms: a comparative study

Michael H. Seymour

Department of Theoretical Physics, University of Lund, Sölvegatan 14A, S-22362 Lund, Sweden

Received 18 June 1993; in revised form 16 September 1993

Abstract. We discuss the reconstruction of the hadronic decays of heavy particles using jet algorithms. The ability to reconstruct the mass of the decaying particle is compared between a traditional cone-type algorithm and a recently proposed cluster-type algorithm. The specific examples considered are the semileptonic decays of a heavy Higgs boson at $\sqrt{s}=16$ TeV, and of top quark–antiquark pairs at $\sqrt{s}=1.8$ TeV. We find that the cluster algorithm offers considerable advantages in the former case, and a slight advantage in the latter. We briefly discuss the effects of calorimeter energy resolution, and show that a typical resolution dilutes these advantages, but does not remove them entirely.

except that the invariant mass of a pair is replaced by the transverse momentum of the softer particle relative to the other.

More recently, this algorithm was extended to collisions with incoming hadrons [5], and a longitudinally-invariant $k_\perp$-clustering algorithm for hadron-hadron collisions was proposed [6]. This algorithm has been compared with the more commonly used cone algorithm from the viewpoints of a parton-shower Monte Carlo program [6, 7], and a fixed-order matrix-element calculation [8], and advantages of the cluster algorithm were reported in both cases. This paper is concerned with a comparison between the algorithms for the task of reconstructing the hadronic decays of heavy particles, which was also studied in a preliminary way in [9].

The only as-yet unobserved particles of the minimal Standard Model are the top quark and Higgs boson. The search for, and study of, these particles are among the most important goals of current and planned hadron-hadron collider experiments. In both cases, which decay

And it is also done

Experiments driving, for once... [Ben's talk]

– 2014/15: first jet image papers

– 2017: first (working) ML top tagger

– ML4Jets 2017: What architecture works best?

– ML4Jets 2018: Lots of architectures work [1902.09914]

⇒ Jet classification understood and done

SciPost Physics Submission

The Machine Learning Landscape of Top Taggers

G. Kasieczka (ed)[1], T. Plehn (ed)[2], A. Butter[2], K. Cranmer[3], D. Debnath[4], B. M. Dillon[5], M. Fairbairn[6], D. A. Faroughy[5], W. Fedorko[7], C. Gay[7], L. Gouskos[8], J. F. Kamenik[5,9], P. T. Komiske[10], S. Leiss[1], A. Lister[7], S. Macaluso[3,4], E. M. Metodiev[10], L. Moore[11], B. Nachman,[12,13], K. Nordström[14,15], J. Pearkes[7], H. Qu[8], Y. Rath[16], M. Rieger[16], D. Shih[4], J. M. Thompson[2], and S. Varma[6]

1 Institut für Experimentalphysik, Universität Hamburg, Germany
2 Institut für Theoretische Physik, Universität Heidelberg, Germany
3 Center for Cosmology and Particle Physics and Center for Data Science, NYU, USA
4 NHECT, Dept. of Physics and Astronomy, Rutgers, The State University of NJ, USA
5 Jozef Stefan Institute, Ljubljana, Slovenia
6 Theoretical Particle Physics and Cosmology, King's College London, United Kingdom
7 Department of Physics and Astronomy, The University of British Columbia, Canada
8 Department of Physics, University of California, Santa Barbara, USA
9 Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia
10 Center for Theoretical Physics, MIT, Cambridge, USA
11 CP3, Universitéxx Catholique de Louvain, Louvain-la-Neuve, Belgium
12 Physics Division, Lawrence Berkeley National Laboratory, Berkeley, USA
13 Simons Inst. for the Theory of Computing, University of California, Berkeley, USA
14 National Institute for Subatomic Physics (NIKHEF), Amsterdam, Netherlands
15 LPTHE, CNRS & Sorbonne Université, Paris, France
16 III. Physics Institute A, RWTH Aachen University, Germany

gregor.kasieczka@uni-hamburg.de
plehn@uni-heidelberg.de

July 24, 2019

Abstract

Based on the established task of identifying boosted, hadronically decaying top quarks, we compare a wide range of modern machine learning approaches. Unlike most established methods they rely on low-level input, for instance calorimeter output. While their network architectures are vastly different, their performance is comparatively similar. In general, we find that these new approaches are extremely powerful and great fun.

Content

And it is also done

Experiments driving, for once... [Ben's talk]

– 2014/15: first jet image papers

– 2017: first (working) ML top tagger

– ML4Jets 2017: What architecture works best?

– ML4Jets 2018: Lots of architectures work [1902.09914]

$\Rightarrow$ Jet classification understood and done

SciPost Physics Submission

The Machine Learning Landscape of Top Taggers

G. Kasieczka (ed)[1], T. Plehn (ed)[2], A. Butter[2], K. Cranmer[3], D. Debnath[4], B. M. Dillon[5], M. Fairbairn[6], D. A. Faroughy[5], W. Fedorko[7], C. Gay[7], L. Gouskos[8], J. F. Kamenik[5,9], P. T. Komiske[10], S. Leiss[1], A. Lister[7], S. Macaluso[3,4], E. M. Metodiev[10], L. Moore[11], B. Nachman,[12,13], K. Nordström[14,15], J. Pearkes[7], H. Qu[8], Y. Rath[16], M. Rieger[16], D. Shih[4], J. M. Thompson[2], and S. Varma[6]

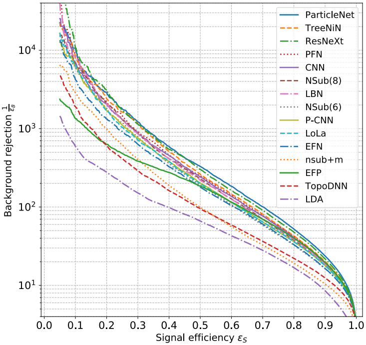
1 Institut für Experimentalphysik, Universität Hamburg, Germany
2 Institut für Theoretische Physik, Universität Heidelberg, Germany
3 Center for Cosmology and Particle Physics and Center for Data Science, NYU, USA
4 NHECT, Dept. of Physics and Astronomy, Rutgers, The State University of NJ, USA
5 Jozef Stefan Institute, Ljubljana, Slovenia
6 Theoretical Particle Physics and Cosmology, King's College London, United Kingdom
7 Department of Physics and Astronomy, The University of British Columbia, Canada
8 Department of Physics, University of California, Santa Barbara, USA
9 Faculty of Mathematics and Physics, University of Ljubljana, Ljubljana, Slovenia
10 Center for Theoretical Physics, MIT, Cambridge, USA
11 CP3, Universitéxx Catholique de Louvain, Louvain-la-Neuve, Belgium
12 Physics Division, Lawrence Berkeley National Laboratory, Berkeley, USA
13 Simons Inst. for the Theory of Computing, University of California, Berkeley, USA
14 National Institute for Subatomic Physics (NIKHEF), Amsterdam, Netherlands
15 LPTHE, CNRS & Sorbonne Université, Paris, France
16 III. Physics Institute A, RWTH Aachen University, Germany

gregor.kasieczka@uni-hamburg.de
plehn@uni-heidelberg.de

July 24, 2019

Abstract

Based on the established task of identifying boosted, hadronically decaying top quarks, we compare a wide range of modern machine learning approaches. Unlike most established methods they rely on low-level input, for instance calorimeter output. While their network architectures are vastly different, their performance is comparatively similar. In general, we find that these new approaches are extremely powerful and great fun.



And it is also done

Experiments driving, for once... [Ben's talk]

- 2014/15: first jet image papers
- 2017: first (working) ML top tagger
- ML4Jets 2017: What architecture works best?
- ML4Jets 2018: Lots of architectures work [1902.09914]

$\Rightarrow$ Jet classification understood and done

What is new and cool and fun?

What about error bars?

Jet-by-jet uncertainties [Walter: Bayesians have more fun]

- $(60\pm??)\%$ top,
- probability for test event $p(c^*|C)$ [classifier output C, network ω]

$$p(c^*|C) = \int d\omega \; p(c^*|\omega, C) \; p(\omega|C) = \int d\omega \; p(c^*|\omega, C) \; q(\omega)$$

- loss: minimize KL-divergence + Bayes

$$\begin{aligned} \text{KL}[q(\omega), p(\omega|C)] &= \int d\omega \; q(\omega) \; \log \frac{q(\omega)}{p(\omega|C)} \\ &= \int d\omega \; q(\omega) \; \log \frac{q(\omega)p(C)}{p(C|\omega)p(\omega)} \\ &= \underbrace{\text{KL}[q(\omega), p(\omega)]}_{\text{L2-regularization}} + \underbrace{\log p(C) \int d\omega \; q(\omega)}_{\text{normalization of } q\text{, irrelevant}} - \underbrace{\int d\omega \; q(\omega) \log p(C|\omega)}_{\text{likelihood, maximized}} \end{aligned}$$

$$\Rightarrow L = \text{KL}[q(\omega), p(\omega)] - \int d\omega \; q(\omega) \log p(C|\omega)$$

What about error bars?

Jet-by-jet uncertainties [Walter: Bayesians have more fun]

– (60±??)% top,

– probability for test event $p(c^*|C)$ [classifier output C, network ω]

$$p(c^*|C) = \int d\omega \; p(c^*|\omega, C) \; p(\omega|C) = \int d\omega \; p(c^*|\omega, C) \; q(\omega)$$

– loss: minimize KL-divergence + Bayes

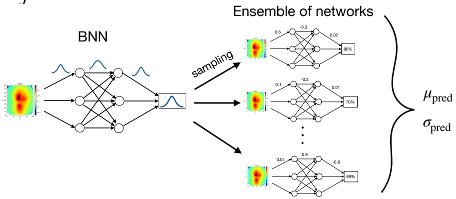
$$\begin{aligned} \text{KL}[q(\omega), p(\omega|C)] &= \int d\omega \; q(\omega) \; \log \frac{q(\omega)}{p(\omega|C)} \\ &= \int d\omega \; q(\omega) \; \log \frac{q(\omega)p(C)}{p(C|\omega)p(\omega)} \\ &= \underbrace{\text{KL}[q(\omega), p(\omega)]}_{\text{L2-regularization}} + \underbrace{\log p(C) \int d\omega \; q(\omega)}_{\text{normalization of } q\text{, irrelevant}} - \underbrace{\int d\omega \; q(\omega) \log p(C|\omega)}_{\text{likelihood, maximized}} \end{aligned}$$

$$\Rightarrow L = \text{KL}[q(\omega), p(\omega)] - \int d\omega \; q(\omega) \log p(C|\omega)$$

$\Rightarrow$ sample ω to extract $(\mu_\text{pred}, \sigma_\text{pred})$

check prior independence

check frequentist many-networks

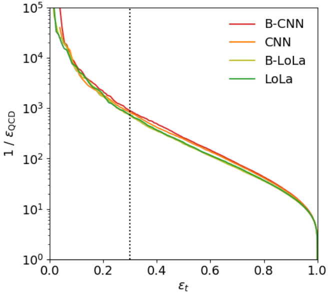


Statistics

Training statistics [Bollweg, Haussmann, Kasieczka, Luchmann, TP, Thompson]

– Bayesian version of DeepTop and LoLa

– similar performance as deterministic network
training time somewhat increased



Statistics

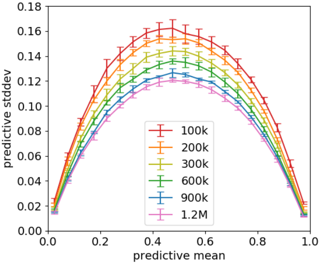
Training statistics [Bollweg, Haussmann, Kasieczka, Luchmann, TP, Thompson]

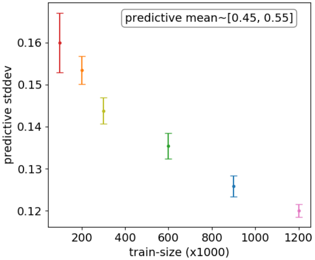
– Bayesian version of DeepTop and LoLa

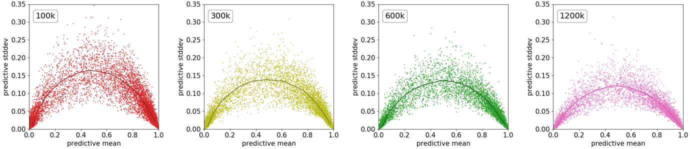
– similar performance as deterministic network
  training time somewhat increased

– correlation between μ_{pred} and σ_{pred} [toy network, 10k jets]

– increasing training statistics [parabola from closed interval output]







Statistics and systematics

Regression: measure $p_{T,t}$ [Kasieczka, Luchmann, Otterpohl, TP]

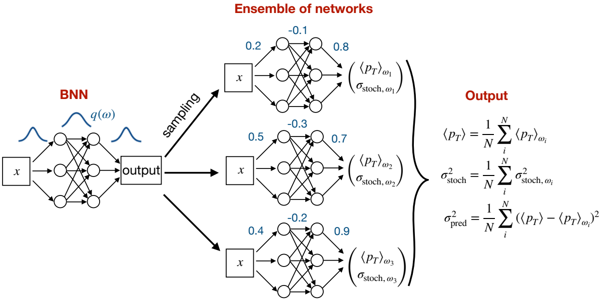
– effect of noisy and size-limited data separated

σ_{pred}: limited training sample

σ_{stoch}: statistical behavior of training data [Gaussian likelihood]

$$\log p(C|\omega) \to \log p(C|\mu, \sigma_{\text{stoch}}) = \frac{(C-\mu)^2}{2\sigma_{\text{stoch}}^2} + \frac{1}{2}\log \sigma_{\text{stoch}}^2 + \text{const}$$

$$\sigma_{\text{tot}}^2 = \sigma_{\text{pred}}^2 + \sigma_{\text{stoch}}^2 \quad \text{[all Gaussian]}$$



Statistics and systematics

Regression: measure $p_{T,t}$ [Kasieczka, Luchmann, Otterpohl, TP]

– effect of noisy and size-limited data separated

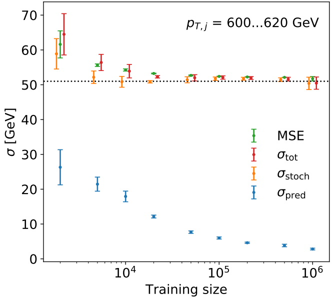
σ_{pred}: limited training sample

σ_{stoch}: statistical behavior of training data [Gaussian likelihood]

$$\log p(C|\omega) \to \log p(C|\mu, \sigma_{\text{stoch}}) = \frac{(C-\mu)^2}{2\sigma_{\text{stoch}}^2} + \frac{1}{2} \log \sigma_{\text{stoch}}^2 + \text{const}$$

$\sigma_{\text{tot}}^2 = \sigma_{\text{pred}}^2 + \sigma_{\text{stoch}}^2$ [all Gaussian]

– sample size dependence [statistics saturating]



Statistics and systematics

Regression: measure $p_{T,t}$ [Kasieczka, Luchmann, Otterpohl, TP]

– effect of noisy and size-limited data separated

σ_pred: limited training sample

σ_stoch: statistical behavior of training data [Gaussian likelihood]

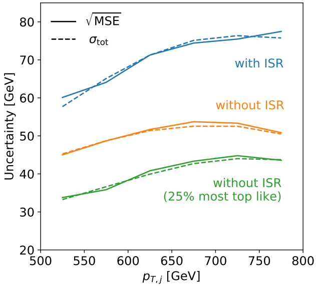
$$\log p(C|\omega) \to \log p(C|\mu, \sigma_\text{stoch}) = \frac{(C-\mu)^2}{2\sigma_\text{stoch}^2} + \frac{1}{2}\log\sigma_\text{stoch}^2 + \text{const}$$

$\sigma_\text{tot}^2 = \sigma_\text{pred}^2 + \sigma_\text{stoch}^2$ [all Gaussian]

– sample size dependence [statistics saturating]

– dependence on ISR and top-ness

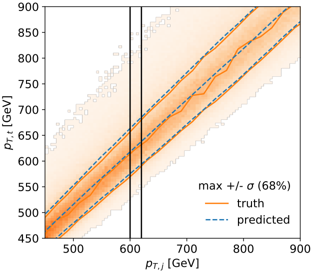
⇒ Reasonable error estimate

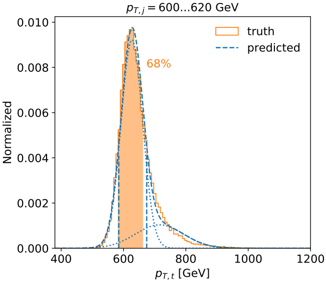


Jet calibration

Calibration means error propagation

- training on smeared data??
  better: training with smeared labels [p_T measured elsewhere, with error]
- Gaussian noise over $p_{T,t}$ label [e.g. 4%]
- distribution of extracted $p_{T,t}$
  correlation extending to error bars
  slice with expected non-Gaussian tail from QCD radiation



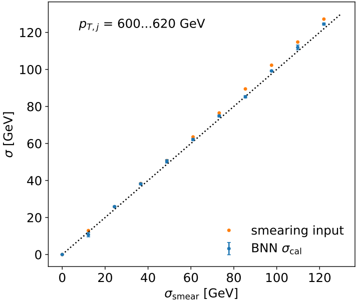


Jet calibration

Calibration means error propagation

- training on smeared data??
  better: training with smeared labels [p_T measured elsewhere, with error]
- Gaussian noise over $p_{T,t}$ label [e.g. 4%]
- distribution of extracted $p_{T,t}$
  correlation extending to error bars
  slice with expected non-Gaussian tail from QCD radiation
- trace label smearing to network output
  making sense of σ_{noise}

$\Rightarrow$ Works!



2– Learning from art

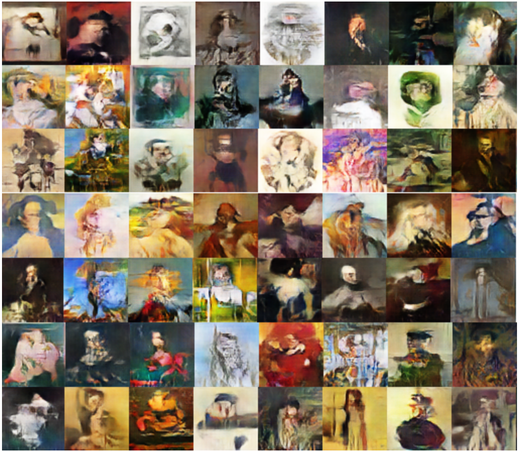
GANGogh [Bonafilia, Jones Danyluk (2017)]

– old news: NNs turning pictures into art of a certain epoch
  but can they create new pieces of art?
– train on 80,000 pictures [organized by style and genre]
– map noise vector to images
– generate flowers

2– Learning from art

GANGogh [Bonafilia, Jones Danyluk (2017)]

- old news: NNs turning pictures into art of a certain epoch
  but can they create new pieces of art?
- train on 80,000 pictures [organized by style and genre]
- map noise vector to images
- generate portraits



2– Learning from art

GANGogh [Bonafilia, Jones Danyluk (2017)]

– old news: NNs turning pictures into art of a certain epoch
  but can they create new pieces of art?

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

– map noise vector to images

Edmond de Belamy [Caselles-Dupre, Fautrel, Vernier]

– trained on 15,000 portraits

– sold for $432.500

$\Rightarrow$ all about marketing and sales

2– Learning from art

GANGogh [Bonafilia, Jones Danyluk (2017)]

– old news: NNs turning pictures into art of a certain epoch but can they create new pieces of art?

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

– map noise vector to images

GANGogh for jet images [de Oliveira, Paganini, Nachman]

– start with calorimeter images or jet images sparsity the technical challenge

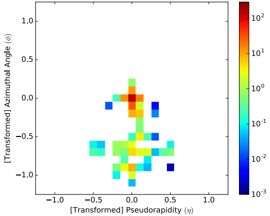
1- reproduce valid jet images from training data

2- organize them by QCD vs W-decay jets

– high-level observables to check

– not sold for cash

⇒ all about understanding



GANs at LHC

Phase space networks

- MC integration [Bendavit (2017)]
- NNVegas [Klimek (2018), not really generative network]

Existing GAN studies [Anja's talk]

- Jet Images [de Oliveira (2017), Carazza (2019)]
- Detector simulations [Paganini (2017), Musella (2018), Erdmann (2018), Ghosh (2018), Buhmann (2020)]
- Event generation [Otten(2019), Hashemi (2019), Di Sipio (2019), Butter (2019), Martinez (2019), Alanazi (2020)]
- Unfolding [Datta (2018), Bellagente (2019)]
- Templates for QCD factorization [Lin (2019)]
- EFT models [Erbin (2018)]
- Event subtraction [Butter (2019)]

Event generators

- neural importance sampling [Bothmann (2020)]
- i-flow in SHERPA [Gao (2020)]

GANs at LHC

Phase space networks

- MC integration [Bendavit (2017)]
- NNVegas [Klimek (2018), not really generative network]

Existing GAN studies [Anja's talk]

- Jet Images [de Oliveira (2017), Carazza (2019)]
- Detector simulations [Paganini (2017), Musella (2018), Erdmann (2018), Ghosh (2018), Buhmann (2020)]
- Event generation [Otten(2019), Hashemi (2019), Di Sipio (2019), Butter (2019), Martinez (2019), Alanazi (2020)]
- Unfolding [Datta (2018), Bellagente (2019)]
- Templates for QCD factorization [Lin (2019)]
- EFT models [Erbin (2018)]
- Event subtraction [Butter (2019)]

Event generators

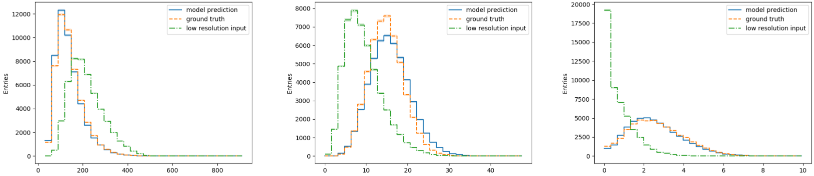
- neural importance sampling [Bothmann (2020)]
- i-flow in SHERPA [Gao (2020)]

What is new and cool and fun?

Super-resolution (preview)?

Getting inspired [Blecher, Butter, Keilbach, TP + Irvine]

- take high-resolution calorimeter images
  down-sample to 1/8th 1D resolution
  GAN inversion
- start from low-resolution calorimeter images
  GAN high-resolution images
- works because GANs learn structure [showers are QCD]
- energy of constituents no.1,10,30



⇒ GANs are (kind of) magic

What about MC-inversion?

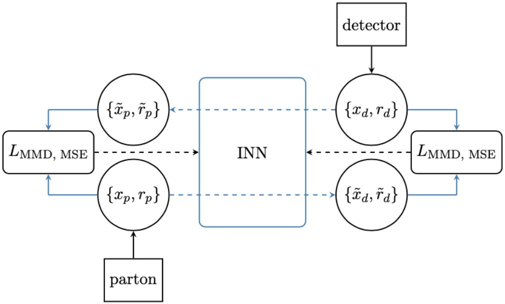
Unfolding as inversion [Bellagente, Butter, Kasieczka, TP, Rousselot, Winterhalder]

- network as bijective transformation — normalizing flow
  Jacobian tractable — normalizing flow
  evaluation in both directions — INN [Ardizzone, Kruse, Rother, Köthe]
- building block: coupling layer

$$x_d \sim g(x_p) \qquad \text{with} \qquad \frac{\partial g(x_p)}{\partial x_p} = \begin{pmatrix} \text{diag } e^{s_2(x_{p,2})} & \text{finite} \\ 0 & \text{diag } e^{s_1(x_{d,1})} \end{pmatrix}$$

- padding by yet more random numbers

$$\begin{pmatrix} x_p \\ r_p \end{pmatrix} \xleftrightarrow[\leftarrow\ \text{unfolding}:\bar{g}]{\textsc{Pythia},\textsc{Delphes}:g\rightarrow} \begin{pmatrix} x_d \\ r_d \end{pmatrix}$$



What about MC-inversion?

Unfolding as inversion [Bellagente, Butter, Kasieczka, TP, Rousselot, Winterhalder]

– network as bijective transformation — normalizing flow
  Jacobian tractable — normalizing flow
  evaluation in both directions — INN [Ardizzone, Kruse, Rother, Köthe]

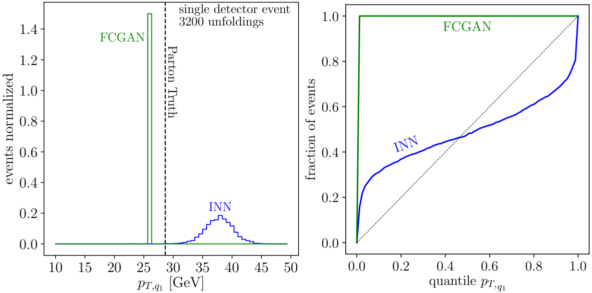
– building block: coupling layer

$$x_d \sim g(x_p) \qquad \text{with} \qquad \frac{\partial g(x_p)}{\partial x_p} = \begin{pmatrix} \text{diag } e^{s_2(x_{p,2})} & \text{finite} \\ 0 & \text{diag } e^{s_1(x_{d,1})} \end{pmatrix}$$

– padding by yet more random numbers

$$\begin{pmatrix} x_p \\ r_p \end{pmatrix} \xleftrightarrow[\leftarrow \text{ unfolding}:\bar{g}]{\text{PYTHIA,DELPHES}:g \rightarrow} \begin{pmatrix} x_d \\ r_d \end{pmatrix}$$

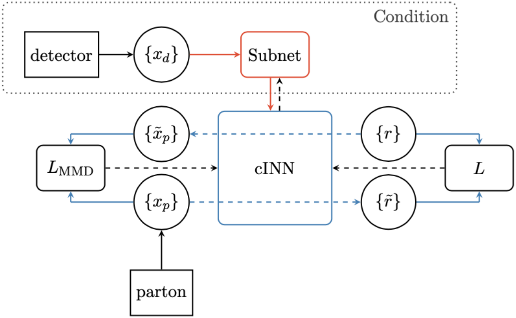
⇒ proper sampling



Conditional INN

Even more random sampling: conditional network

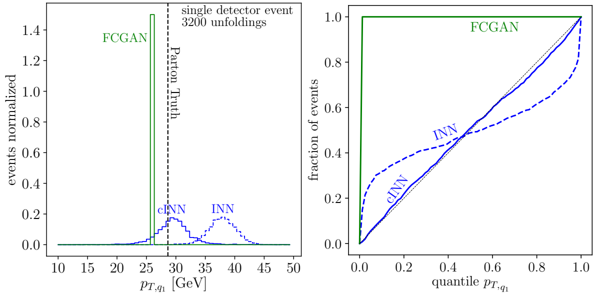
- same as Anja's FCGAN [Omnifold]
- parton-level events from random numbers



Conditional INN

Even more random sampling: conditional network

- same as Anja's FCGAN [Omnifold]
- parton-level events from random numbers
- calibration for statistical unfolding



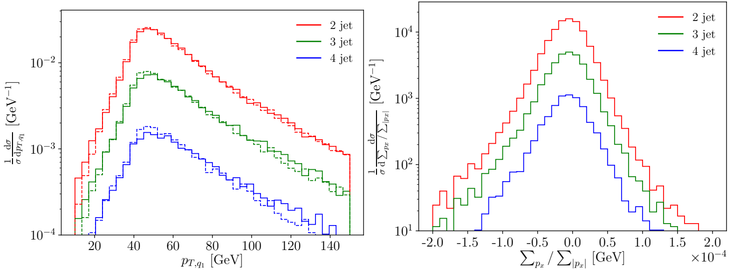
Conditional INN

Even more random sampling: conditional network

- same as Anja's FCGAN [Omnifold]
- parton-level events from random numbers
- calibration for statistical unfolding

Unfolding extra jets

- detector-level process $pp \to ZW$+jets [variable number of objects]
- parton-level hard process chosen $2 \to 2$ [whatever you want]
- ME vs PS jets decided by network [including momentum conservation]



$\Rightarrow$ proper statistical inversion!

Outlook

Machine learning a great tool box

LHC physics really is big data

jet classification was a starting point

generative networks exciting for theory

physics questions: errors, precision, control, theory insight

What is new and cool and fun?