MLhad: A Machine Learning based Simulation for Hadronization

HEP Seminar @JSI, Ljubljana

Based on [SciPost Phys. 14, 027 (2023)](), 2308.nnnnn, and 2308.XXXXX

Ahmed Youssef
Ph.D. Candidate, University of Cincinnati
youssead@ucmail.uc.edu

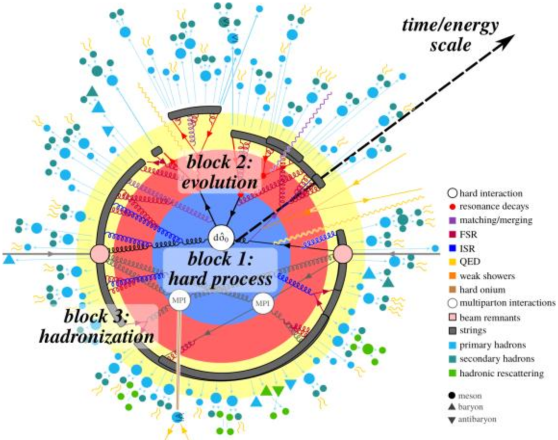
Aug 3rd, 2023

In collaboration with:

P. Ilten, T. Menzo, S. Mrenna, M. Szewc, M.K. Wilkinson, and J. Zupan

Motivation

Simulating Collision



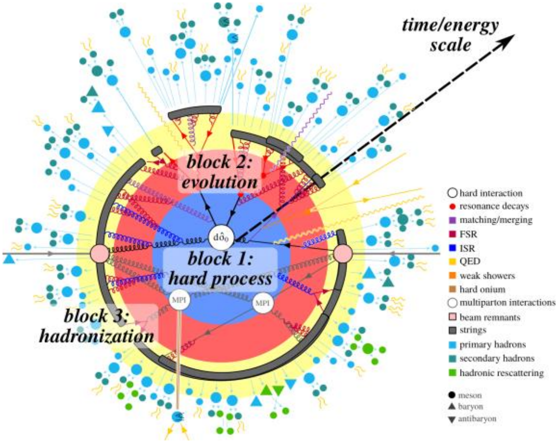
- **Hard process:** initial high-energy interaction
- **Evolution:** parton shower

perturbative

- **Hadronization**: combine quarks and gluons

non -perturbative

Simulating Collision



- **Hard process:**
  initial high-energy interaction
- **Evolution:**
  parton shower

perturbative

- **Hadronization**:
  combine quarks and gluons

non ... e

Use ML!

MLHAD

When is a hadronization model successful?

Motivation

When is a hadronization model successful?

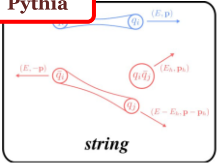
➡ **The performance is judged by their description of experimental measurements!**

Motivation

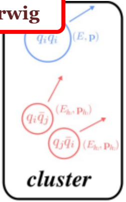
When is a hadronization model successful?

➡ The performance is judged by their description of experimental measurements!

Pythia



Herwig



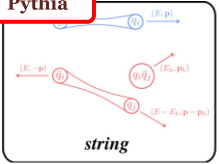
Phenomenological Models (String, Cluster) are currently state of art and are overall very successful, however:

- ➡ comparison of data from proton-proton and ion-ion collision with Pythia
  - ➡ discrepancies at the level of O(20%) to O(50%) N. Fischer and T. Sj¨ostrand, [JHEP 01, 140 (2017), 1610.09818.]
- ➡ recovering collective effects can be challenging, for instance, heavy baryon production at high event multiplicities Alice Collaboration, [arXiv: 1807.11321]
- ➡ no efficient estimation of Uncertainties

When is a hadronization model successful?

➜ The performance is judged by their description of experimental measurements!

Pythia

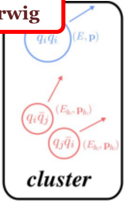


Phenomenological Models (String, Cluster) are currently state of art and are overall very successful, however:

➜ comparison of data from proton-proton and ion-ion collision with Pythia

➜ discrepancies at the level of O(20%) to O(50%) N. Fischer and T. Sj¨ostrand, JHEP 01, 140 (2017), 1610.09818.

Herwig



➜ recovering collective effects can be challenging, for instance, heavy baryon production at high event multiplicities Alice Collaboration, arXiv: 1807.11321

➜ no efficient estimation of Uncertainties

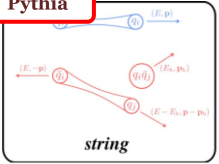
➜ **Both models have a discrepancy in describing experimental measurements!**

Motivation

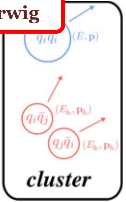
When is a hadronization model successful?

→ The performance is judged by their description of experimental measurements!

Pythia



Herwig



Phenomenological Models (String, Cluster) are currently state of art and are overall very successful, however:

→ comparison of data from proton-proton and ion-io... ...n with Pythia

→ discrepancies at the level of O(20%) to O(... ...and T. Sj¨ostrand, ...01, 140 (2017), 1610.09818.

→ recovering collective effects can... ...g, for instance, heavy baryon production at high event m... Alice Collaboration, arXiv: 1807.11321

→ no efficient estim... ...rtainties

We need an innovative approach!

→ **Both models have a discrepancy in describing experimental measurements!**

Big Picture

A series of progressive steps needs to be done before practically useful in Pythia simulations

- → ML architecture that mimics a simplified Lund string hadronization model
  - → Train on truth level Pythia output (not obs. In exp)
- → Develop a framework to propagate errors
- → Improved ML architecture with full hadron flavor selector
- → Train on mock data (i.e., just observable information)
- → Train on real data (i.e., just already measured information)
- → Replace Pythia string model

We are here

Partial results

Outline

Hadronization Models

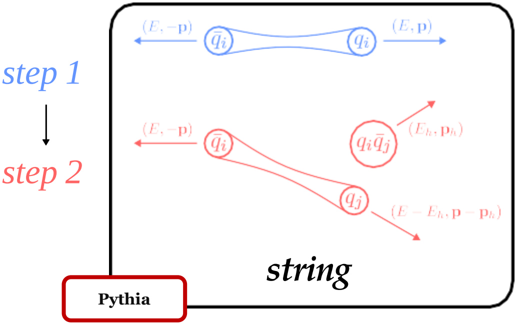
Which Generative Model should we use?

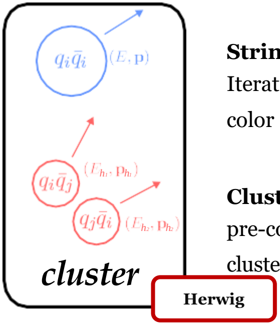
Uncertainties

Further Directions

Hadronization Models

Two primary hadronization models are used





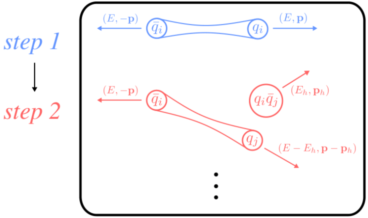
String model:
Iteratively split parton connected by QCD color strings with linear potential

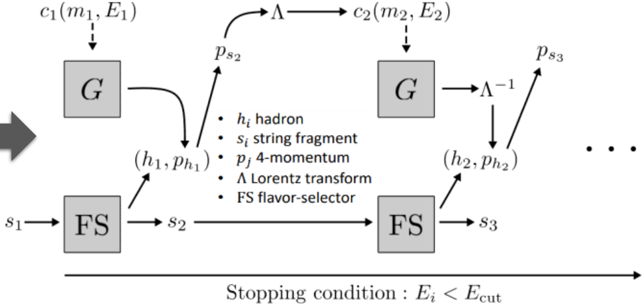
Cluster model:
pre-confine partons into proto-clusters, then split by two-body decays

MLhad: Ilten, Menzo,Youssef, JZ, 2203.04983, https://gitlab.com/uchep/mlhad

HadML: (Chan, Ghosh,) Ju, (Kania), Nachman, (Sangli,) Siodmok, 2203.12660, 2305.17169

MLHAD Pipeline





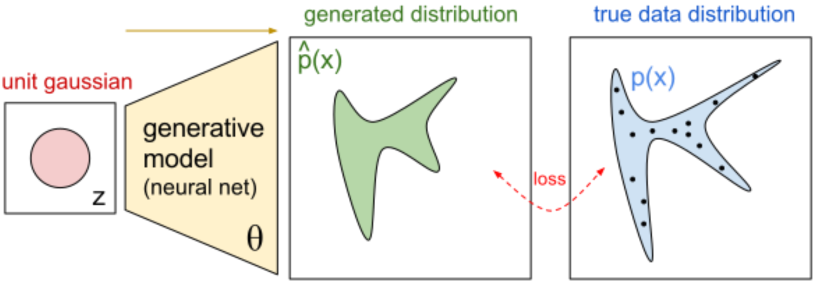
We need a generative model!

Sample hadron kinematics:
Train on $\{\boldsymbol{p_z}, \boldsymbol{p_T}\}$

Emission of different Mesons:
Condition on mass ($\boldsymbol{m}$) and energy ($\boldsymbol{E}$)

Generative Models

https://openai.com/research/generative-models



Source: generative models

$\Longrightarrow$ Task: Learn the probability distribution $p(x)$ of the data

Which generative model should we choose?

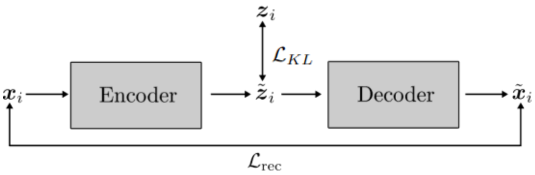
Is it able to learn **complex distributions**?

Do we have access to the **exact probability distribution**?

Generative Models

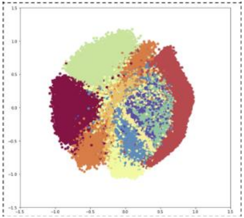
Variational Autoencoder (VAE)

Kingma et al, [arXiv:1312.6114]



Vanilla VAE

KL-divergence limits the latent space to a simple analytic distribution

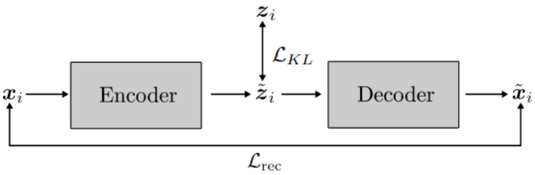


VAE latent space
[arXiv: 1804.01947]

Generative Models

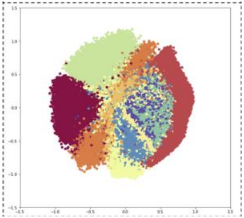
Variational Autoencoder (VAE)

Kingma et al, [arXiv:1312.6114]



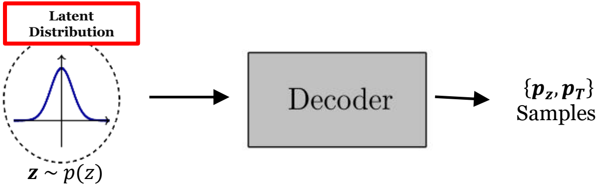
Vanilla VAE

KL-divergence limits the latent space to a simple analytic distribution



VAE latent space
[arXiv: 1804.01947]

Inference

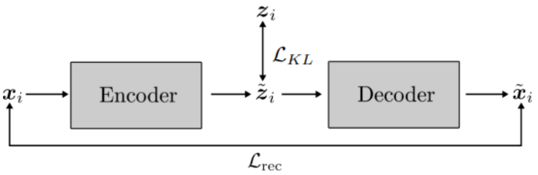


youssead@ucmail.uc.edu

Generative Models

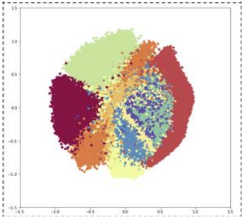
Variational Autoencoder (VAE)

Kingma et al, [arXiv:1312.6114]



Vanilla VAE

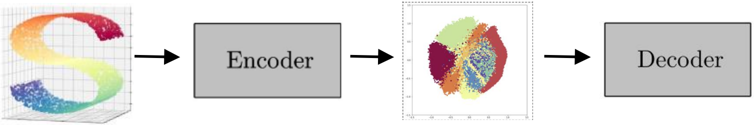
KL-divergence limits the latent space to a simple analytic distribution



VAE latent space
[arXiv: 1804.01947]

Complex input data

Simple latent space

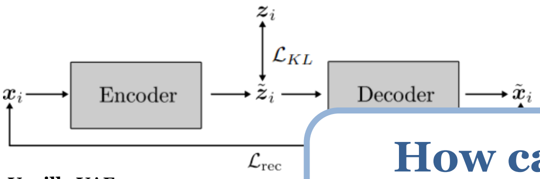


⇒ **Complex distribution are hard to learn!**

Generative Models

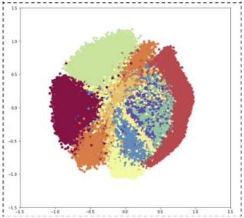
Variational Autoencoder (VAE)

Kingma et al, arXiv:1312.6114



Vanilla VAE

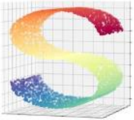
KL-divergence limits the latent space to a simple analytic distribution

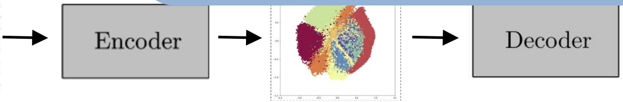


VAE latent space
arXiv: 1804.01947

How can we make VAEs learn more complex distribution?

Complex input data





$\Rightarrow$ Complex distribution are hard to learn!

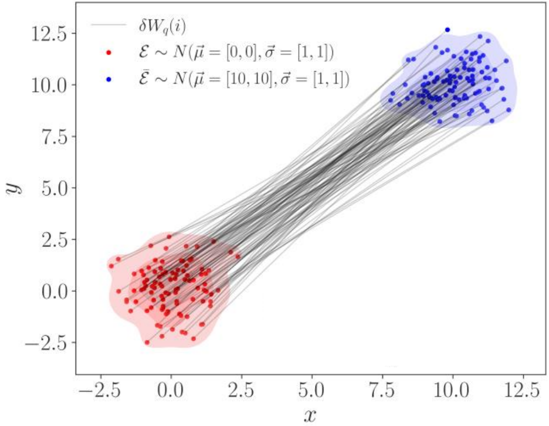
Use Sliced Wasserstein Distance as latent loss function!

youssead@ucmail.uc.edu

Use Sliced Wasserstein Distance as latent loss function!

Wasserstein distance (WD)

$$W_q(\mathcal{E},\bar{\mathcal{E}}) = \left[\min_{\{f_{ij}\geq 0\}} \sum_{i=1}^{N}\sum_{j=1}^{\bar{N}} f_{ij}\left(\hat{d}_{ij}\right)^q\right]^{1/q}$$



Generative Models

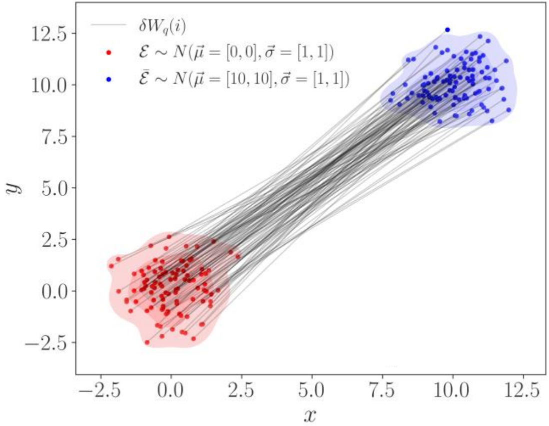
Use Sliced Wasserstein Distance as latent loss function!

Wasserstein distance (WD)

$$W_q(\mathcal{E},\bar{\mathcal{E}}) = \left[\min_{\{f_{ij}\geq 0\}} \sum_{i=1}^{N}\sum_{j=1}^{\bar{N}} f_{ij}\left(\hat{d}_{ij}\right)^q\right]^{1/q}$$

Sliced Wasserstein distance

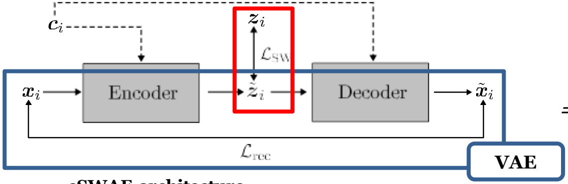
- Projects high dimensional data into one dimensional "slices"
- WD in 1D has a closed form solution
  - Sorted Difference of the two samples



Generative Models

[SciPost Phys. 14, 027 (2023)](#) **Conditional Sliced Wasserstein (SW) Autoencoder (cSWAE)**

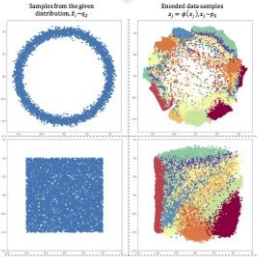
Restricted to Pion emissions



cSWAE architecture
(Architecture used in SciPost Phys. 14, 027 (2023))

SW distance enables learning any sampleable latent distribution

$\Rightarrow$ **Can learn complex distributions!**

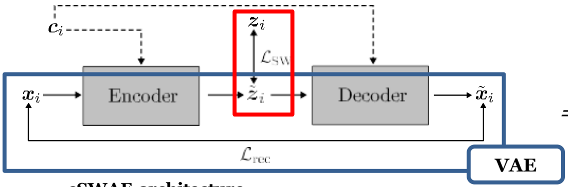


SWAE latent space
(arXiv: 1804.01947)

Generative Models

[SciPost Phys. 14, 027 (2023)](#) **Conditional Sliced Wasserstein (SW) Autoencoder (cSWAE)**

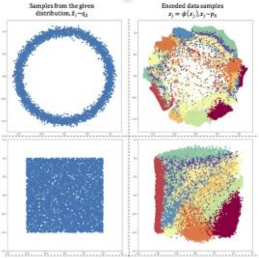
Restricted to Pion emissions



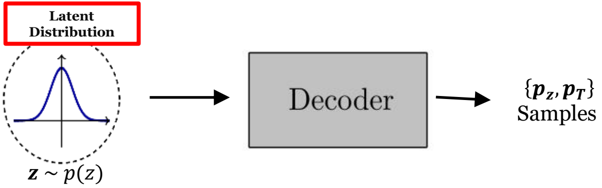
cSWAE architecture
(Architecture used in [SciPost Phys. 14, 027 (2023)](#))

SW distance enables learning any sampleable latent distribution

$\Rightarrow$ **Can learn complex distributions!**



SWAE latent space
(arXiv: 1804.01947)



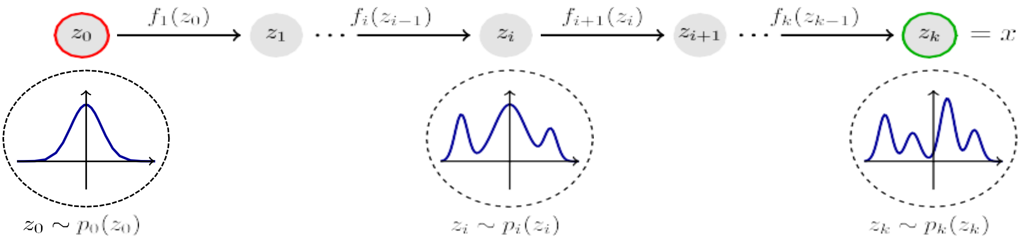
Decoder "just" generates samples

$\Rightarrow$ **No access to the probability distribution**

University of CINCINNATI

Normalizing Flows

MLHAD



Zo- random vector sampled from a Gaussian $p_0(z_0)$

Fi – invertible NN that transforms $p_0(z_0)$ to $p_i(z_i)$ by change of variables

Complex target distribution $p_k(z_k)$ is learned

$\Rightarrow$ **Can learn complex distributions!**

Exact probability distribution is obtained by change of variables

$$p_k(z_k) = p_0(z_0) \prod_{i=1}^{K} \left|\det\left(\frac{\partial f_i(z_{i-1})}{\partial z_{i-1}}\right)\right|^{-1}$$

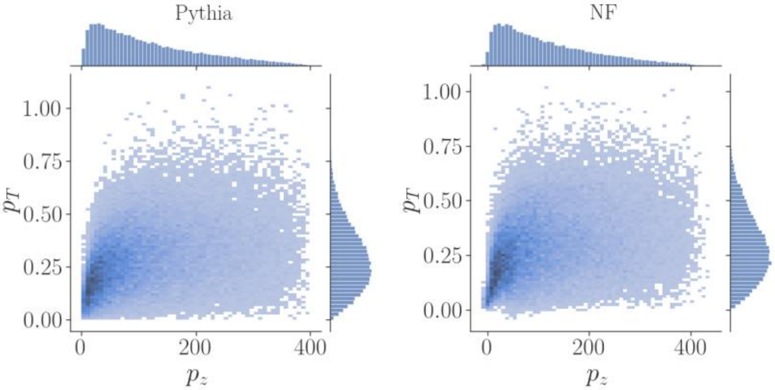
$\Rightarrow$ **Access to the exact probability distribution**

Removed pion emission restriction

hHps://github.com/janosh/awesome-normalizing-flows

Back to Physics

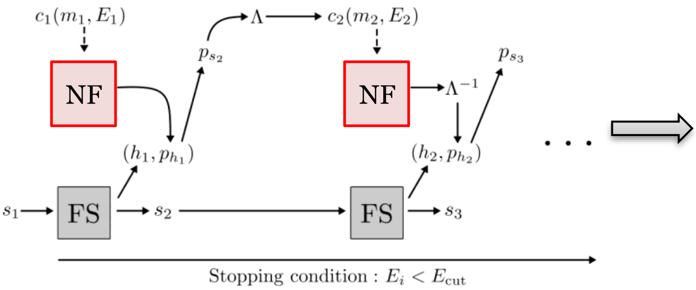
*Preliminary

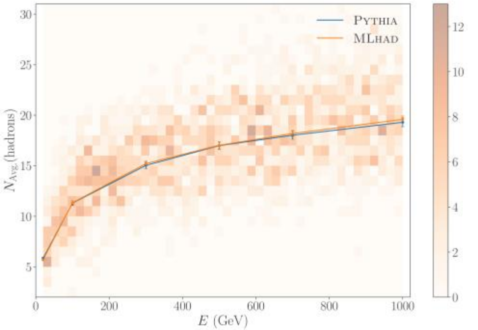


NFs, conditioned on different masses and energies, learn the correlation between p_z and p_T

Back to Physics

Implement NF in the fragmentation chain to obtain physical observables





$\Rightarrow$ ***Multiplicity obtained by MLHad agrees with Pythia!***

Uncertainty Quantification

Uncertainty estimation is crucial for event generator predictions!

youssead@ucmail.uc.edu

Uncertainty Quantification

Uncertainty estimation is crucial for event generator predictions!

- **Hard matrix element**
- **Parton shower**

Efficient solutions exist!

perturbative calculations depend on choices of scale, values of gauge and other couplings, particle masses, and nonperturbative inputs

Giele et al, [Phys. Rev. D84, 054003 (2011)]

S. Mrenna and P. Skands, [Phys. Rev. D94(7), 074005 (2016)]

Uncertainty Quantification

Uncertainty estimation is crucial for event generator predictions!

- **Hard matrix element**
- **Parton shower**

Efficient solutions exist!

perturbative calculations depend on choices of scale, values of gauge and other couplings, particle masses, and nonperturbative inputs

Giele et al, [Phys. Rev. D84, 054003 (2011)]
S. Mrenna and P. Skands, [Phys. Rev. D94(7), 074005 (2016)]

- **Hadronization**

Efficient solution has remained elusive!

Standard procedure: perform repeated simulations with different sets of values for the model parameters

→ **Computationally very expensive!**

youssead@ucmail.uc.edu

University of CINCINNATI

Uncertainty Quantification

MLHAD

Uncertainty estimation is crucial for event generator predictions!

- **Hard matrix element**
- **Parton shower**

Efficient solutions exist!

perturbative calculations depend on choices of scale, values of gauge and other couplings, particle masses, and nonperturbative inputs

Giele et al, [Phys. Rev. D84, 054003 (2011)]

S. Mrenna and P. Skands, [Phys. Rev. D94(7), 074005 (2016)]

- **Hadronization**

Efficient solution has rem ... ve!

Standard procedure: ... ated simulations with different s ... or the model parameters

→ ... utationally very expensive!

Need a more efficient way!

youssead@ucmail.uc.edu

Reweighting Hadronized Pythia Events

Small Detour:
No ML, only Had

Ilten et al, 2308.nnnnn

Reweighting Monte Carlo Predictions and Automated Fragmentation Variations in Pythia 8

Phil Ilten[1†], Tony Menzo[1⋆], Stephen Mrenna[2¶], Manuel Szewc[1ℸ], Michael K. Wilkinson[1*], Ahmed Youssef[1‡], and Jure Zupan[1§]

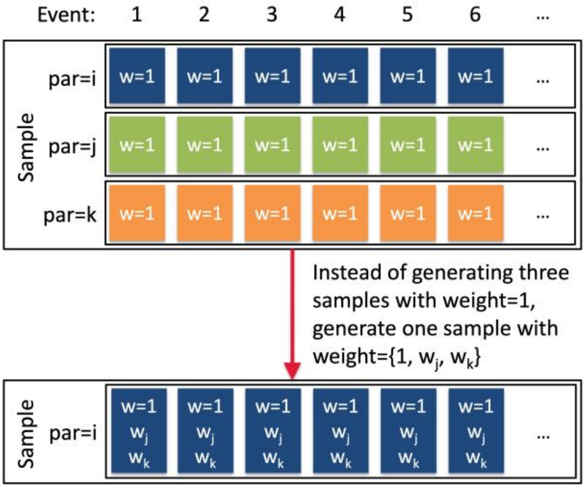
[1] Department of Physics, University of Cincinnati, Cincinnati, Ohio 45221,USA
[2] Scientific Computing Division, Fermilab, Batavia, Illinois, USA
†philten@cern.ch, ⋆menzoad@mail.uc.edu, ¶mrenna@fnal.gov, ℸszewcml@ucmail.uc.edu,
*michael.wilkinson@uc.edu, ‡youssead@ucmail.uc.edu, §zupanje@ucmail.uc.edu,

Abstract

This work reports on a method for uncertainty estimation in simulated collider-event predictions. The method is based on a Monte Carlo-veto algorithm, and extends the previous work on uncertainty estimates in parton showers by adding the uncertainty estimates for the Lund string-fragmentation model. The method is advantageous from the perspective of simulation costs: a single ensemble of generated events can be reinterpreted as though it was obtained using a different set of input parameters, where each event now gets accompanied with an appropriate weight. This allows for a robust exploration of the uncertainties arising from the choice of input model parameters. Such explorations are important when determining the sensitivities of precision physics measurements.

Reweighting Hadronized Pythia Events

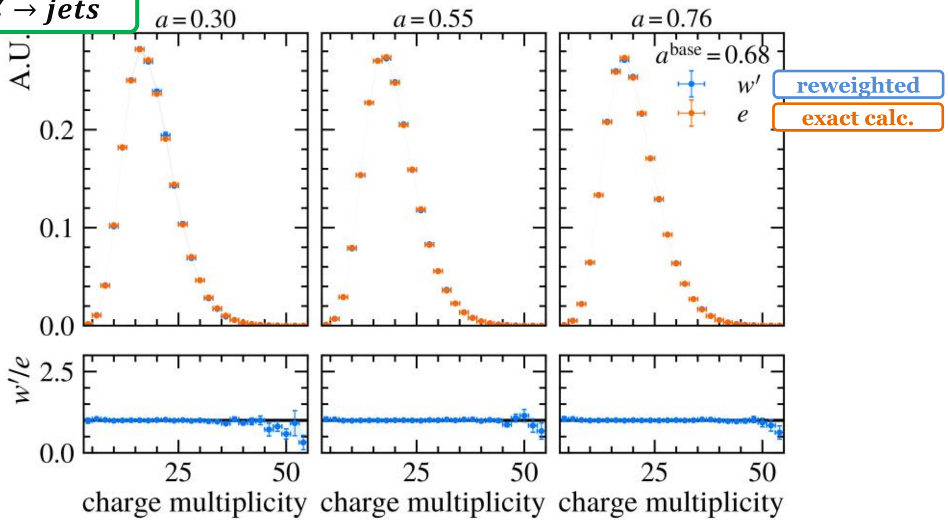
- **Event generation is time consuming**
  - **We want to reweight events without regenerating**
- **Use a modified veto algorithm**
  - **New event weights for different hadronization param are book kept**



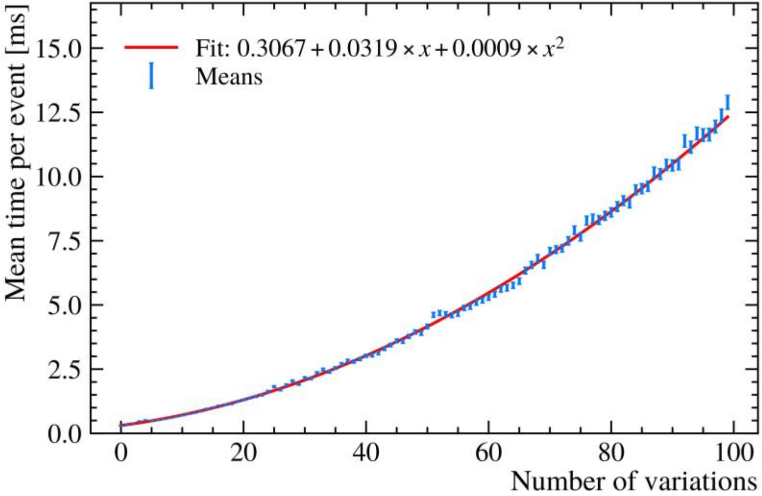
- **We calculate event weights for different hadronization options in a single event generation!**

Reweighting Hadronized Pythia Events

$e^+e^- \to Z \to jets$



Reweighting Hadronized Pythia Events



- Generate 100 samples with different variations of aLund
- Each sample has 1000 events
- For 10 alternative values we have a speed up by a factor ~4

Back to ML

Correlated Uncertainties

Statistical and Training Uncertainties

youssead@ucmail.uc.edu

Recall:

$$p_k(z_k) = p_0(z_0)\prod_{i=1}^{K}\left|\det\left(\frac{\partial f_i(z_{i-1})}{\partial z_{i-1}}\right)\right|^{-1}$$

→ **Can learn complex distributions**

→ **Access to the exact probability distribution**

Correlated Uncertainties

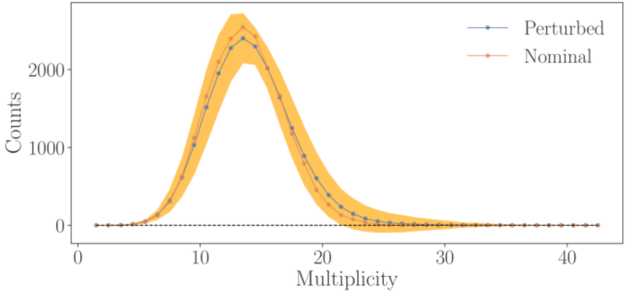
Recall:

$$p_k(z_k) = p_0(z_0)\prod_{i=1}^{K}\left|\det\left(\frac{\partial f_i(z_{i-1})}{\partial z_{i-1}}\right)\right|^{-1}$$

→ **Can learn complex distributions**

→ **Access to the exact probability distribution**

Mimic correlated uncertainty:



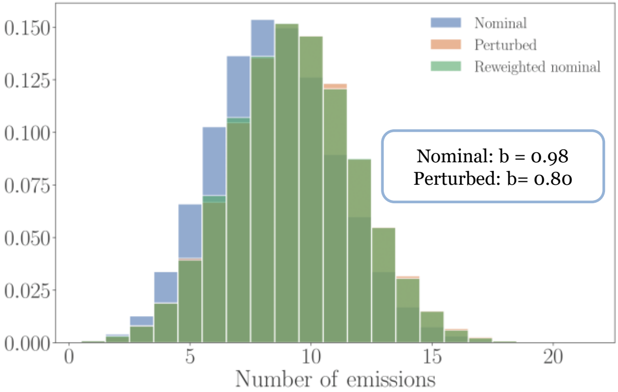
Generate multiple datasets with varied Pythia parameters with their standard deviation as error band

We can reweight between error bands by reweighting:

$$w = \prod_i \frac{p^{(i)}_{nom}(z)}{p^{(i)}_{pert}(z)}$$

Reweighting with NFs

*Preliminary



b is a free parameter in the Lund function used in Pythia: StringZ:bLund

Train nominal NF
→ Get likelihood

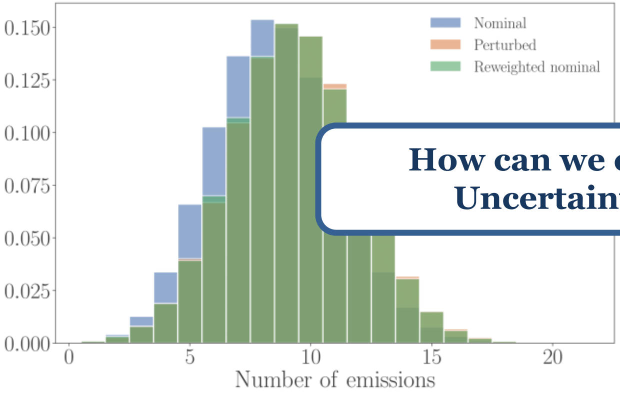
Train perturbed NF
→ Get likelihood

→ Reweight nominal output using ratio of likelihoods:

$$w = \prod_i \frac{p_{nom}^{(i)}(z)}{p_{pert}^{(i)}(z)}$$

Reweighting with NFs

*Preliminary



b is a free parameter in the Lund function used in Pythia: StringZ:bLund

NF
od

Train perturbed NF → Get likelihood

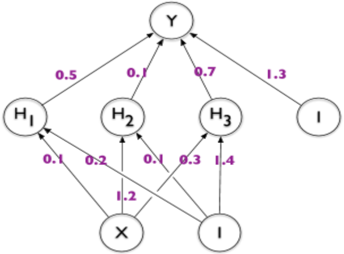
How can we capture Uncertainties?

→ Reweight nominal output using ratio of likelihoods:

$$w = \prod_i \frac{p^{(i)}_{nom}(z)}{p^{(i)}_{pert}(z)}$$

Statistical (and Training) Uncertainties

„Classical“ Neural Networks



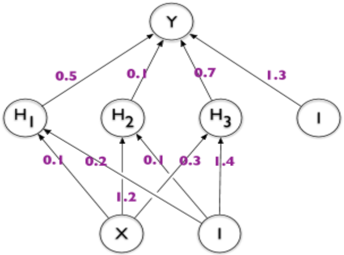
Weights have a fixed value
→ Weight values are updated in each epoch

(Image source: The very Basics of Bayesian Neural Networks)

youssead@ucmail.uc.edu

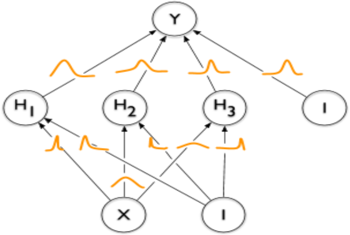
Statistical (and Training) Uncertainties

„Classical“ Neural Networks



Weights have a fixed value
→ Weight values are updated in each epoch

Bayesian Neural Networks (BNN)



Weights are sampled from a distribution
→ Distribution parameter are updated in each epoch

→ **BNN are easy to implement: Add additional loss function for weight distribution**

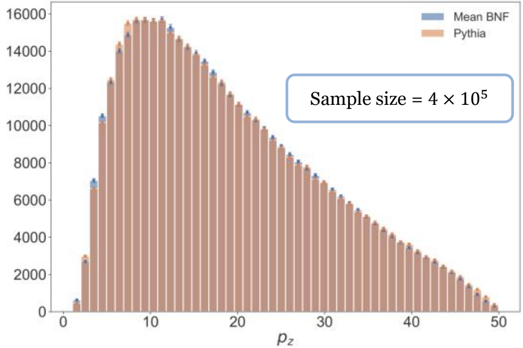
→ **Capture statistical and training uncertainties**

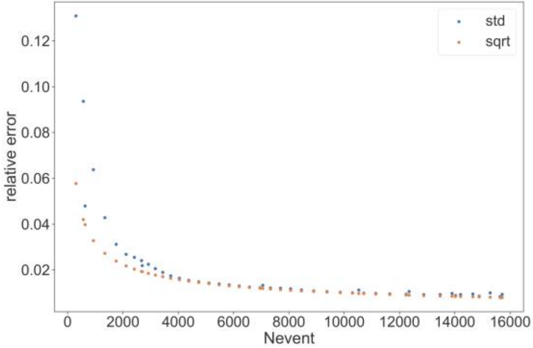
(Image source: The very Basics of Bayesian Neural Networks)

youssead@ucmail.uc.edu

Bayesian NF Results

*Preliminary





Pythia Sample:
One sample with errors corresponding to $\sqrt{N_{bin}}$

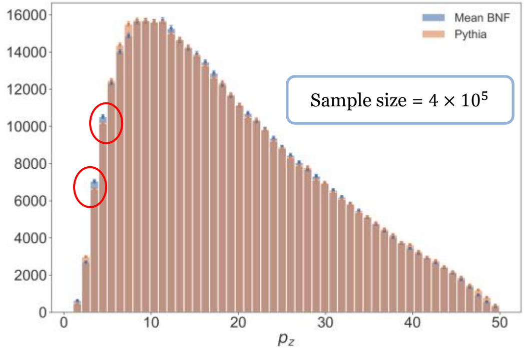
Mean BNF:
5×10^5 samples with errors corresponding to the standard deviation

BNF capture the statistical and training uncertainties

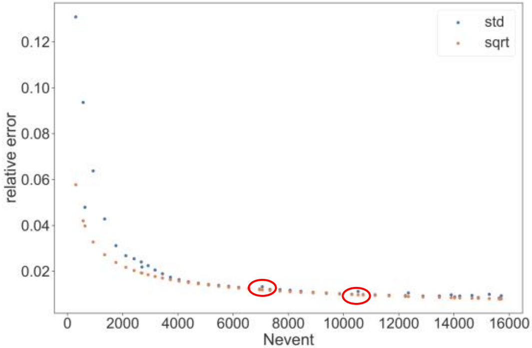
youssead@ucmail.uc.edu

Bayesian NF Results

*Preliminary



Sample size = 4×10^5



Pythia Sample:
One sample with errors corresponding to $\sqrt{N_{bin}}$

Mean BNF:
5×10^5 samples with errors corresponding to the standard deviation

BNF capture the statistical and training uncertainties

MLhadPipeline

We developed a pipeline for Hadronization based on the Lund model

Observables

Experiment

Validate

Not Observed!

Model

ML-Based

Sample

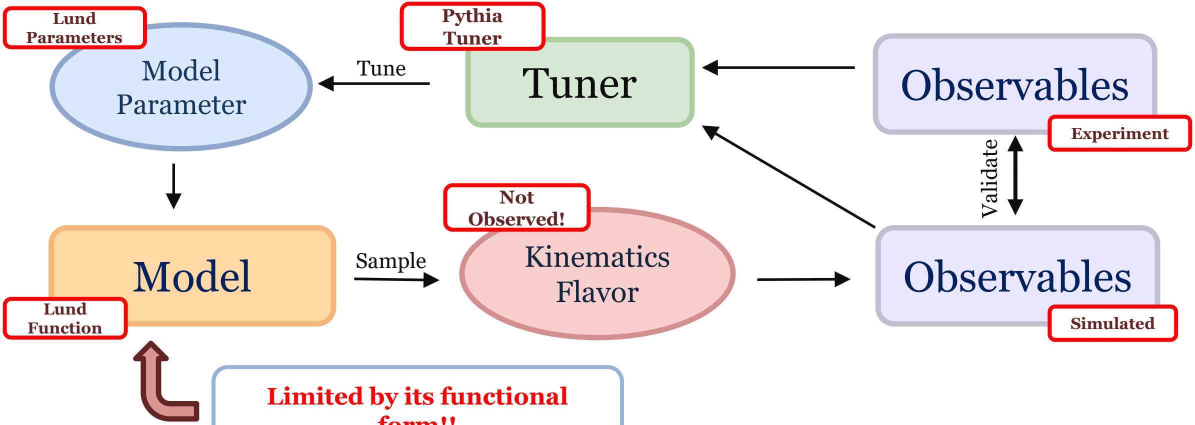
Kinematics
Flavor

Observables

Simulated

First Step:
Train on **"pseudo" data**
from Pythia

Pythia Pipeline

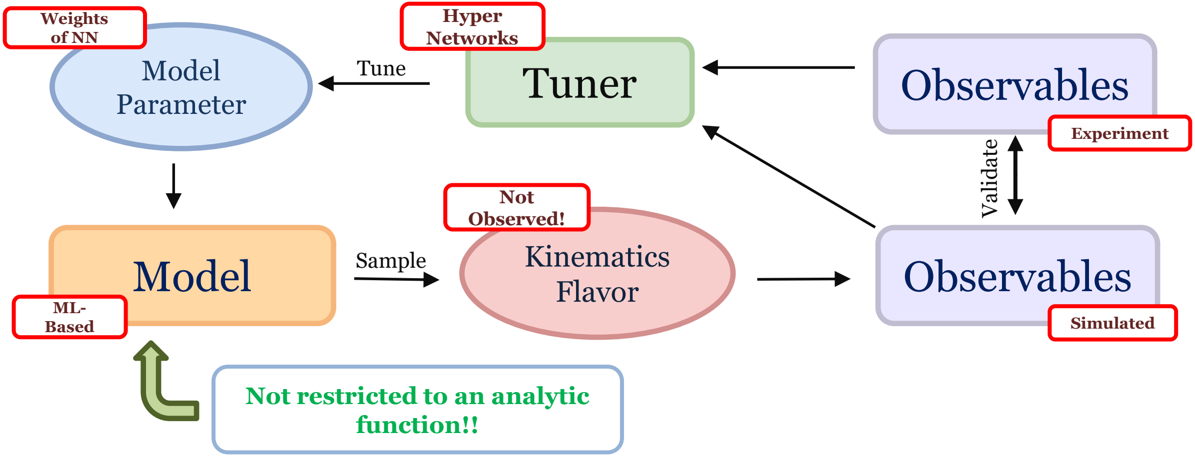


Limited by its functional form!!

Further Directions

MLhad Pipeline

*Preliminary



Further Directions

- **Propagation of errors**
  - **ML architecture with Bayesian Normalizing Flows (presented in part)**

Ilten et al, 2308.xxxxx

- **Train on observables only**
  - **Two part reweighter (not part of the talk)**
  - **Train on global observables with HN (results not shown in this talk)**

- **To train on experimental data**
  - **Want fast evaluation of parameter dependency**
  - **Use reweighting method**
  - **First implementation in Pythia for Lund string model (to be released soon in Pythia)**

Ilten et al, 2308.nnnnn

Big Picture

A series of progressive steps needs to be done before practically useful in Pythia simulations

- → ML architecture that mimicks a simplified Lund string hadronization model
  - → Train on truth level Pythia output (not obs. In exp)
- → Develop a framework to propagate errors

We are here

- → Improved ML architecture with full hadron flavor selector
- → Train on mock data (i.e., just observable information)

Partial results (shown by part)

- → Train on real data (i.e., just already measured information)
- → Replace Pythia string model

Conclusion and Outlook

- First MLHAD pipeline based on cSWAE was published in [SciPost Phys. 14, 027 (2023)]
- NFs overcome the limitations of cSWAE - can emit in principle any meson and have access to pdf
- NFs allow us to reweight events and capture uncertainties

Work in progress

- Finalize normalizing flows architecture (include model uncertainty)
- PYTHIA reweighting (Release as part of Pythia)
- Flavor Selector
- Performing training on physically accessible observables to train MLHAD on **experimental data**

Back up

youssead@ucmail.uc.edu

Training Results cNF

*Preliminary