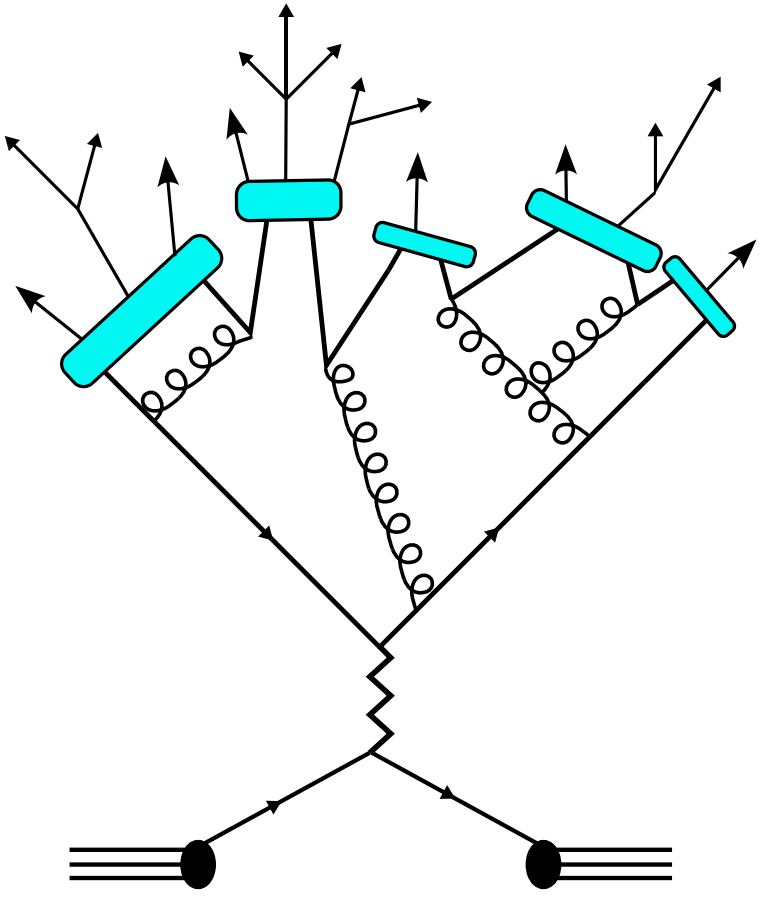
Data-driven hadronization models



Rutgers NHETC Seminar
November 19th, 2024

Tony Menzo
PhD candidate, University of Cincinnati

In collaboration with:

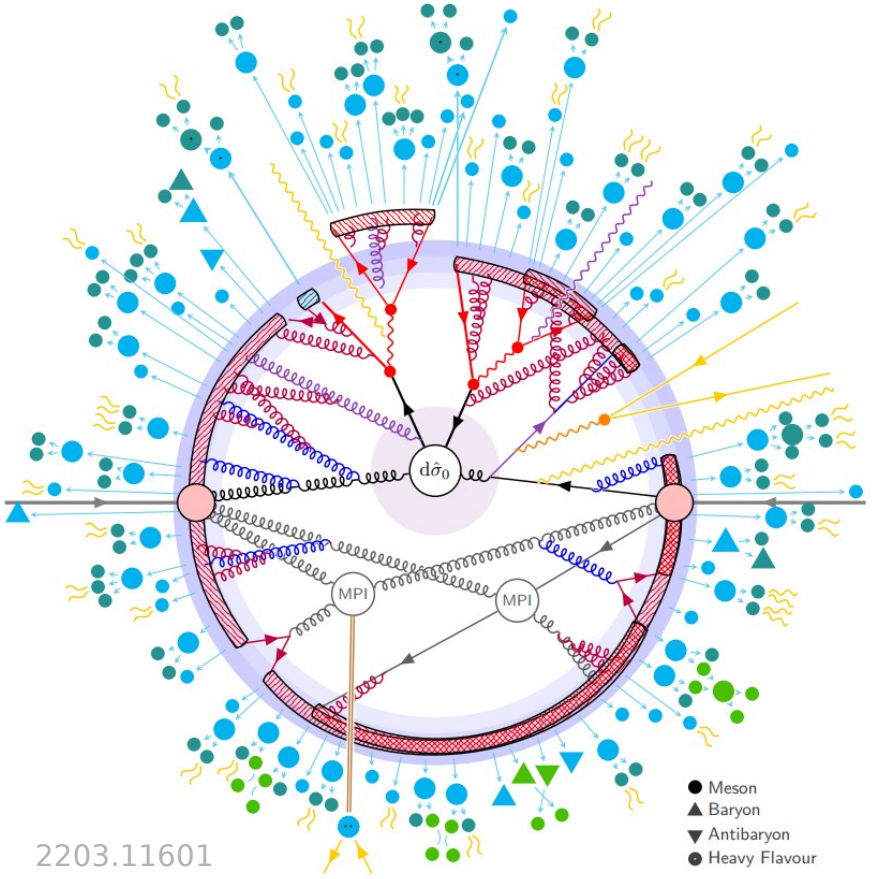
MLHAD

Christian Bierlich, Phil Ilten, Stephen Mrenna, Manuel Szewc,
Michael Wilkinson, Ahmed Youssef, Jure Zupan,

Nick Heller, Ben Nachman, Andrzej Siodmok

Based upon work in 2203.04983, 2308.13459, 2311.09296, 2410.06342, 2411.02194

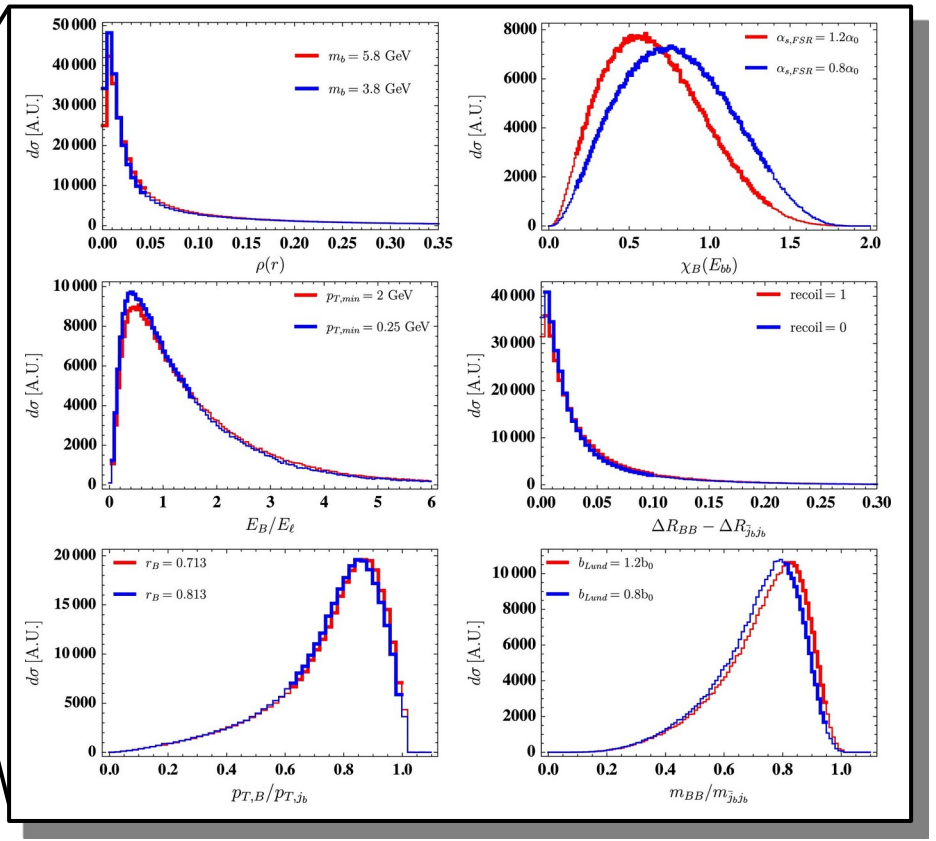
Event generators



Motivation

Precision (exclusive) measurements dependent on hadronization!

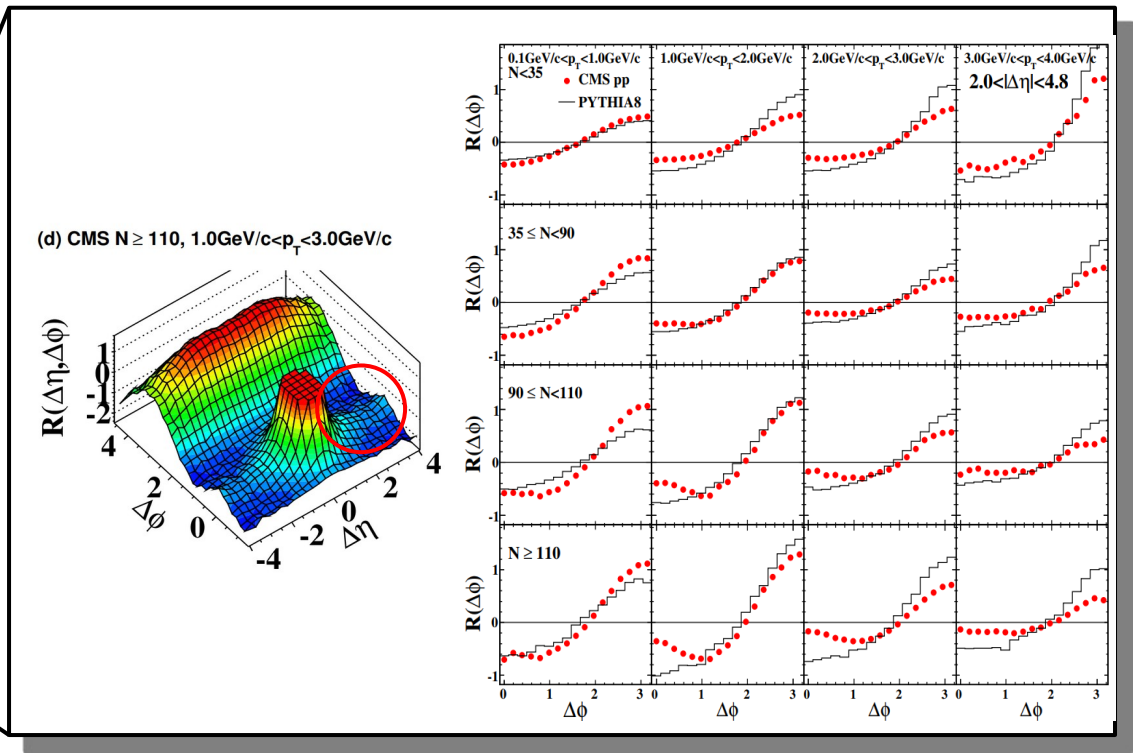
- Uncertainty reduction
  - Top quark mass measurement (r_b)
  - e^+e^- determination of α_s
- Mis-modeling
  - High-multiplicity events
  - Tuning discrepancies



Motivation

Precision (exclusive) measurements dependent on hadronization!

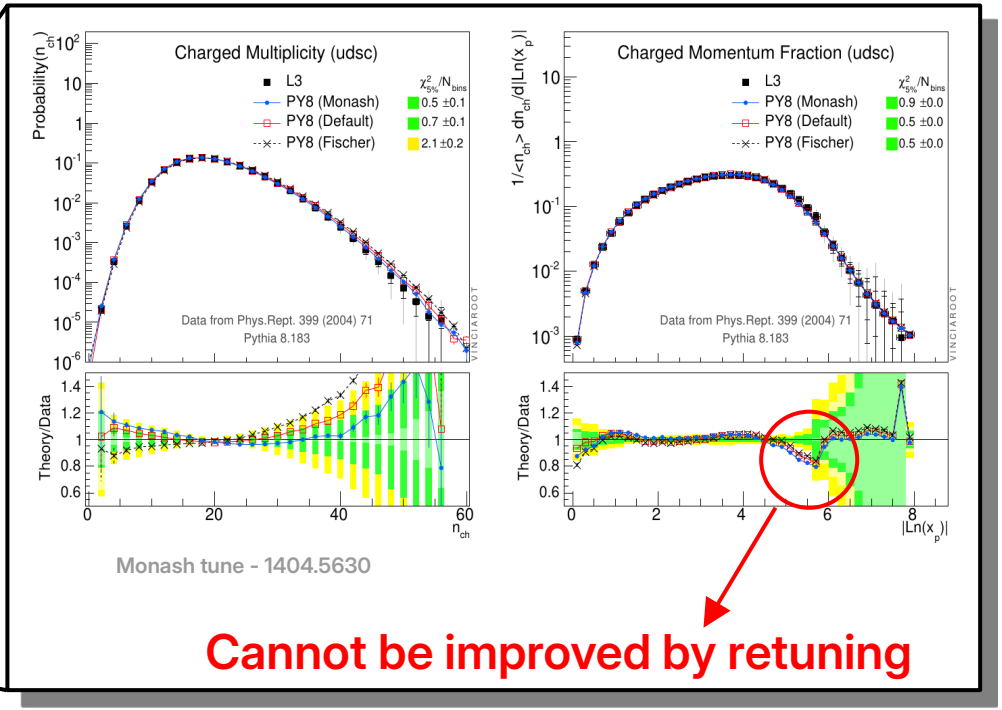
- Uncertainty reduction
  - Top quark mass measurement (r_b)
  - e^+e^- determination of α_s
- Mis-modeling
  - High-multiplicity events
  - Tuning discrepancies



Motivation

Precision (exclusive) measurements dependent on hadronization!

- Uncertainty reduction
  - Top quark mass measurement (r_b)
  - e^+e^- determination of α_s
- Mis-modeling
  - High-multiplicity events
  - Tuning discrepancies



Three roads to improvement

- Improve model
  - MPIs, rope hadronization, transverse mass supression, flavor asymmetries, hadronic rescattering, multiscale models (string → hydrodynamical), flavor selector, etc.

    Hard to come up with mathematically precise model without established calculational techniques
- Tend towards model independence
  - Sample directly from global distributions

    Non-universal and extremely difficult to convert into **representative** particle flow data (uninterpretable)
- Improve observables
  - Come up with better, hadronization sensitive (IR unsafe), observables

    Non-trivial to come up with new observables given current data resolution

Three* roads to improvement

- Improve model
  - MPIs, rope hadronization, transverse mass supression, flavor asymmetries, hadronic rescattering, multiscale models (string → hydrodynamical), flavor selector, etc.

    Hard to come up with mathematically precise model without established calculational techniques
- Tend towards model independence
  - Sample directly from global distributions

    Non-universal and extremely difficult to convert into **representative** particle flow data (uninterpretable)
- Improve observables
  - Come up with better, hadronization sensitive (IR unsafe), observables

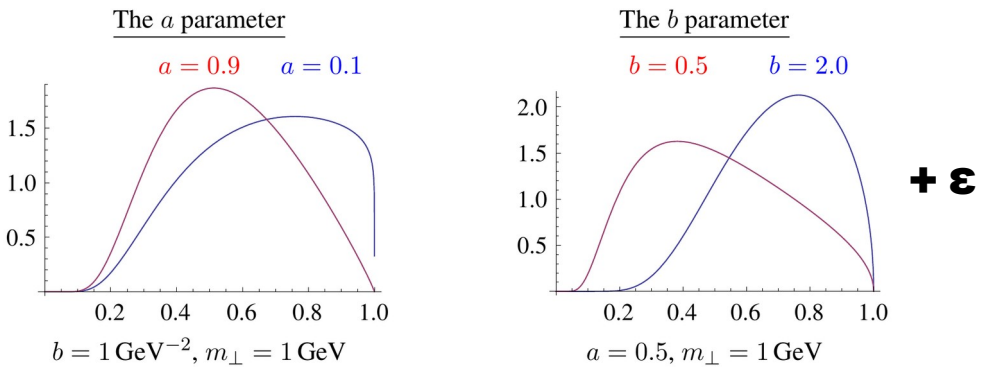
    Non-trivial to come up with new observables given current data resolution

*or some combination of all three

Hybrid data-driven approach

The phenomenological models of hadronization already give an acceptable description of a large amount of data.

Hybrid approach: Keep the underlying paradigm e.g. strings but modify the microscopic kinematics to accommodate the discrepant global experimental observables.

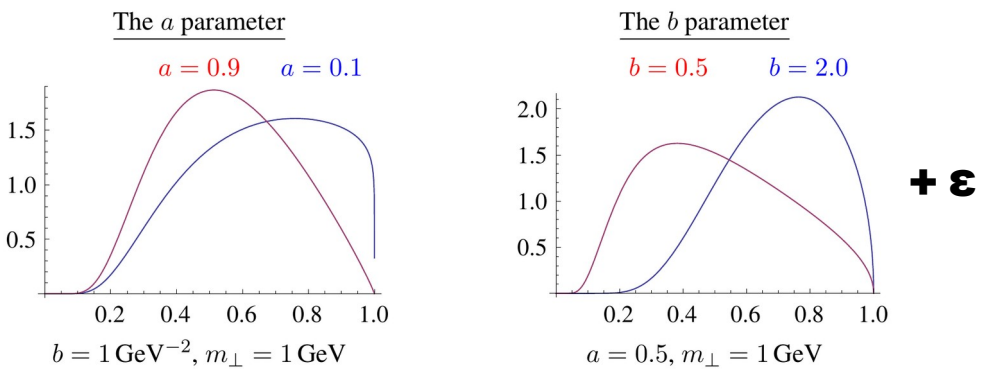


+ ε

Hybrid data-driven approach

The phenomenological models of hadronization already give an acceptable description of a large amount of data.

Hybrid approach: Keep the underlying paradigm e.g. strings but modify the microscopic kinematics to accommodate the discrepant global experimental observables.



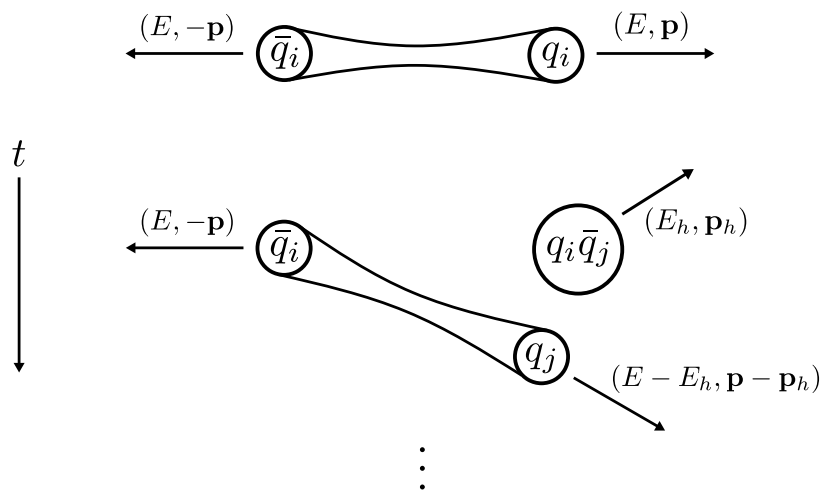
+ ε

Machine learning offers a nice framework to tackle this problem

Phenomenological models

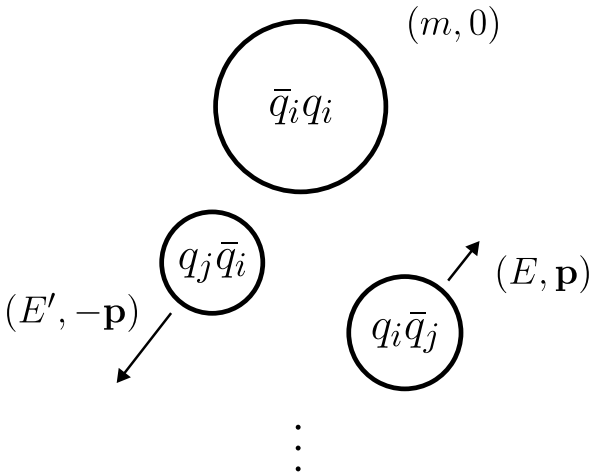
Lund string model

(used in Pythia)



Cluster model

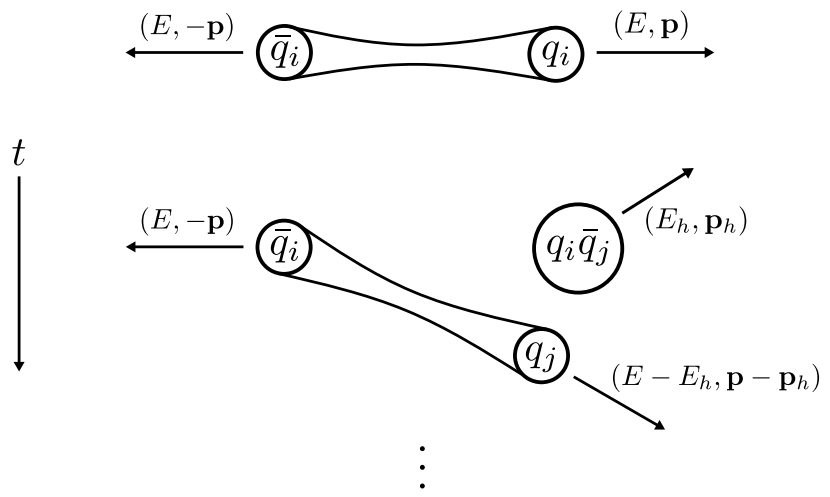
(used in Herwig)



Phenomenological models

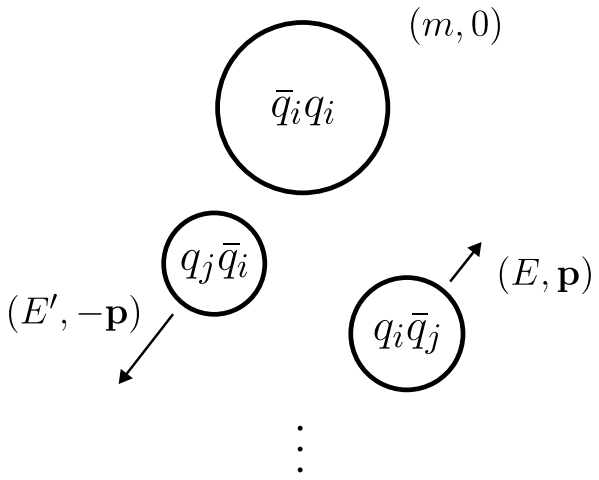
Lund string model

(used in Pythia)



Cluster model

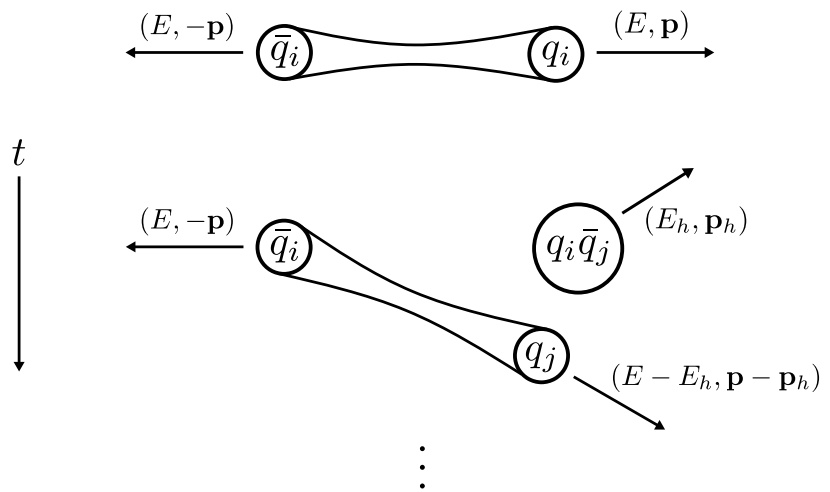
(used in Herwig)



Phenomenological models

Lund string model

(used in Pythia)



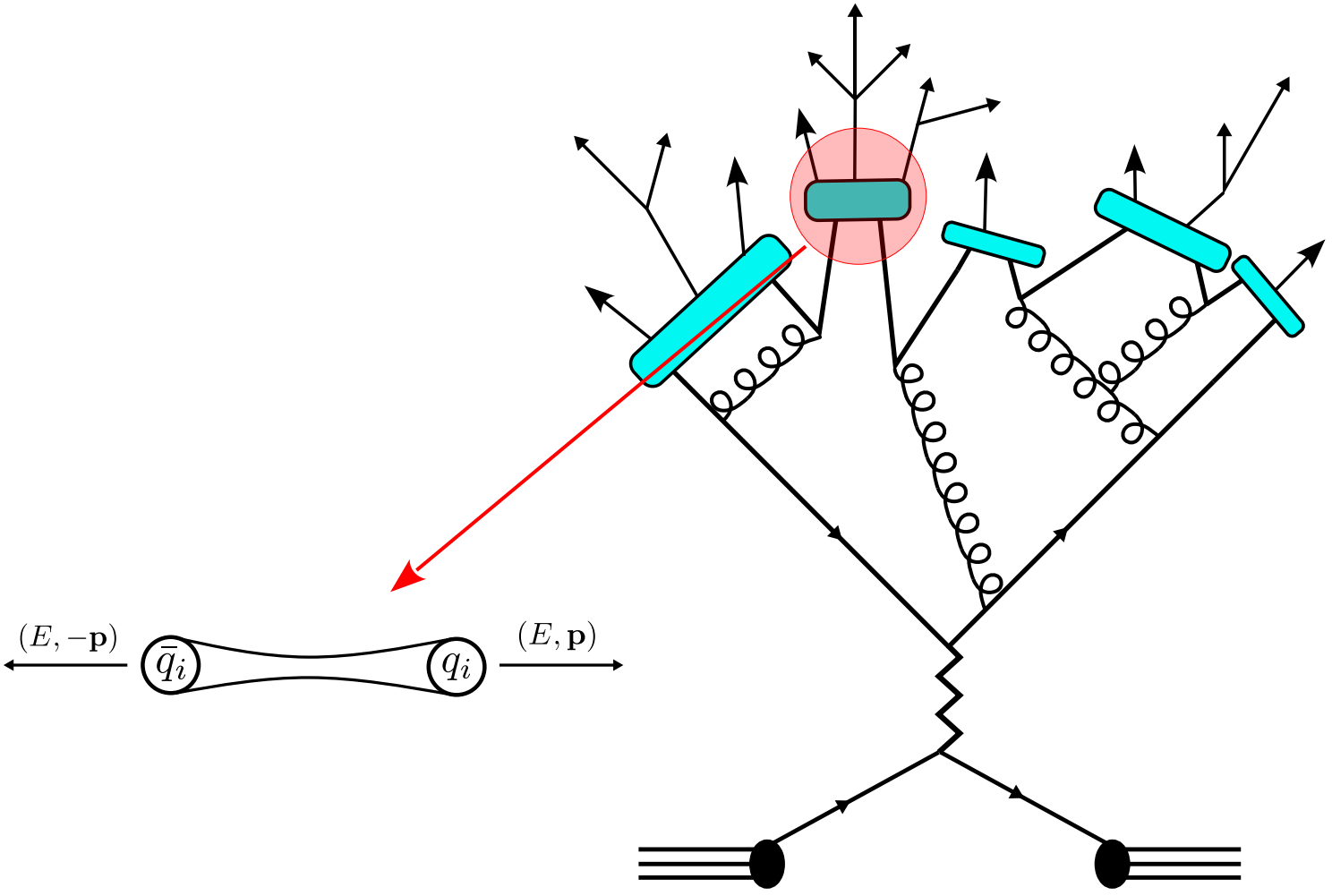
Cluster model

(used in Herwig)

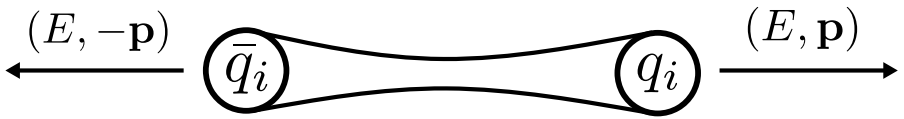
HadML collaboration

For ML methods applied to the cluster model:

2203.12660, 2305.17169, 2312.08453


$(E, -\mathbf{p})$
$\bar{q}_i$
q_i
$(E, \mathbf{p})$

The algorithm ($q\bar{q}$)



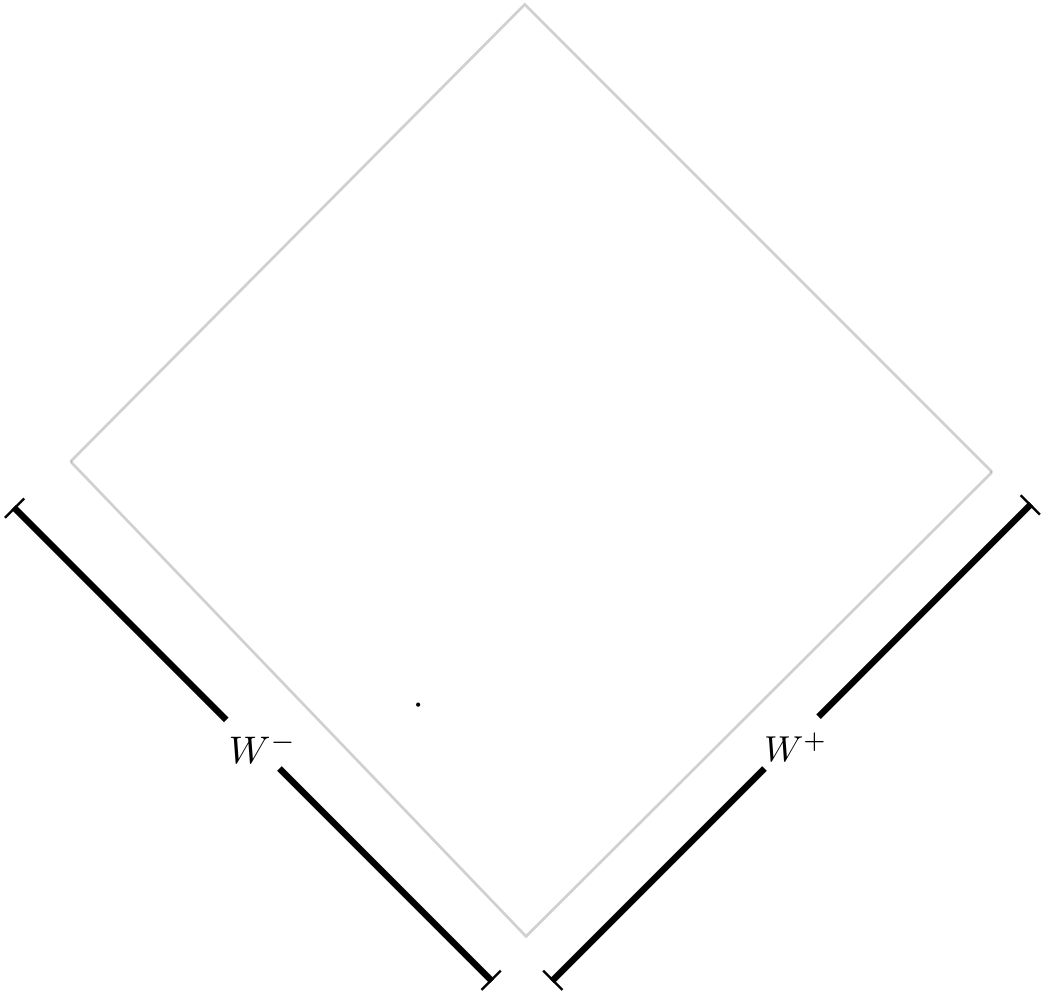
1) Randomly select one of the string ends
2) Sample new quark flavor
3) Sample transverse momentum of new quarks

$$\mathcal{P}(p_x, p_y; \sigma_{p_T}) = \frac{1}{\pi\sigma_{p_T}^2} \exp\left(-\frac{p_x^2+p_y^2}{\sigma_{p_T}^2}\right)$$

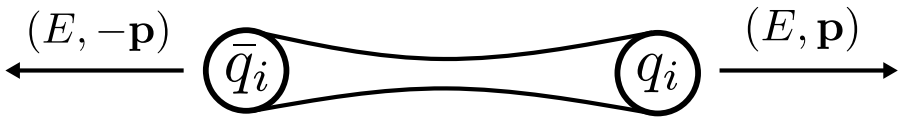
4) Sample longitudinal momentum fraction of new hadron

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_T^2}{z}\right), \; z = \frac{p_z + E_h}{2E}$$

5) Repeat steps 1-4



The algorithm ($q\bar{q}$)



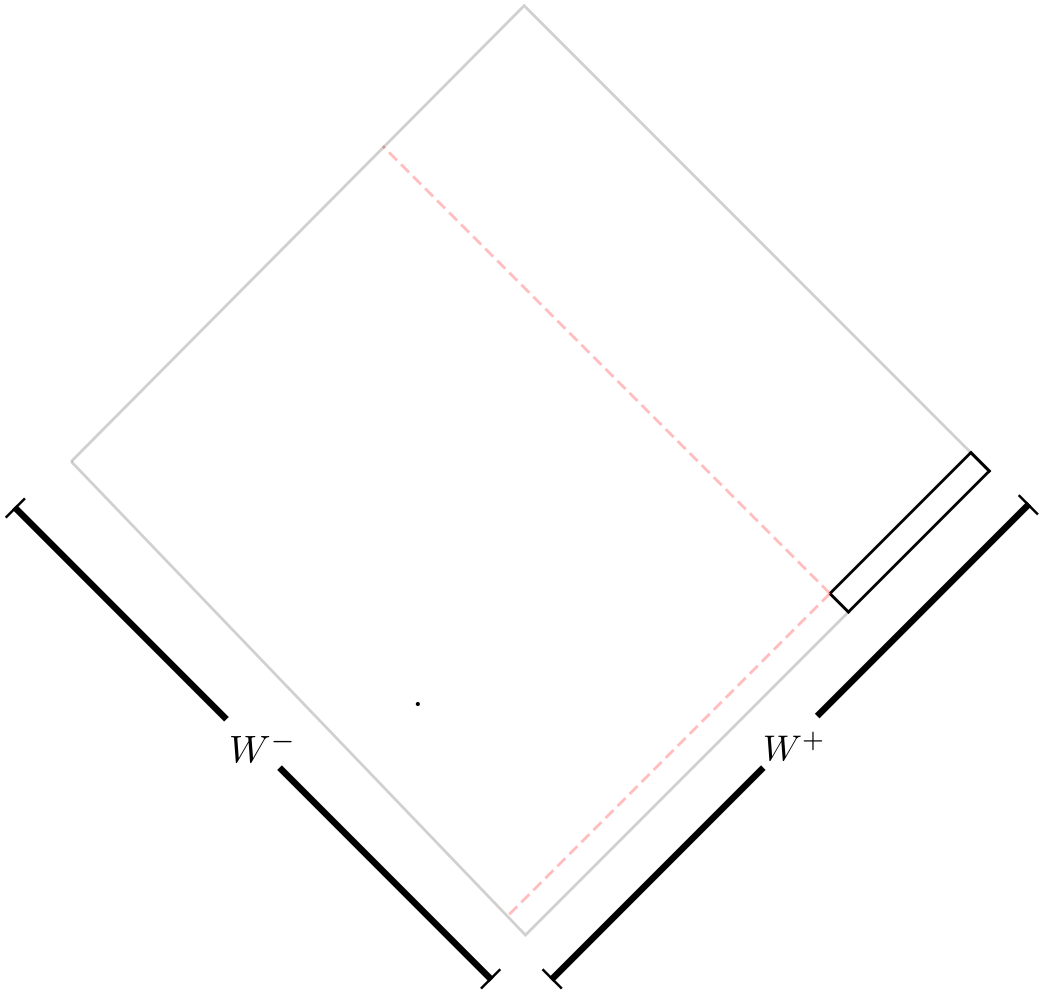
1) Randomly select one of the string ends
2) Sample new quark flavor
3) Sample transverse momentum of new quarks

$$\mathcal{P}(p_x, p_y; \sigma_{p_T}) = \frac{1}{\pi\sigma_{p_T}^2} \exp\left(-\frac{p_x^2+p_y^2}{\sigma_{p_T}^2}\right)$$

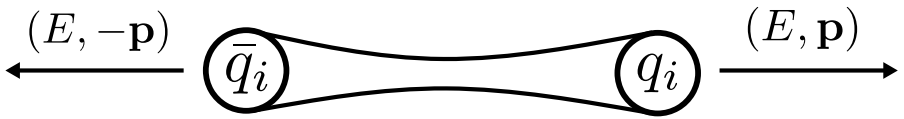
4) Sample longitudinal momentum fraction of new hadron

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_T^2}{z}\right), \quad z = \frac{p_z + E_h}{2E}$$

5) Repeat steps 1-4



The algorithm ($q\bar{q}$)



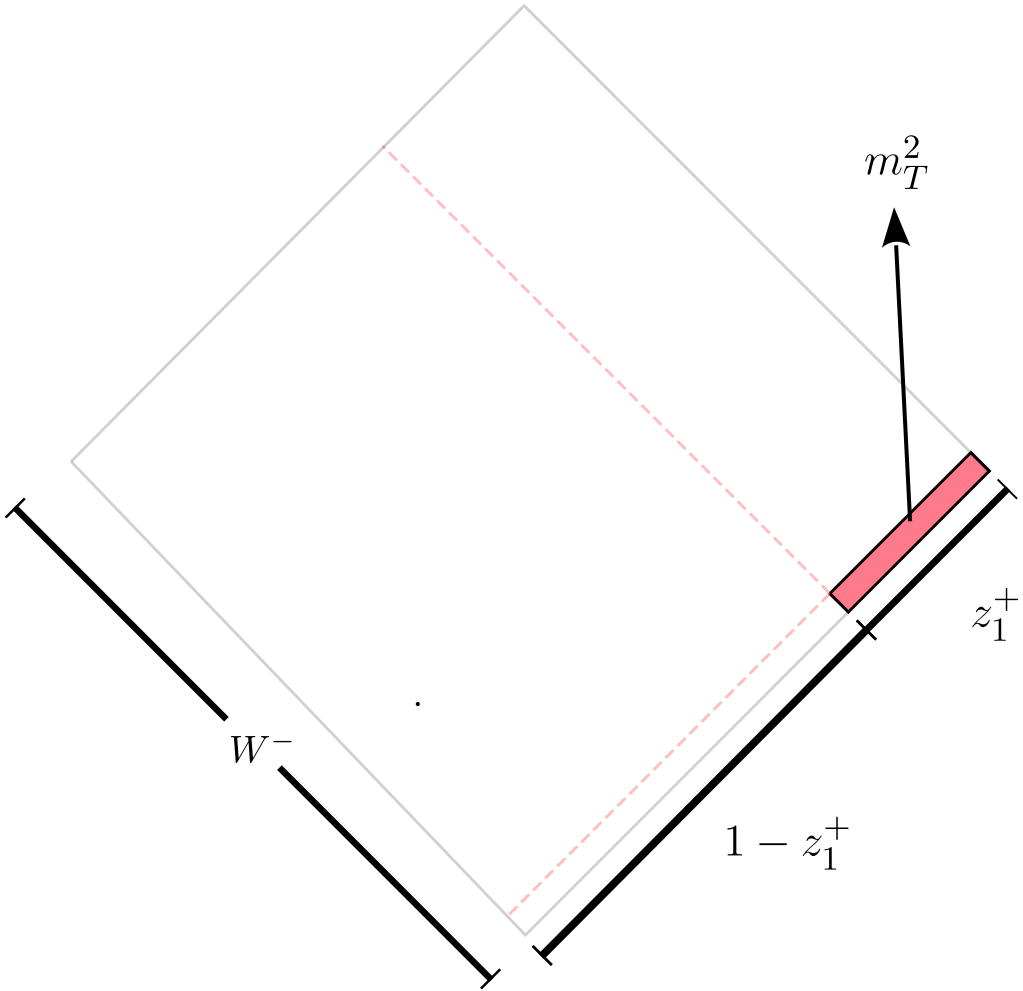
1) Randomly select one of the string ends
2) Sample new quark flavor
3) Sample transverse momentum of new quarks

$$\mathcal{P}(p_x, p_y; \sigma_{p_T}) = \frac{1}{\pi\sigma_{p_T}^2} \exp\left(-\frac{p_x^2+p_y^2}{\sigma_{p_T}^2}\right)$$

4) Sample longitudinal momentum fraction of new hadron

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_T^2}{z}\right), \quad z = \frac{p_z + E_h}{2E}$$

5) Repeat steps 1-4



The algorithm ($q\bar{q}$)

$(E, -\mathbf{p})$ $\bar{q}_i$

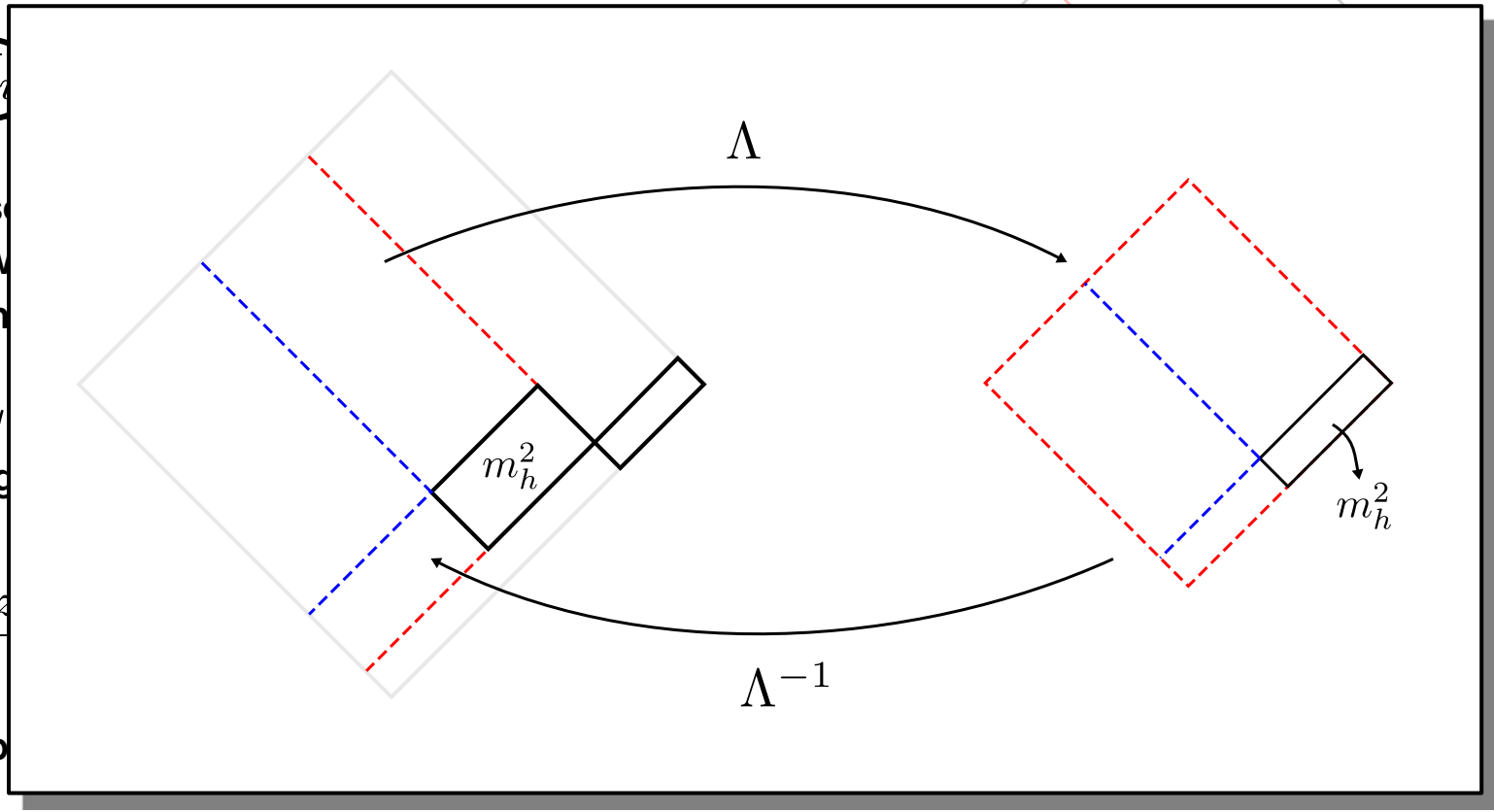
1) Randomly s
2) Sample new
3) Sample tran

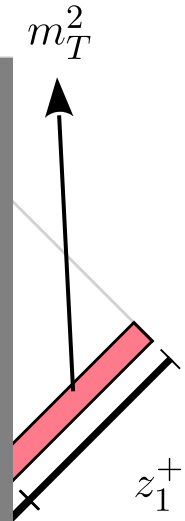
$\mathcal{P}(p_x, p_y$

4) Sample long
new hadron

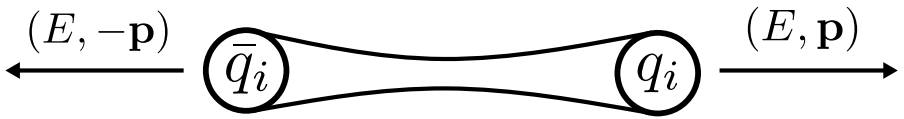
$f(z) \propto \frac{(1-z}{z}$

5) Repeat step





The algorithm ($q\bar{q}$)



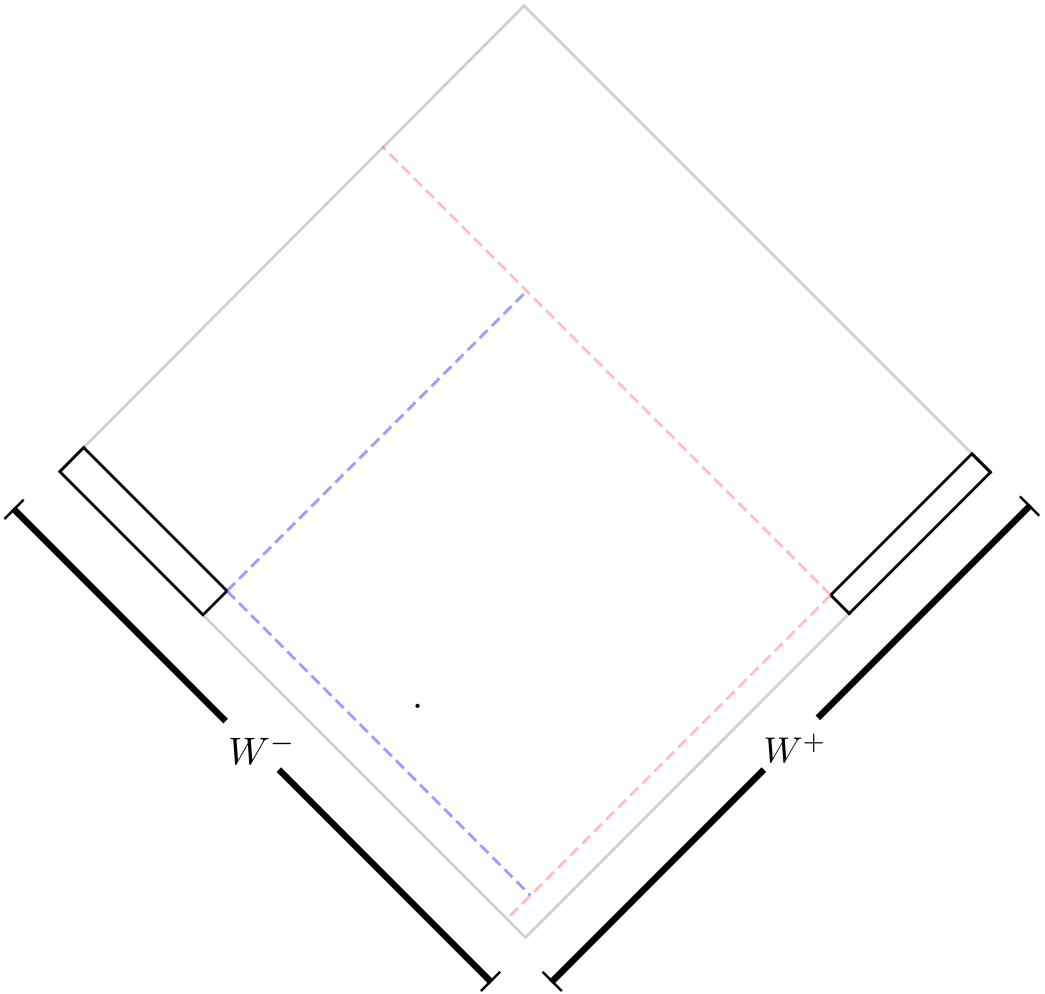
1) Randomly select one of the string ends
2) Sample new quark flavor
3) Sample transverse momentum of new quarks

$$\mathcal{P}(p_x, p_y; \sigma_{p_T}) = \frac{1}{\pi\sigma_{p_T}^2} \exp\left(-\frac{p_x^2+p_y^2}{\sigma_{p_T}^2}\right)$$

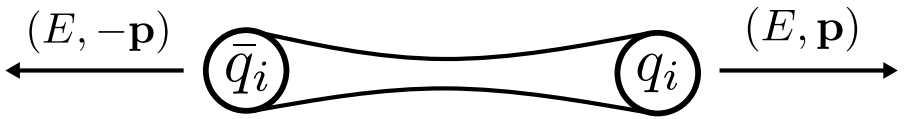
4) Sample longitudinal momentum fraction of new hadron

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_T^2}{z}\right), \quad z = \frac{p_z + E_h}{2E}$$

5) Repeat steps 1-4



The algorithm ($q\bar{q}$)



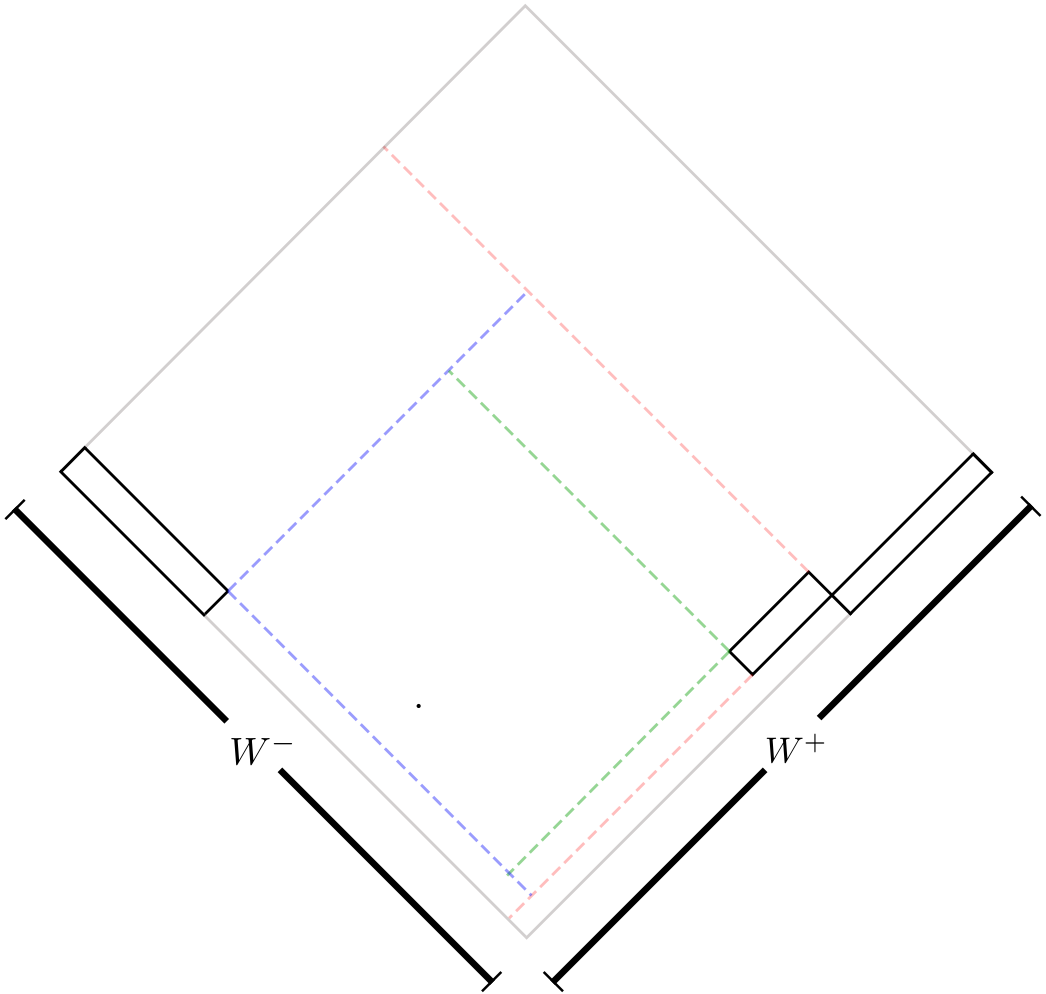
1) Randomly select one of the string ends
2) Sample new quark flavor
3) Sample transverse momentum of new quarks

$$\mathcal{P}(p_x, p_y; \sigma_{p_T}) = \frac{1}{\pi\sigma_{p_T}^2} \exp\left(-\frac{p_x^2+p_y^2}{\sigma_{p_T}^2}\right)$$

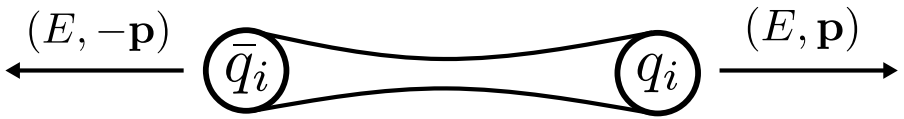
4) Sample longitudinal momentum fraction of new hadron

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_T^2}{z}\right), \quad z = \frac{p_z + E_h}{2E}$$

5) Repeat steps 1-4



The algorithm ($q\bar{q}$)



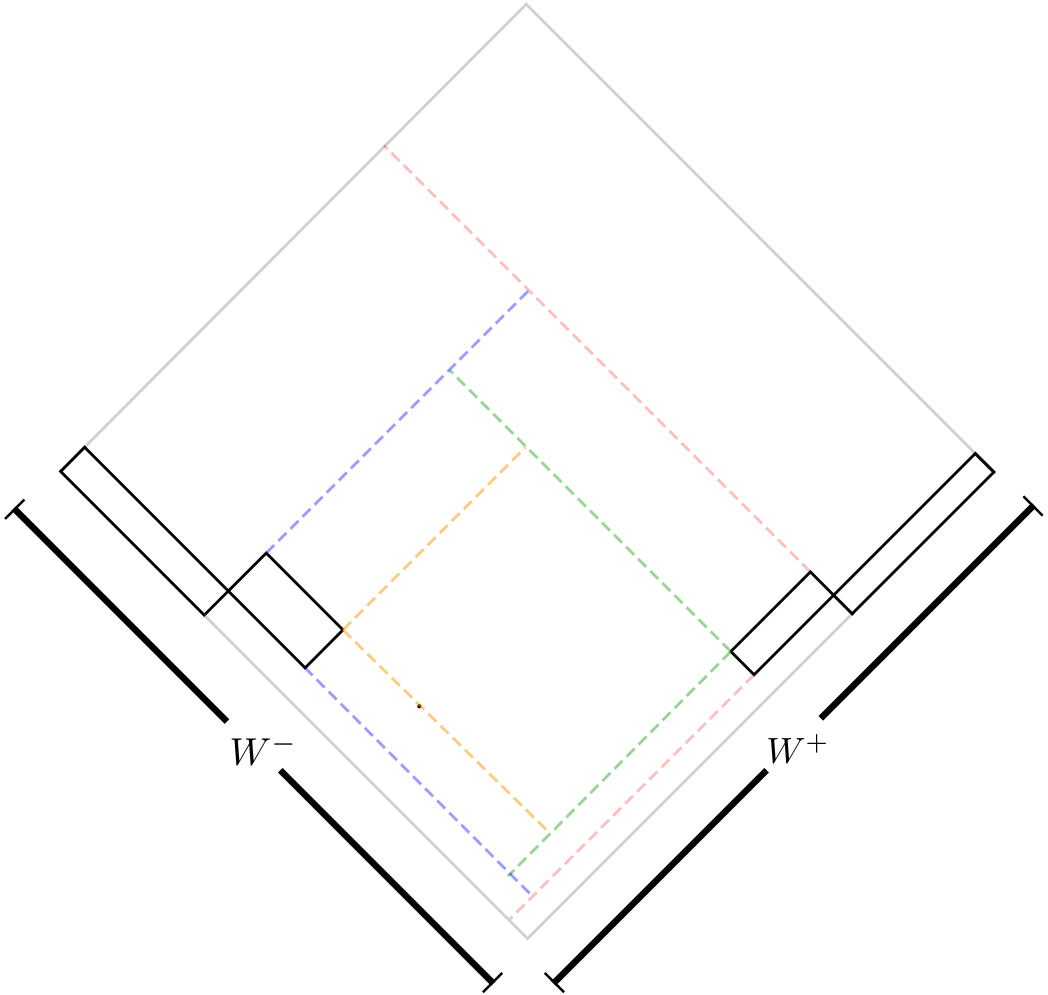
1) Randomly select one of the string ends
2) Sample new quark flavor
3) Sample transverse momentum of new quarks

$$\mathcal{P}(p_x, p_y; \sigma_{p_T}) = \frac{1}{\pi\sigma_{p_T}^2} \exp\left(-\frac{p_x^2+p_y^2}{\sigma_{p_T}^2}\right)$$

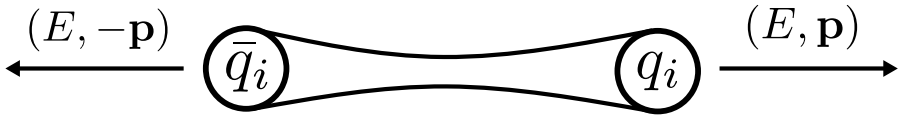
4) Sample longitudinal momentum fraction of new hadron

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_T^2}{z}\right), \quad z = \frac{p_z + E_h}{2E}$$

5) Repeat steps 1-4



The algorithm ($q\bar{q}$)



1) Randomly select one of the string ends
2) Sample new quark flavor
3) Sample transverse momentum of new quarks

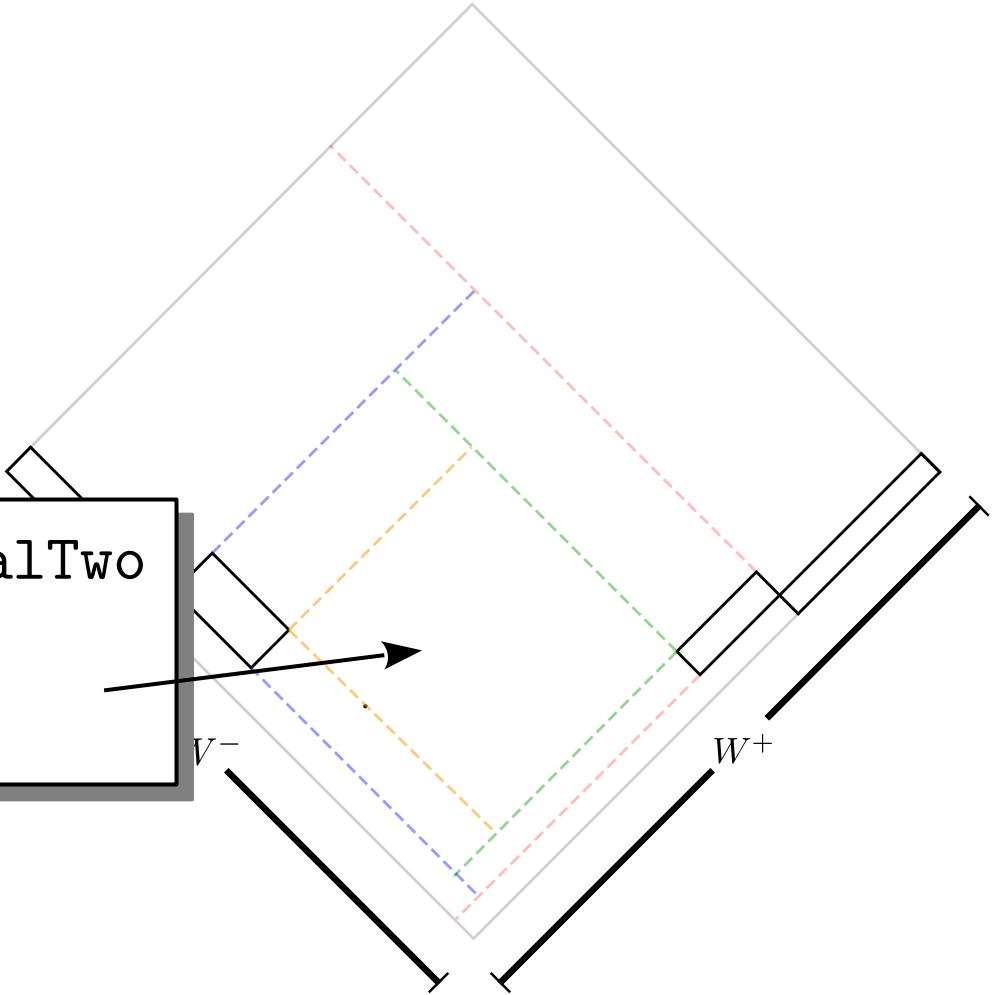
$$\mathcal{P}($$

4) Samp
new h

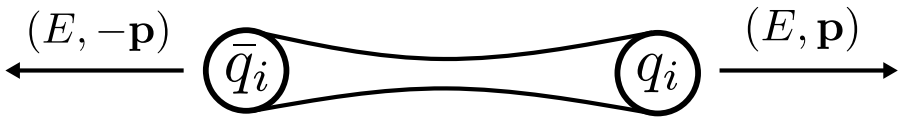
$$f(z) \propto \frac{}{z} \exp\left(\frac{}{z}\right), \quad z = \frac{}{2E}$$

5) Repeat steps 1-4

When E_{CM} goes below ~2 GeV, `finalTwo` is called. This can fail, when it does, the full string system is re-simulated from the beginning.



The algorithm ($q\bar{q}$)



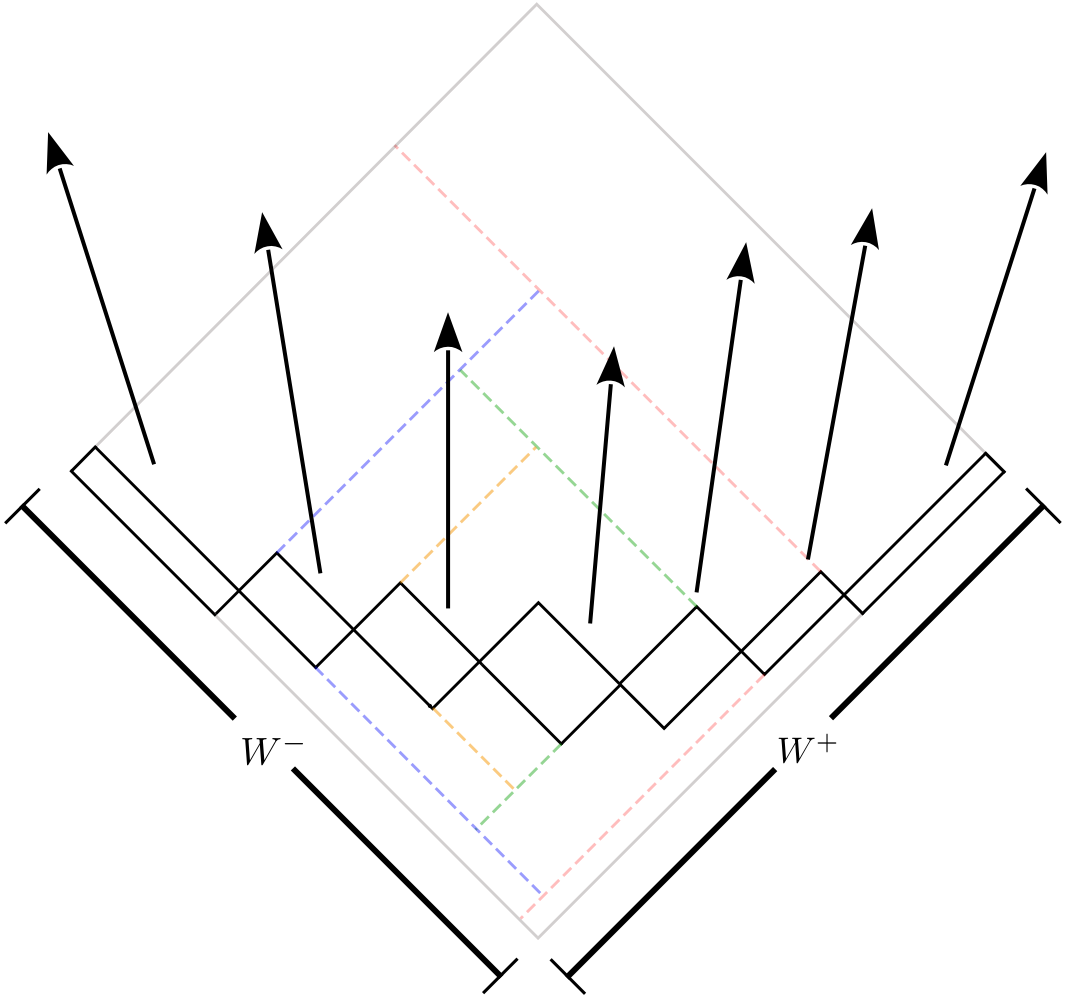
1) Randomly select one of the string ends
2) Sample new quark flavor
3) Sample transverse momentum of new quarks

$$\mathcal{P}(p_x, p_y; \sigma_{p_T}) = \frac{1}{\pi\sigma_{p_T}^2} \exp\left(-\frac{p_x^2+p_y^2}{\sigma_{p_T}^2}\right)$$

4) Sample longitudinal momentum fraction of new hadron

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_T^2}{z}\right), \quad z = \frac{p_z + E_h}{2E}$$

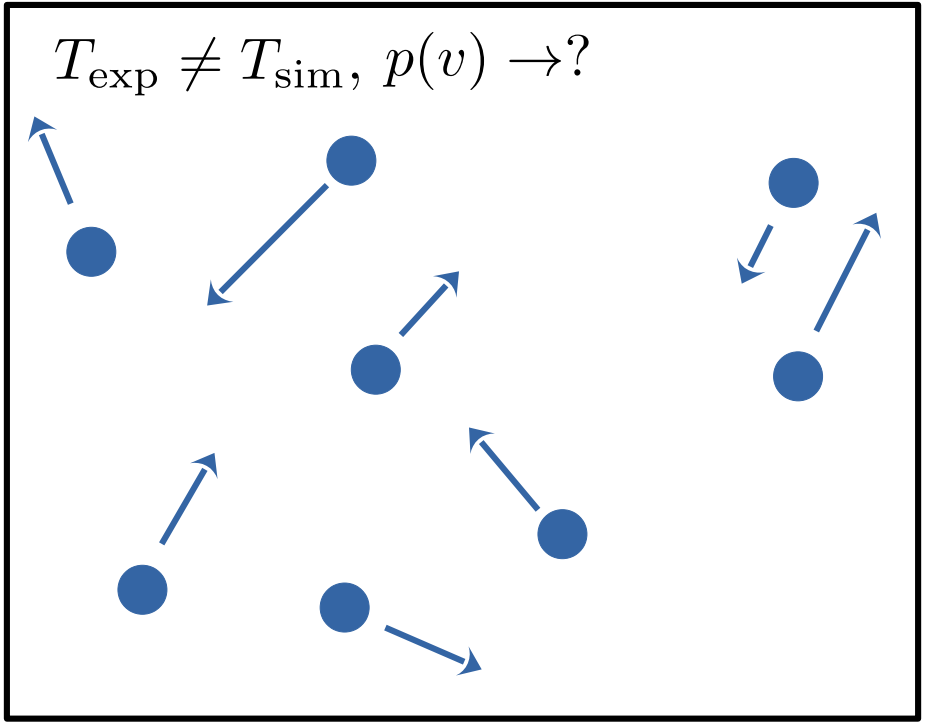
5) Repeat steps 1-4



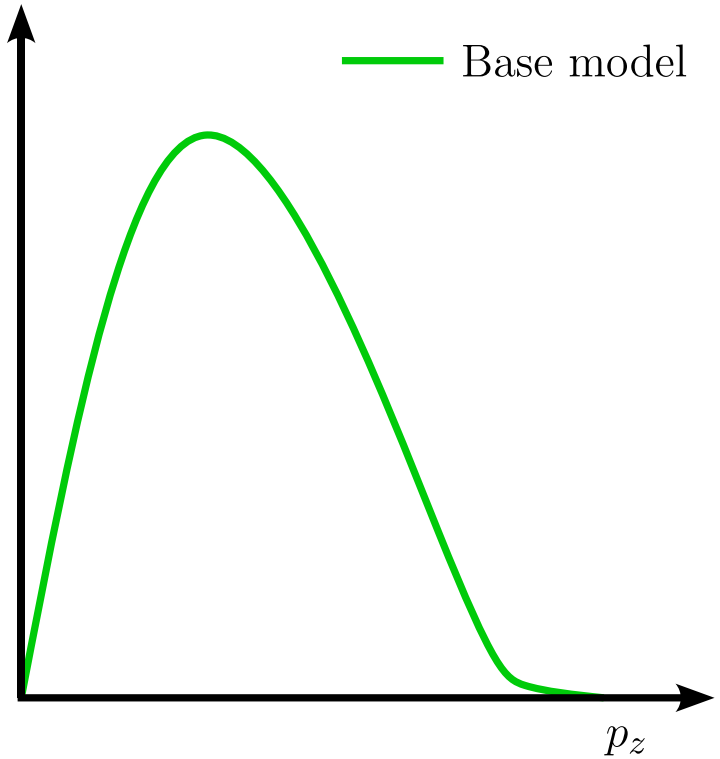
The inverse problem

Task: Given a discrepant macroscopic observable between simulation and experiment, find a way to modify the microscopic dynamics of the simulation to accommodate the observable.

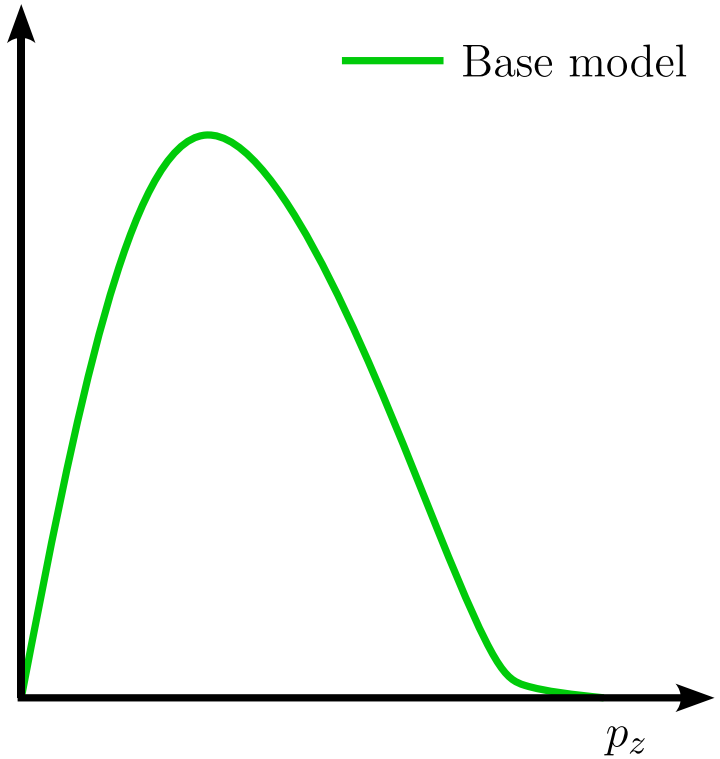
Akin to...

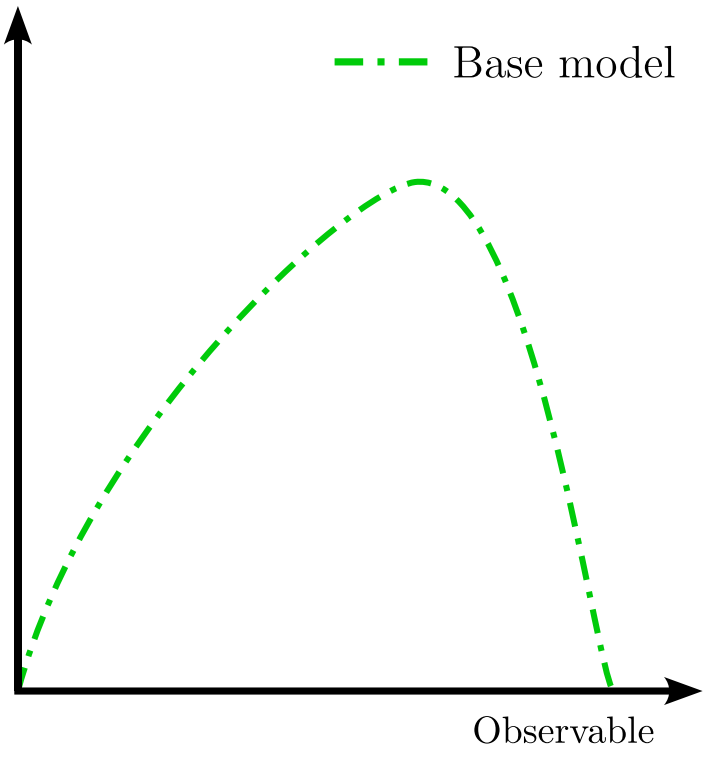


Micro from macro: big picture

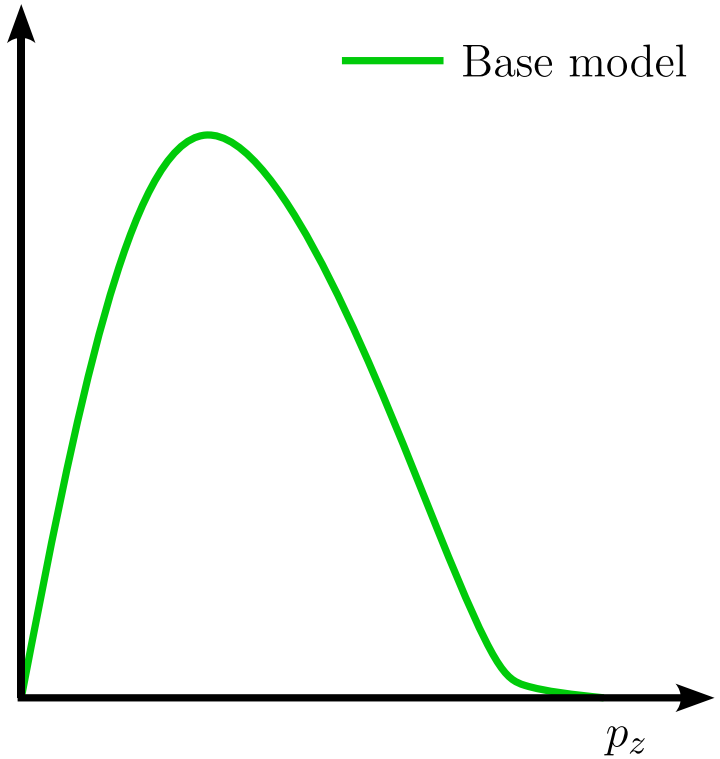


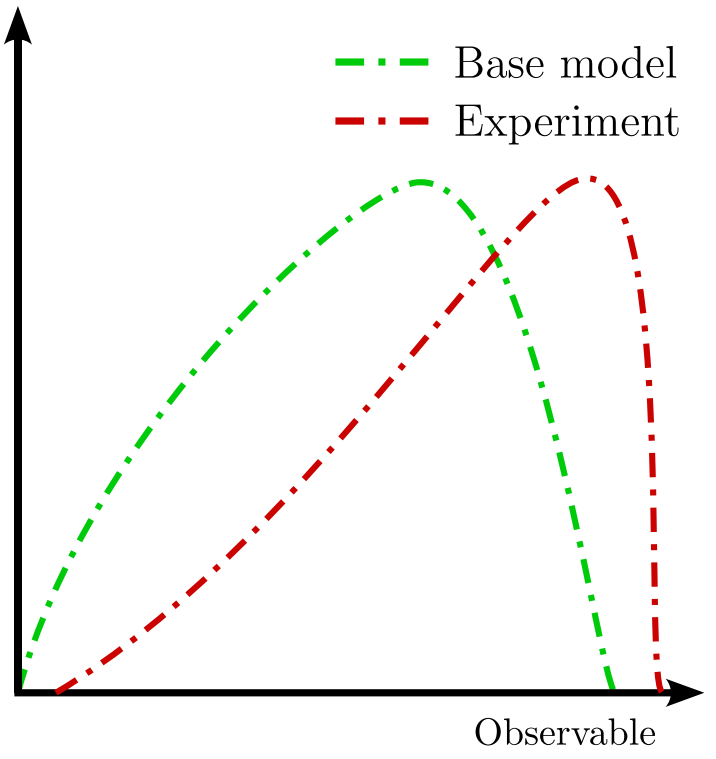
Micro from macro: big picture



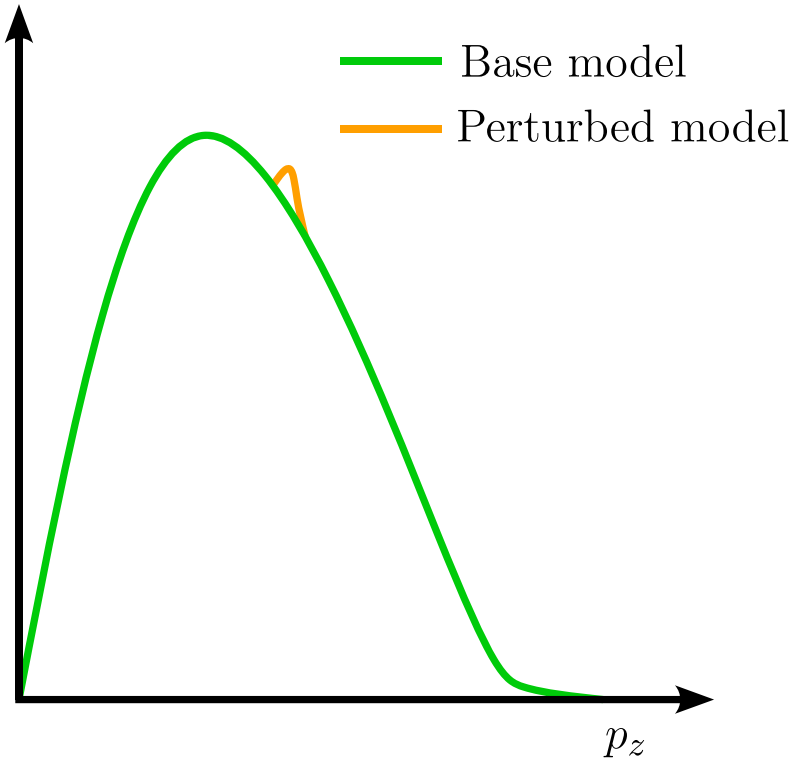


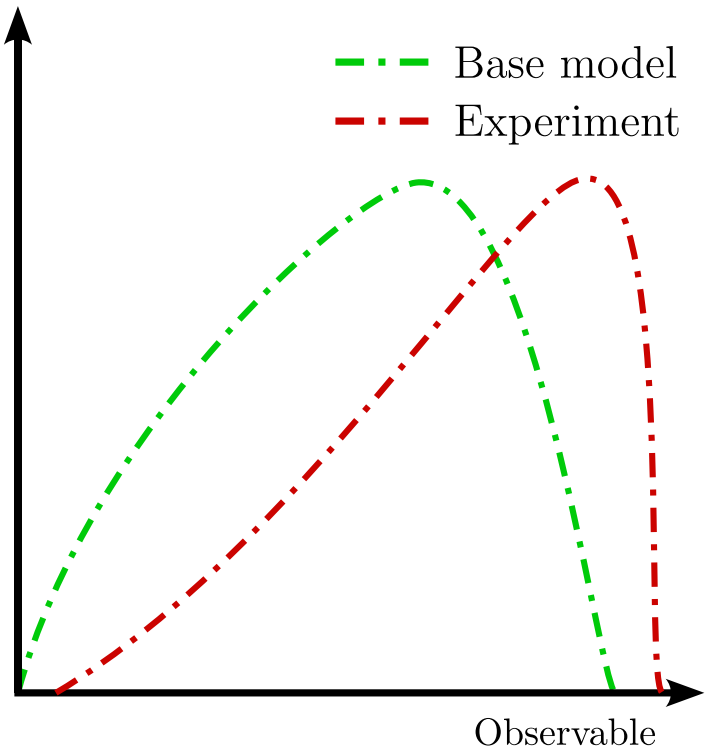
Micro from macro: big picture



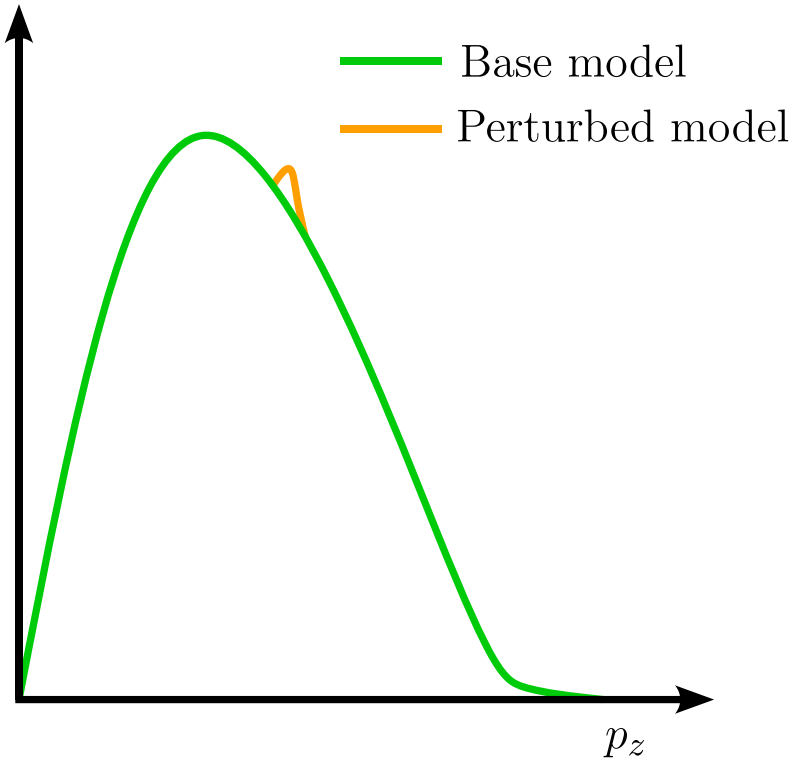


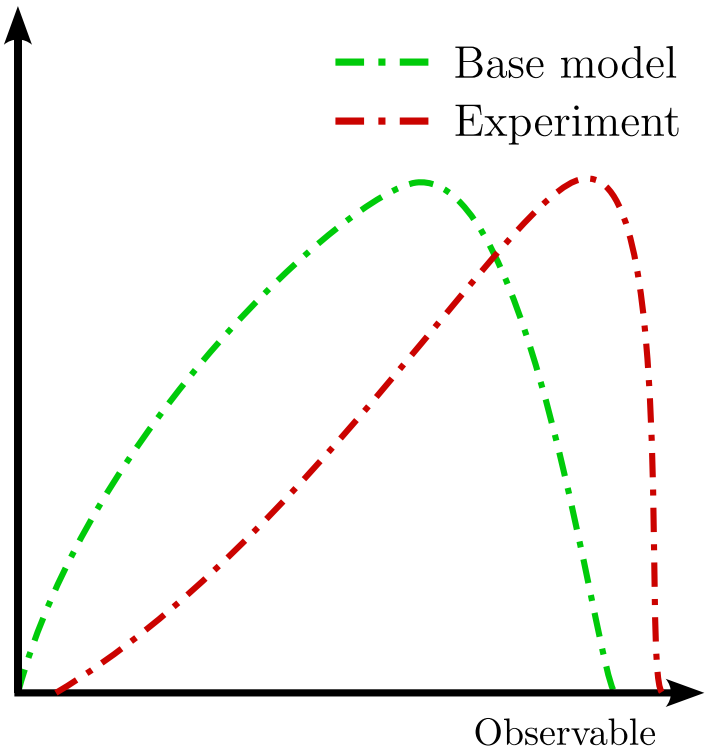
Micro from macro: big picture



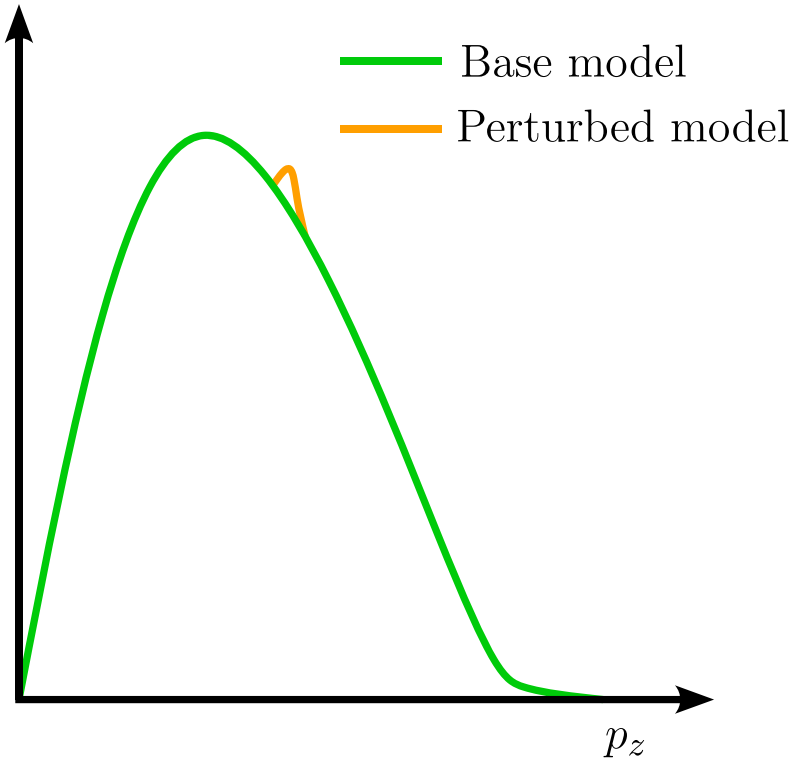


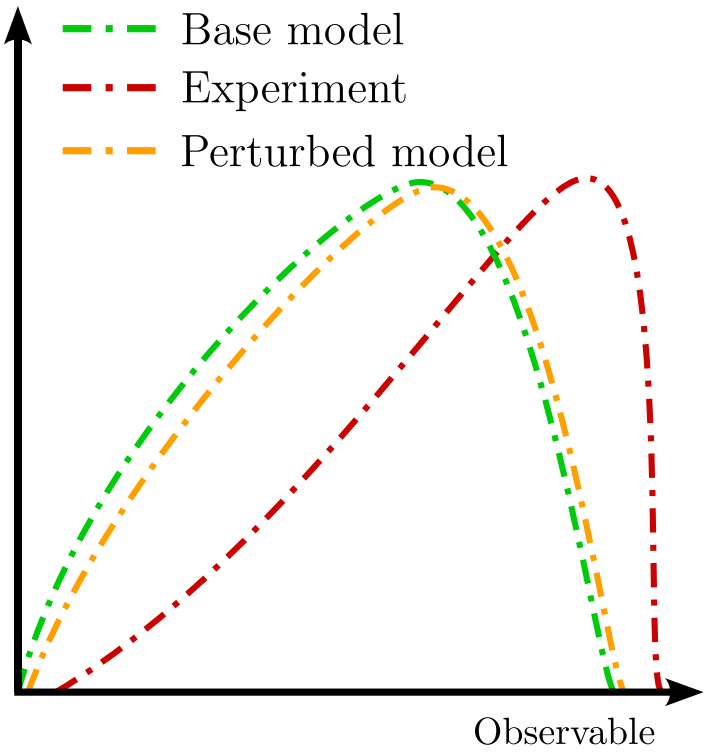
Micro from macro: big picture



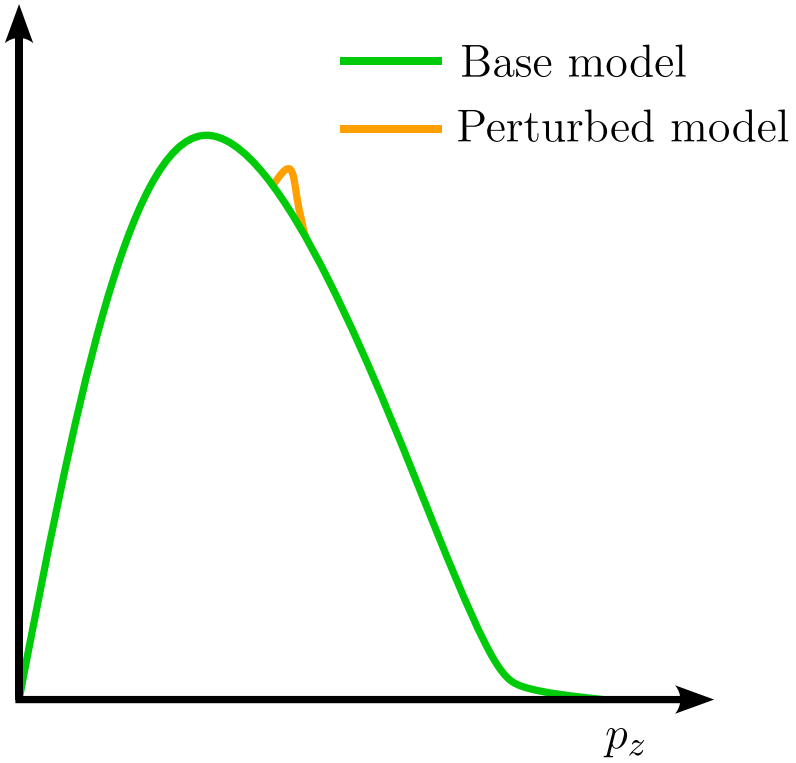


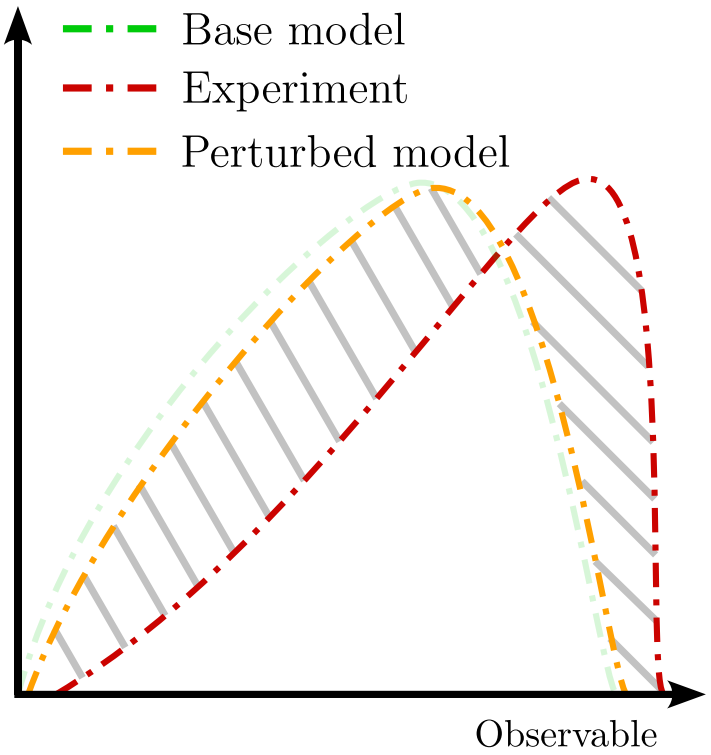
Micro from macro: big picture



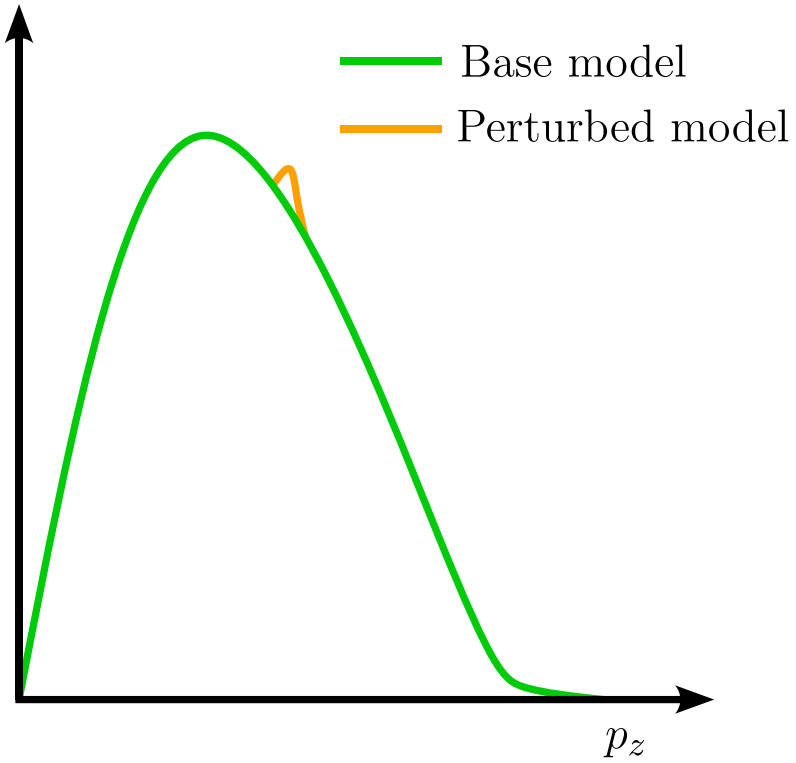


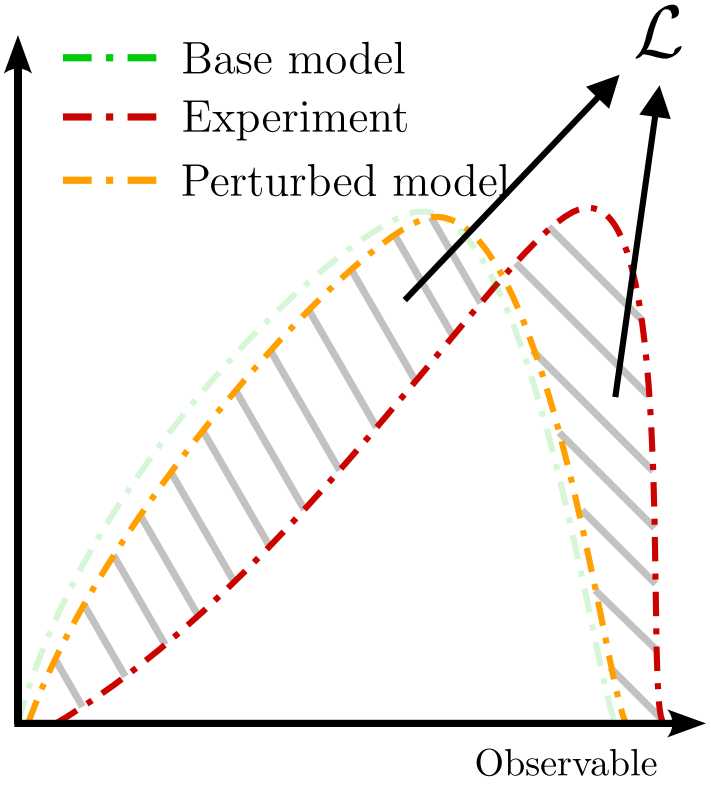
Micro from macro: big picture



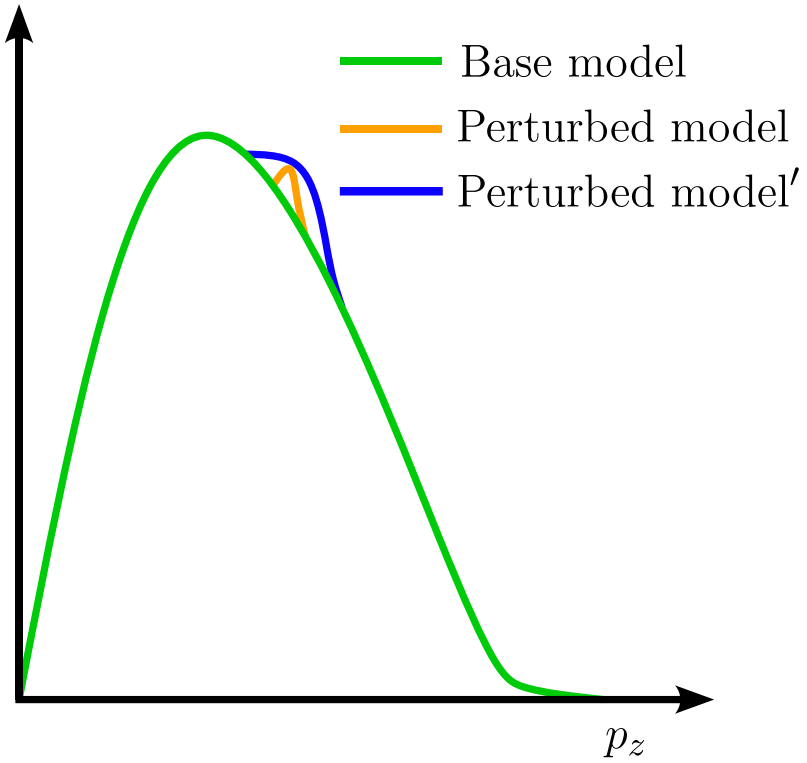


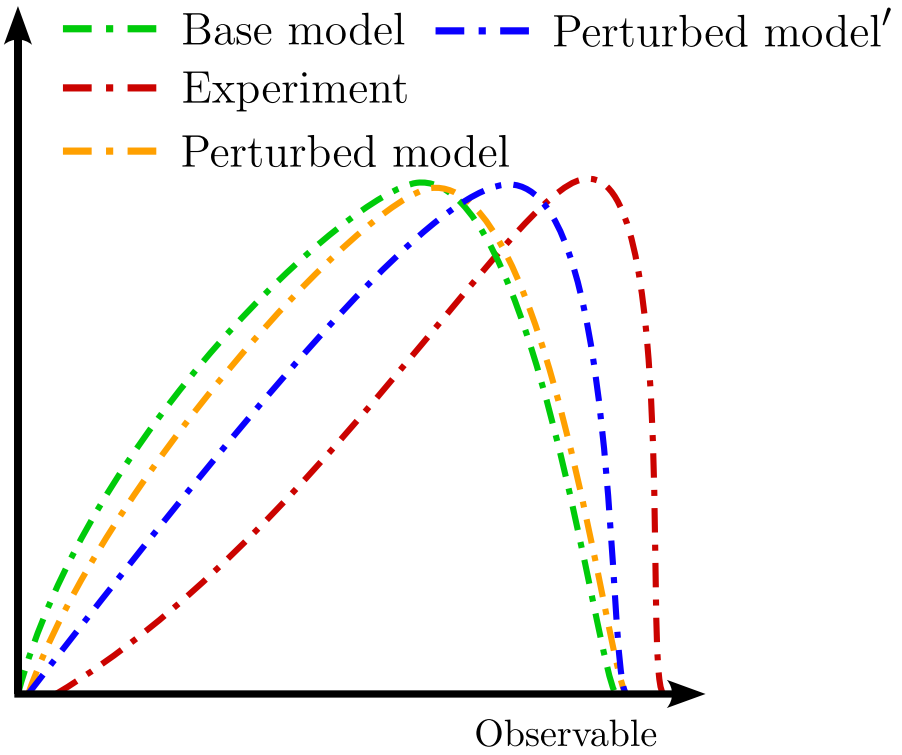
Micro from macro: big picture



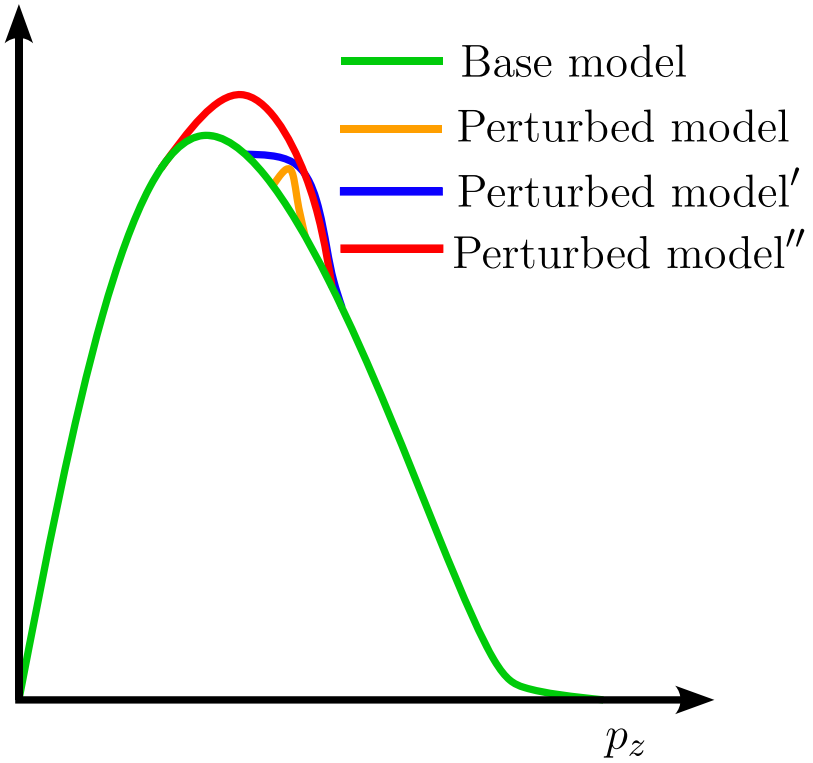


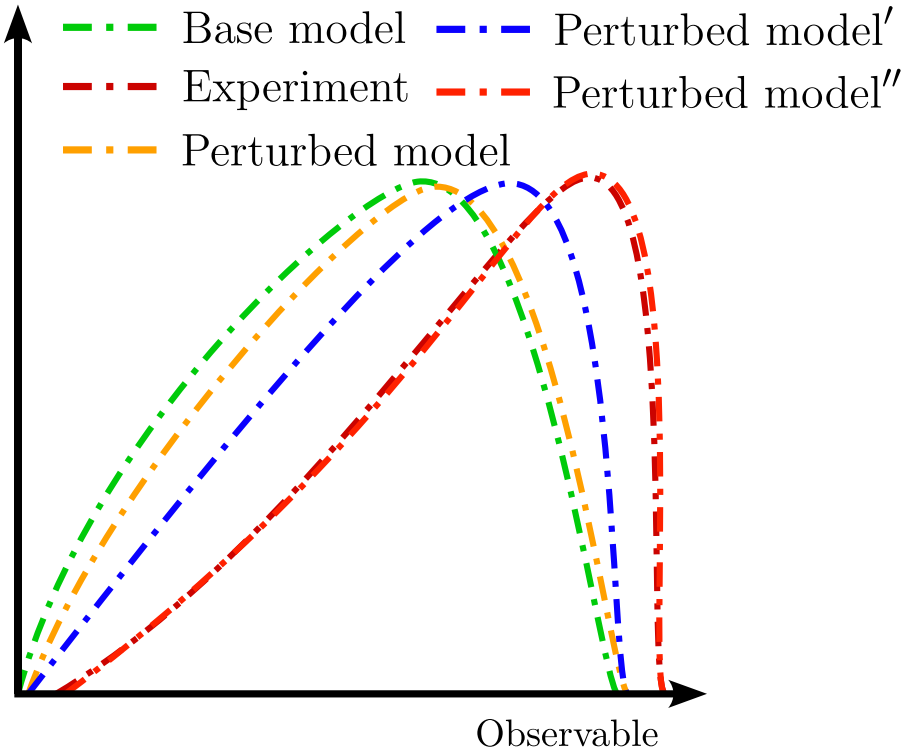
Micro from macro: big picture



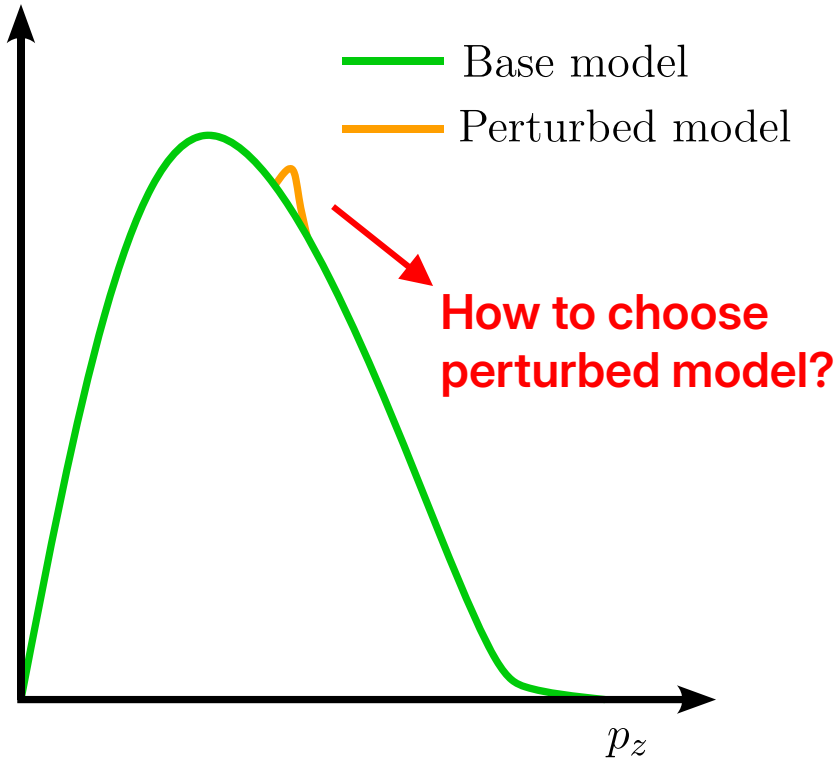


Micro from macro: big picture



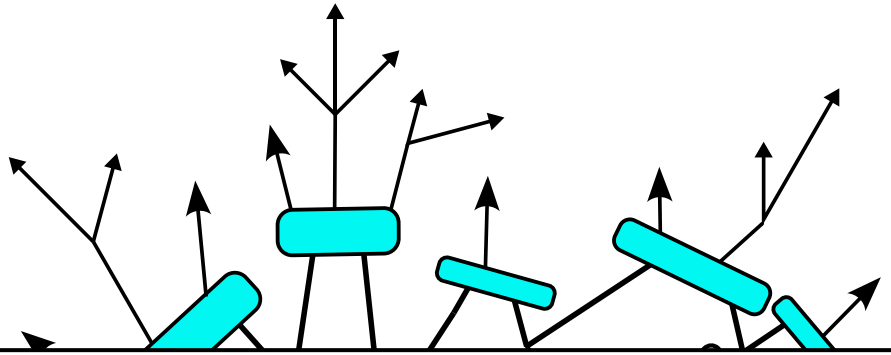


Micro from macro: details



1) Perform an analytic fit of the Lund parameters (traditional fit)

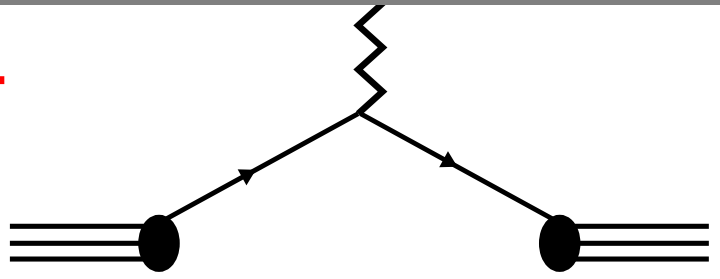
2) Perform an over-parameterized fit (ML fit)



1.

Rejection Sampling with Autodifferentiation (RSA)

2411.02194

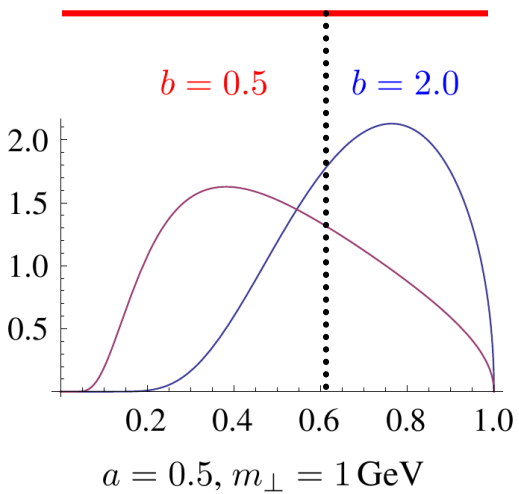


Kinematic reweighting (2308.13459)

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_\perp^2}{z}\right)$$

Can I understand the change in observable distributions for a different values of a and b WITHOUT re-simulating?

Compute event weights i.e. ratios of probabilities



Rejection sampling:

$$w_{\text{accept}} = \frac{P_{\text{accept}}(z;\theta')}{P_{\text{accept}}(z;\theta)}$$

Acceptance probability:

$$P_{\text{accept}} = \frac{p(z,\theta)}{\hat{P}}$$

$$w_{\text{reject}} = \frac{1 - P_{\text{accept}}(z;\theta')}{1 - P_{\text{accept}}(z;\theta)}$$

Rejection probability

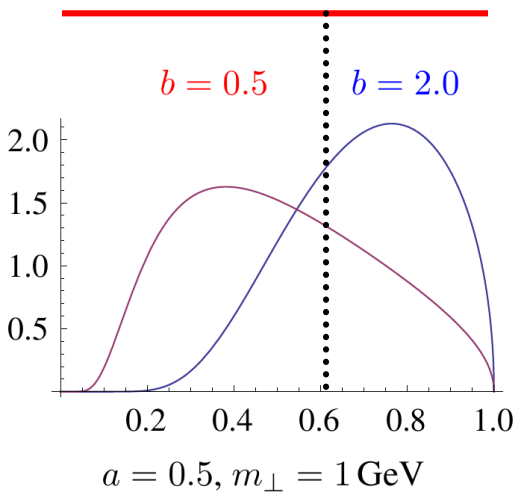
$$P_{\text{reject}} = 1 - P_{\text{accept}}$$

Kinematic reweighting (2308.13459)

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_\perp^2}{z}\right)$$

Can I understand the change in observable distributions for a different values of a and b WITHOUT re-simulating?

Compute event weights i.e. ratios of probabilities



Rejection sampling:

$$w_{\text{accept}} = \frac{p(z;\theta')}{p(z;\theta)}$$

Acceptance probability:

$$P_{\text{accept}} = \frac{p(z,\theta)}{\hat{P}}$$

$$w_{\text{reject}} = \frac{\hat{P} - p(z;\theta')}{\hat{P} - p(z;\theta)}$$

Rejection probability

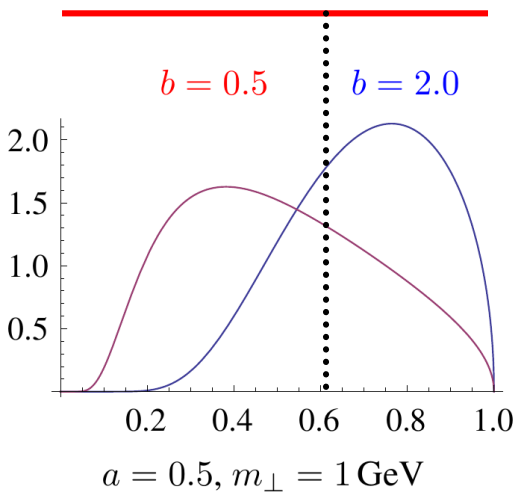
$$P_{\text{reject}} = 1 - P_{\text{accept}}$$

Kinematic reweighting (2308.13459)

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_\perp^2}{z}\right)$$

Can I understand the change in observable distributions for a different values of a and b WITHOUT re-simulating?

Compute event weights i.e. ratios of probabilities



Rejection sampling:

$$w_{\text{accept}} = \frac{p(z;\theta')}{p(z;\theta)}$$

Acceptance probability:

$$P_{\text{accept}} = \frac{p(z,\theta)}{\hat{P}}$$

$$w_{\text{reject}} = \frac{\hat{P} - p(z;\theta')}{\hat{P} - p(z;\theta)}$$

Rejection probability

$$P_{\text{reject}} = 1 - P_{\text{accept}}$$

$$w = w_{\text{accept}} \prod_{i=1}^{n_{\text{rej.}}} w_{\text{reject}}^i$$

Kinematic reweighting (2308.13459)

Can I understand the change in observable

Hadronization data-structure:

$$\boldsymbol{z} = \begin{pmatrix} \boldsymbol{z}_1 = \begin{pmatrix} \{m_T^{h_1}, z_{\text{accept}}^{h_1}, z_{\text{reject}}^{1,h_1}, \cdots, z_{\text{reject}}^{n_{h_1},h_1}\} \\ \{m_T^{h_2}, z_{\text{accept}}^{h_2}, z_{\text{reject}}^{1,h_2}, \cdots, z_{\text{reject}}^{n_{h_2},h_2}\} \\ \{m_T^{h_3}, z_{\text{accept}}^{h_3}, z_{\text{reject}}^{1,h_3}, \cdots, z_{\text{reject}}^{n_{h_3},h_3}\} \end{pmatrix}_1 \\ \boldsymbol{z}_2 = \begin{pmatrix} \{m_T^{h_1}, z_{\text{accept}}^{h_1}, z_{\text{reject}}^{1,h_1}, \cdots, z_{\text{reject}}^{n_{h_1},h_1}\} \\ \vdots \\ \{m_T^{h_4}, z_{\text{accept}}^{h_4}, z_{\text{reject}}^{1,h_4}, \cdots, z_{\text{reject}}^{n_{h_4},h_4}\} \end{pmatrix}_2 \\ \vdots \\ \boldsymbol{z}_N = \cdots \end{pmatrix}$$

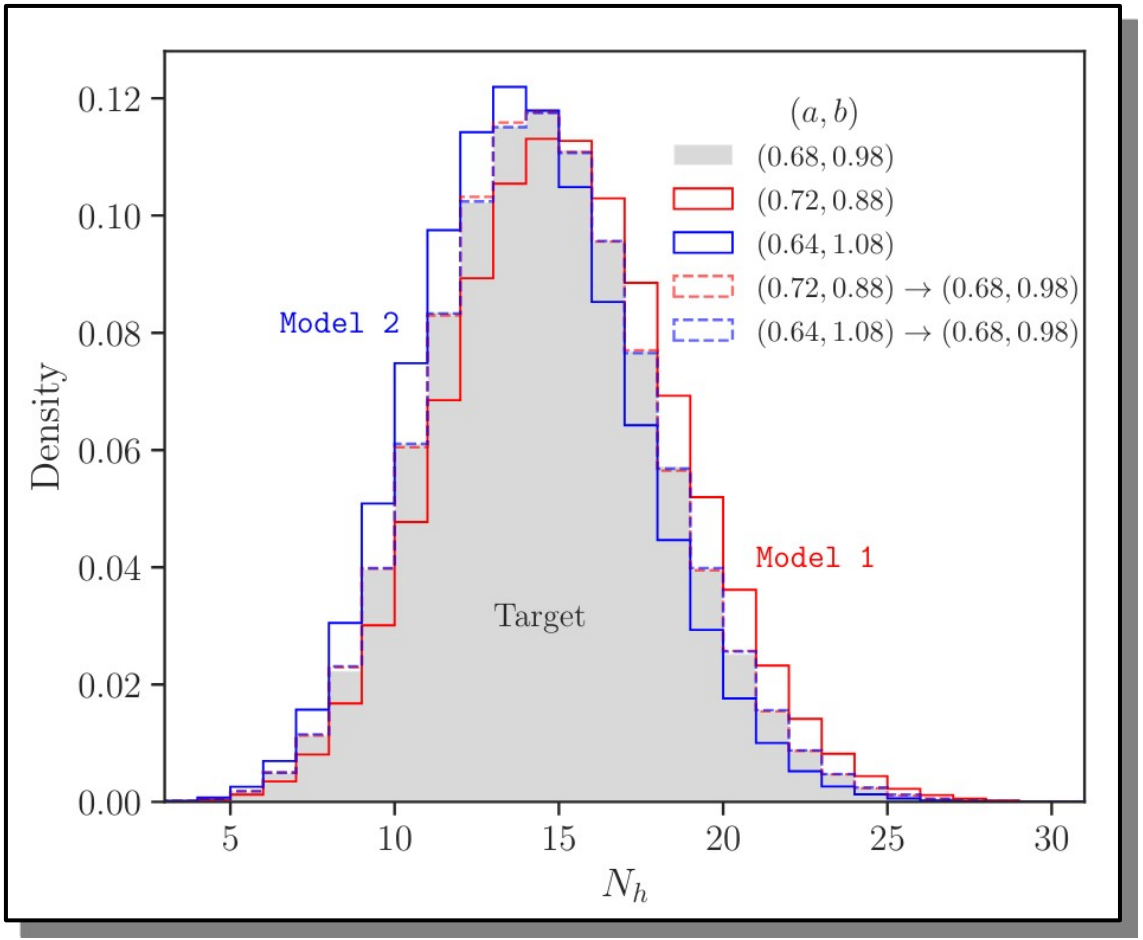
$$w_n = \prod_{i=1}^{\tilde{N}_{h,n}} \left(\frac{f(z_{\text{accept}}^{h_i}; \{a,b\}_P)}{f(z_{\text{accept}}^{h_i}; \{a,b\}_B)} \right) \times \prod_{j=1}^{n_{h_i}} \left(\frac{\hat{f} - f(z_{\text{reject}}^{j,h_i}; \{a,b\}_P)}{\hat{f} - f(z_{\text{reject}}^{j,h_i}; \{a,b\}_B)} \right)$$

$a = 0.5, m_\perp = 1\text{GeV}$

$$P_{\text{reject}} = 1 - P_{\text{accept}}$$

$$\prod_{i=1} w_{\text{reject}}$$

Kinematic reweighting (2308.13459)



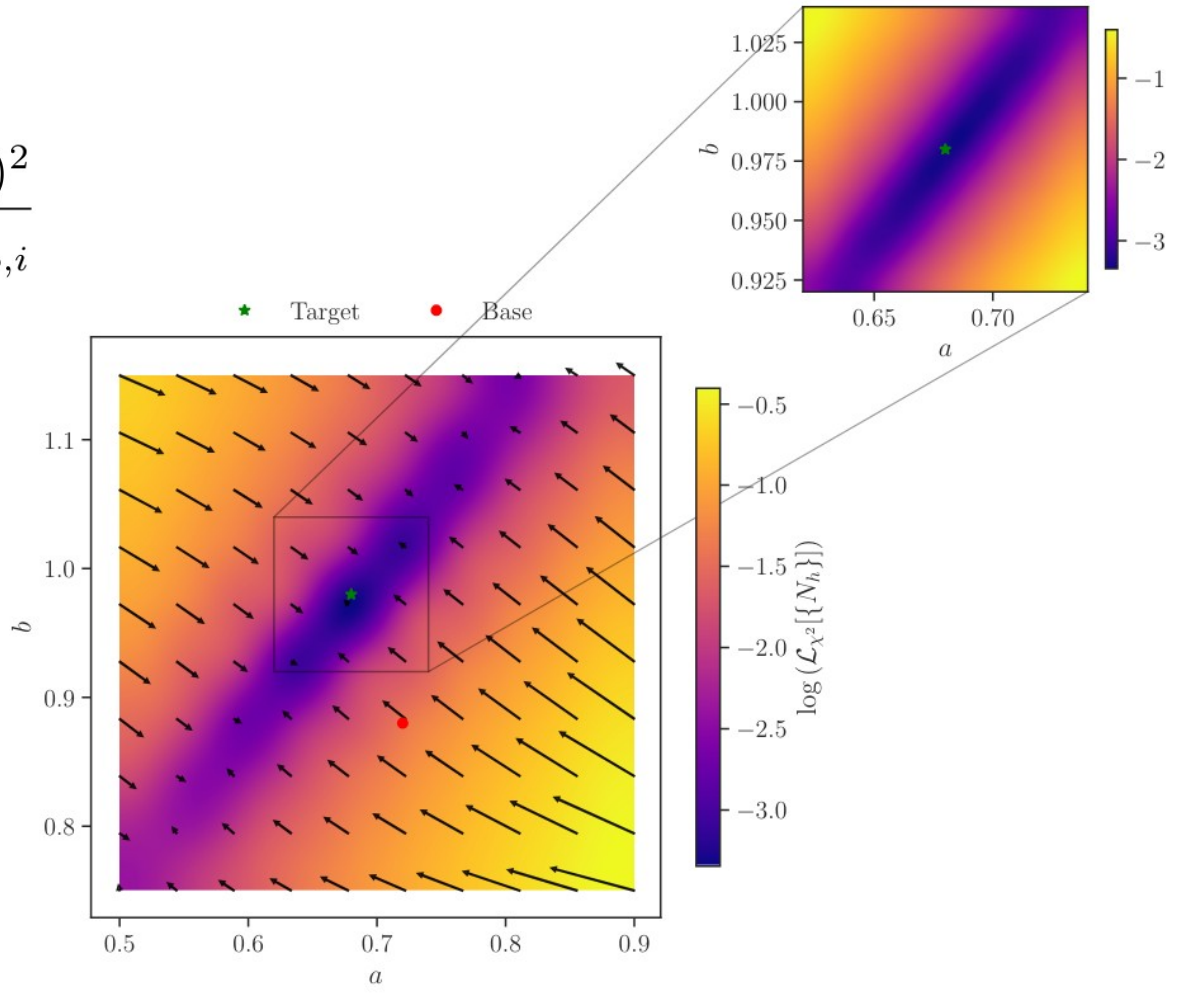
Applications

- Efficient means of exploring parameter space
- Inherently differentiable
- Very useful for tuning!
  - Embed the computation of weights into numerical autodifferentiation engine
  - Picking new parameters in the update step facilitated by well-developed optimizers (SGD, Adam, etc.)

Rejection sampling with Autodifferentiation (RSA)

χ^2-loss landscape

$$\mathcal{L}_{\chi^2}(\boldsymbol{y}_{\text{sim}}, \boldsymbol{y}_{\text{exp}}; \boldsymbol{w}) = \sum_{i=1}^{n_{\text{bins}}} \frac{(y_{\text{sim}}^{(i)} - y_{\text{exp}}^{(i)})^2}{\sigma_{\text{sim},i}^2 + \sigma_{\text{exp},i}^2}$$



The "score" observable

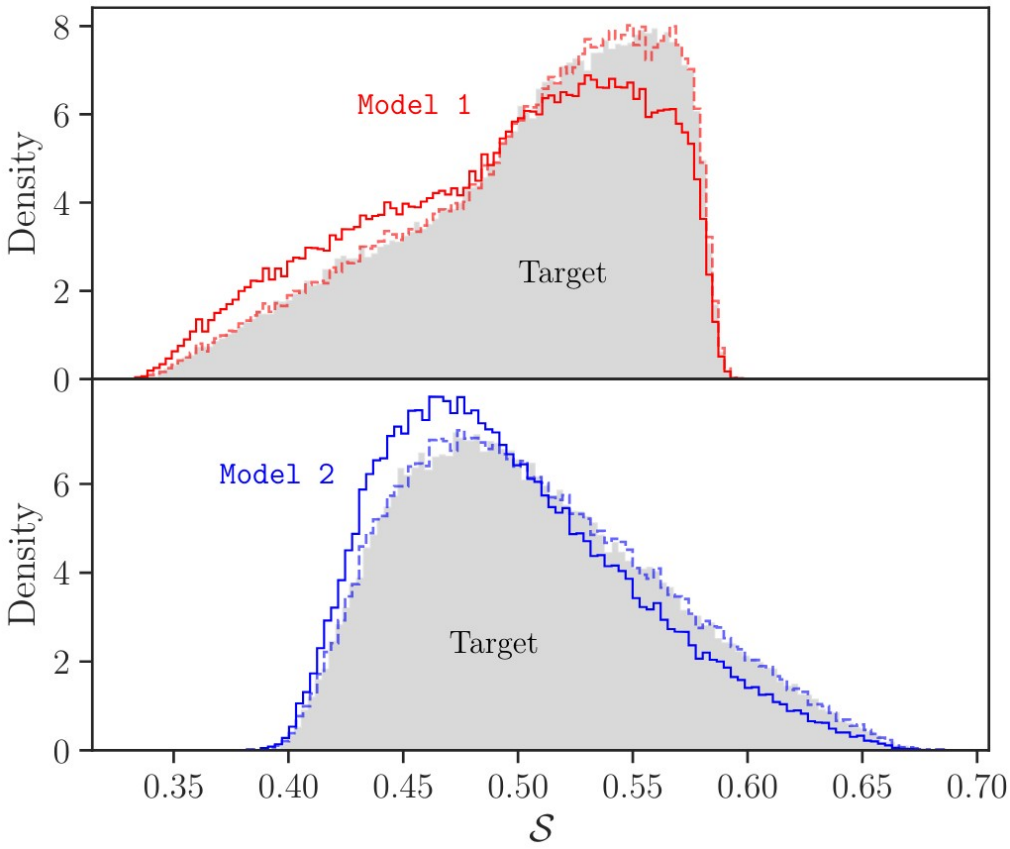
- Train a deepsets classifier to distinguish simulation from experiment
  - Takes full event information as input

$$(E, p_x, p_y, p_z)_1$$

$$(E, p_x, p_y, p_z)_2$$

$$\ldots$$

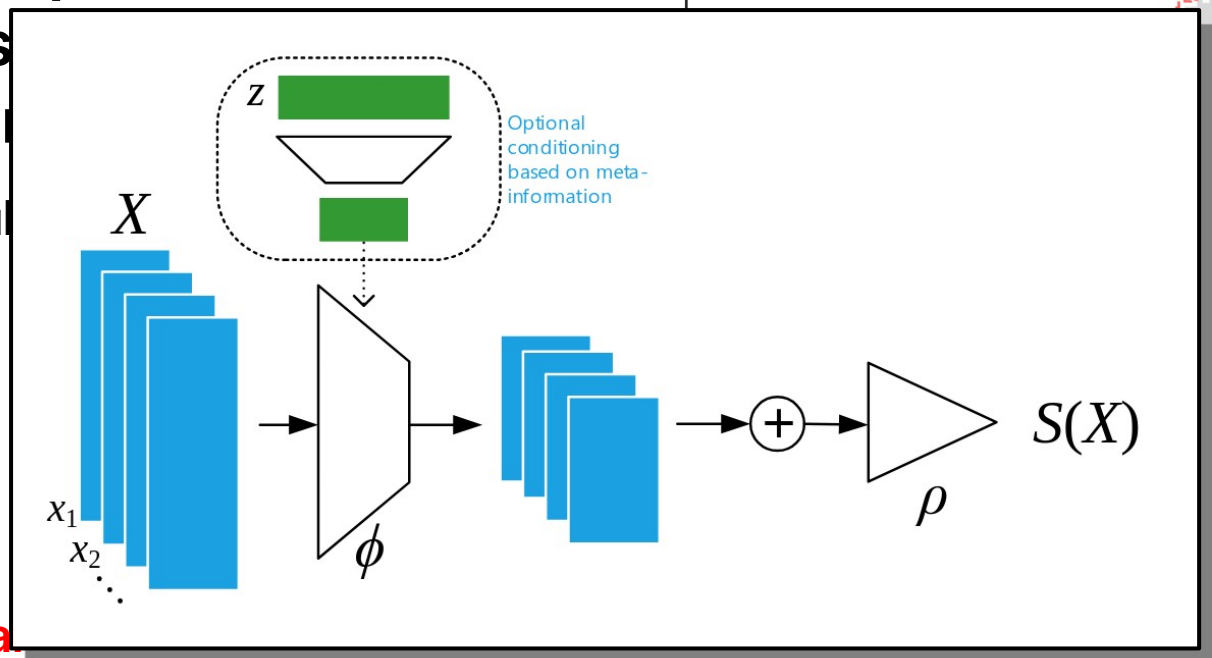
Use trained classifier output (score) as an observable

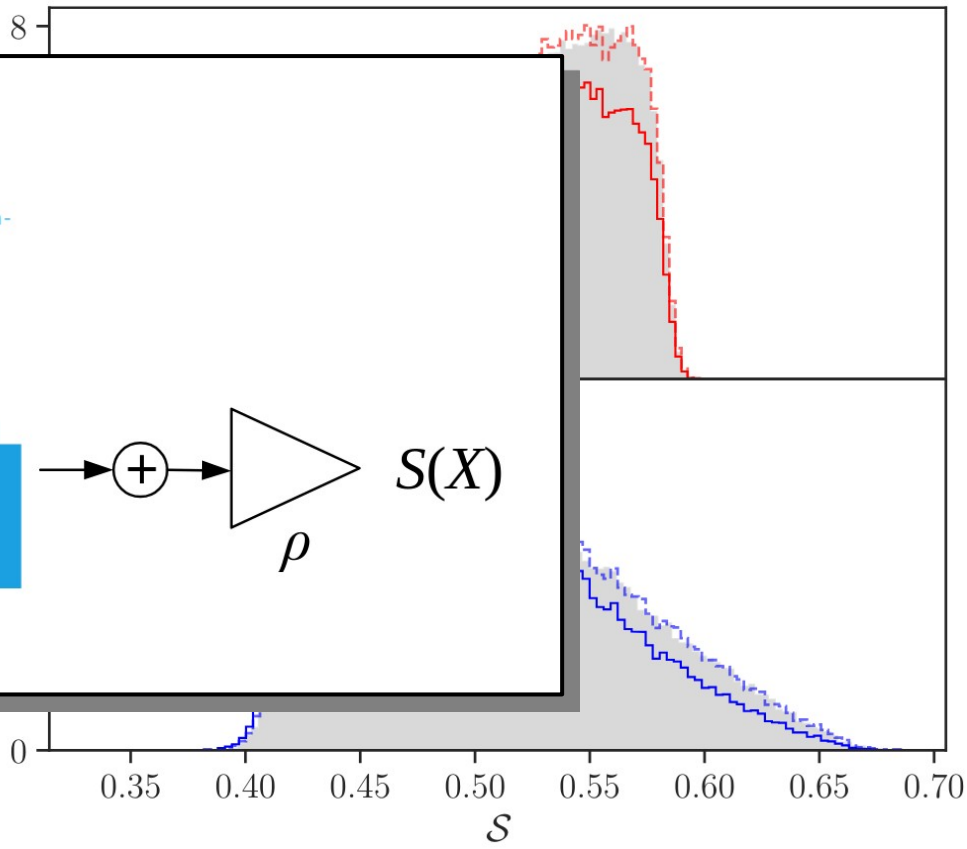


The "score" observable

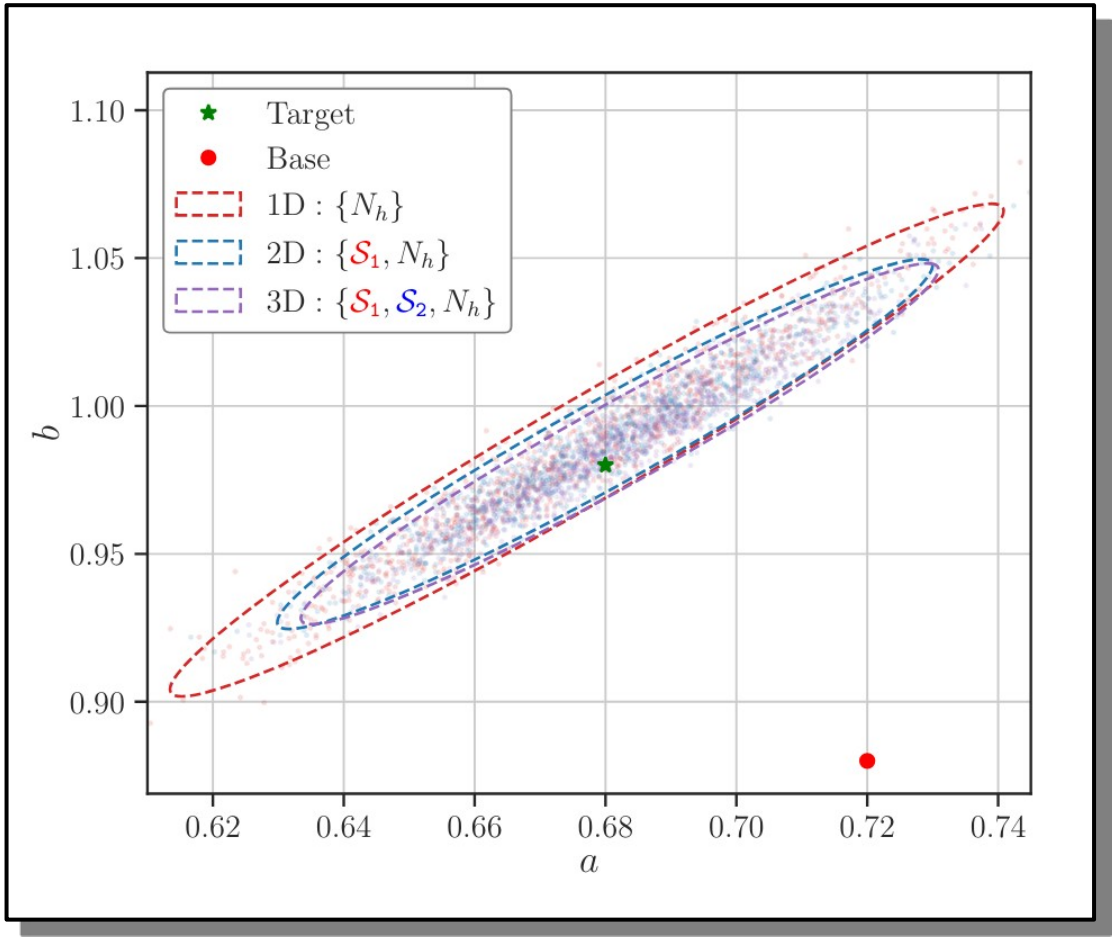
- Train a deepsets classifier to distinguis
  experimer
  - Takes ful
    input

Use tra
(score) as an observable





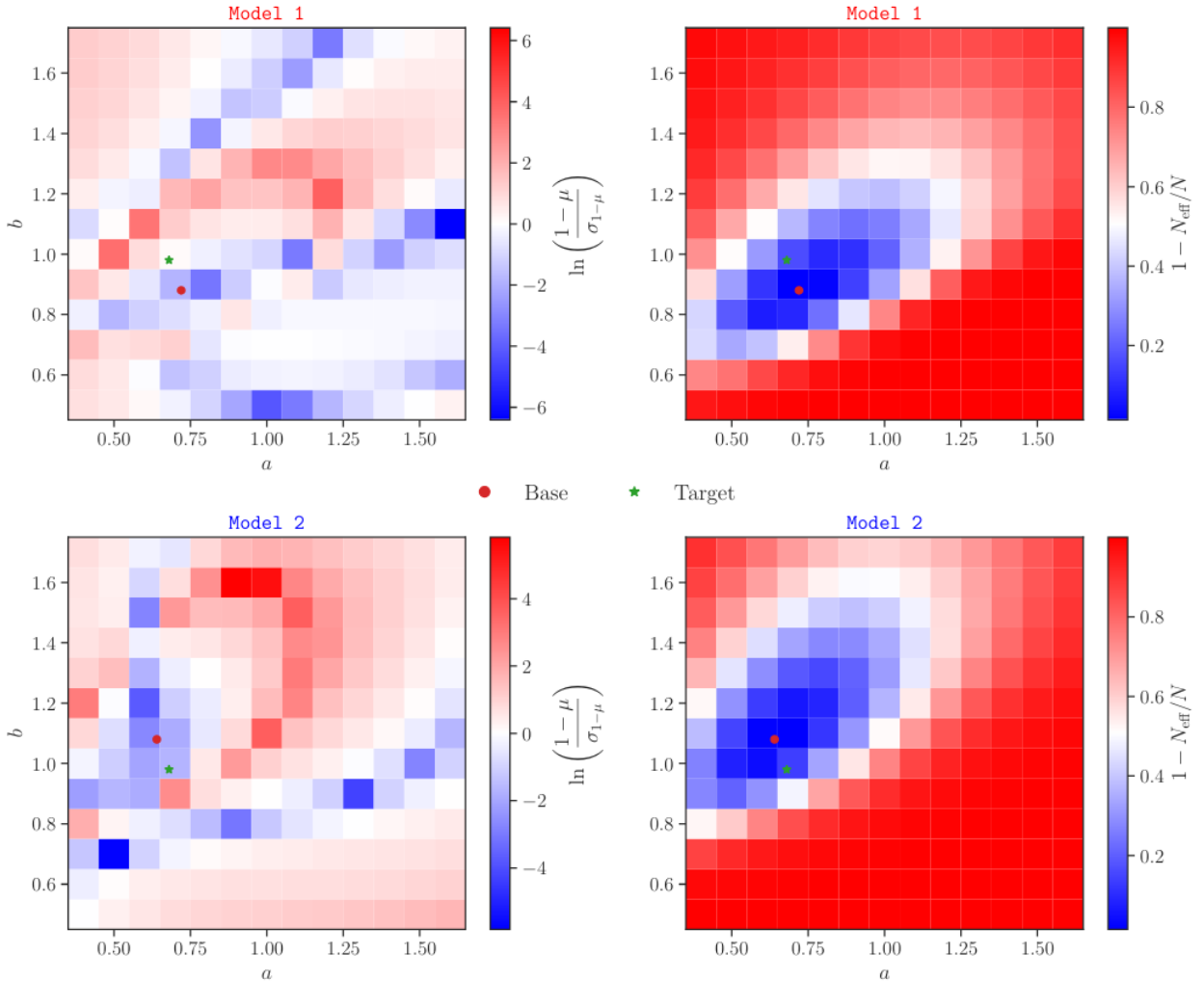
Classifier score with full event info improves tuning convergence

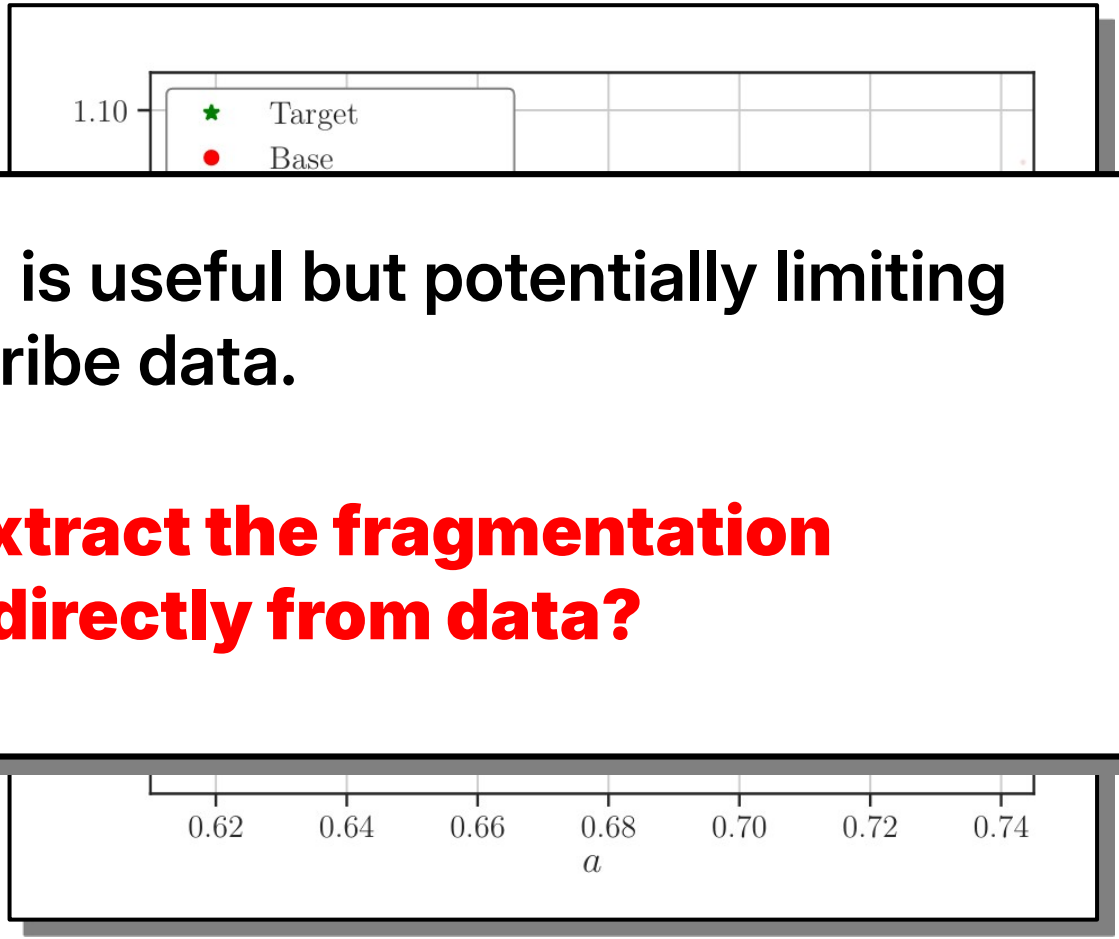


No free lunch

Statistical power drops off quickly as you move away from the base parameterization

$$\mu \equiv \sum_{i=1}^{N} \frac{w_i}{N}, \qquad N_{\text{eff}} = \frac{\left(\sum_{i=1}^{N} w_i\right)^2}{\sum_{i=1}^{N} w_i^2}$$

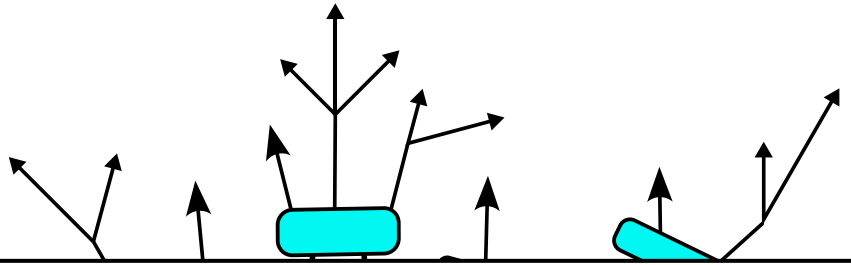




Cla
inf
co

Analytic form of f(z) is useful but potentially limiting when trying to describe data.

Can we extract the fragmentation function directly from data?

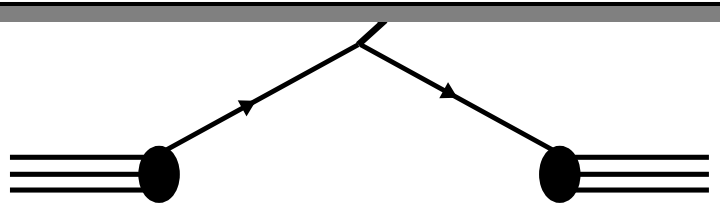


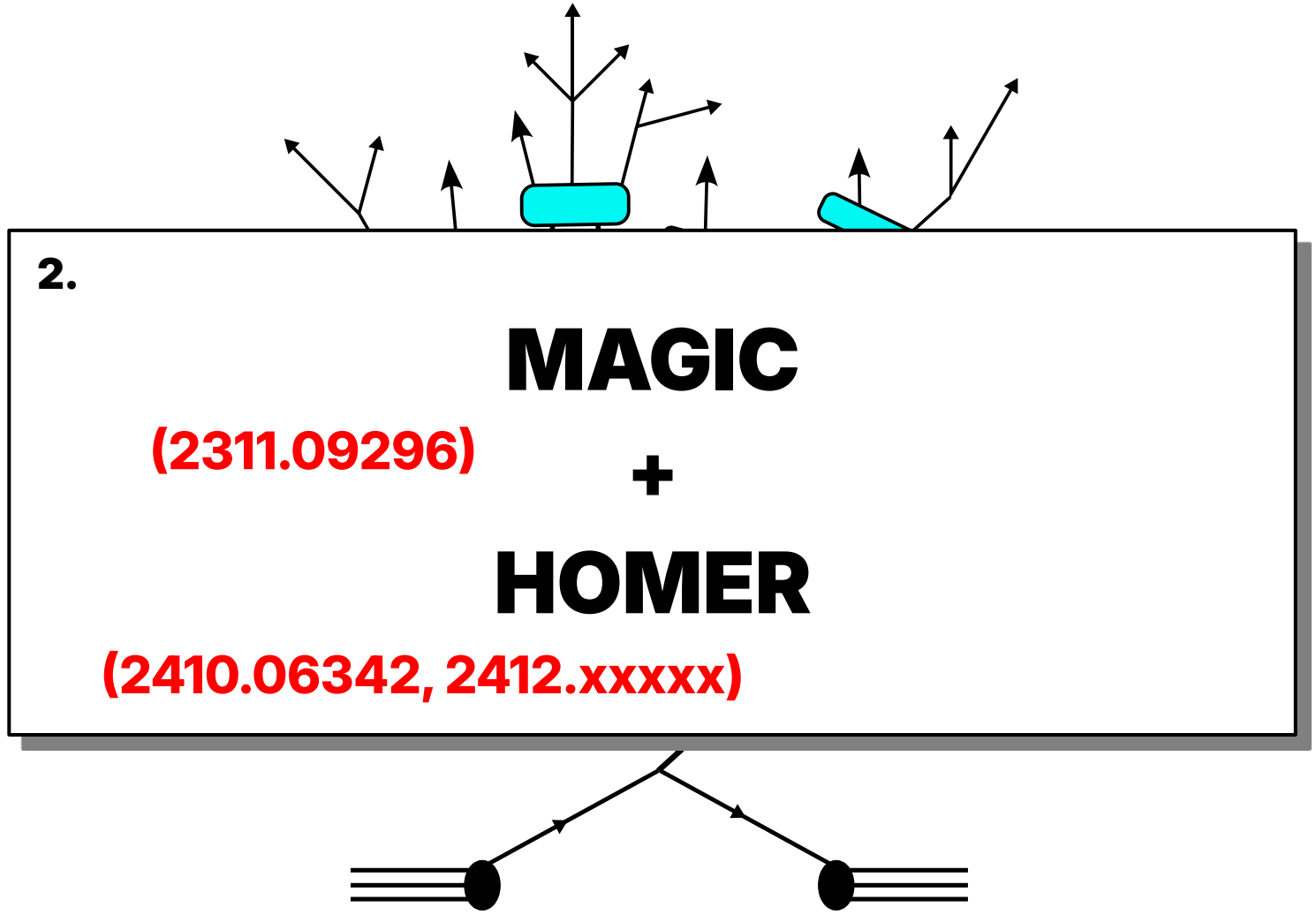
2. Microscopic Alterations Generated from IR Collections

(2311.09296) +

Histories and Observables for Monte Carlo Event Reweighting

(2410.06342, 2412.xxxxx)




2.
MAGIC
(2311.09296)
+
HOMER
(2410.06342, 2412.xxxxx)

Two methods

MAGIC

- "Top-down"
- Relies on the construction of an explicit likelihood (normalizing flow)
- Must be embedded into the simulation pipeline

HOMER

- "Bottom-up"
- Relies on the reconstruction of a classifier weight (derived from the score) from a product of string-break weights
- No modifications to the simulation pipeline required

MAGIC: Invertible nerual networks (INN)

a.k.a normalizing flow

Invertible Real NVP transformations:

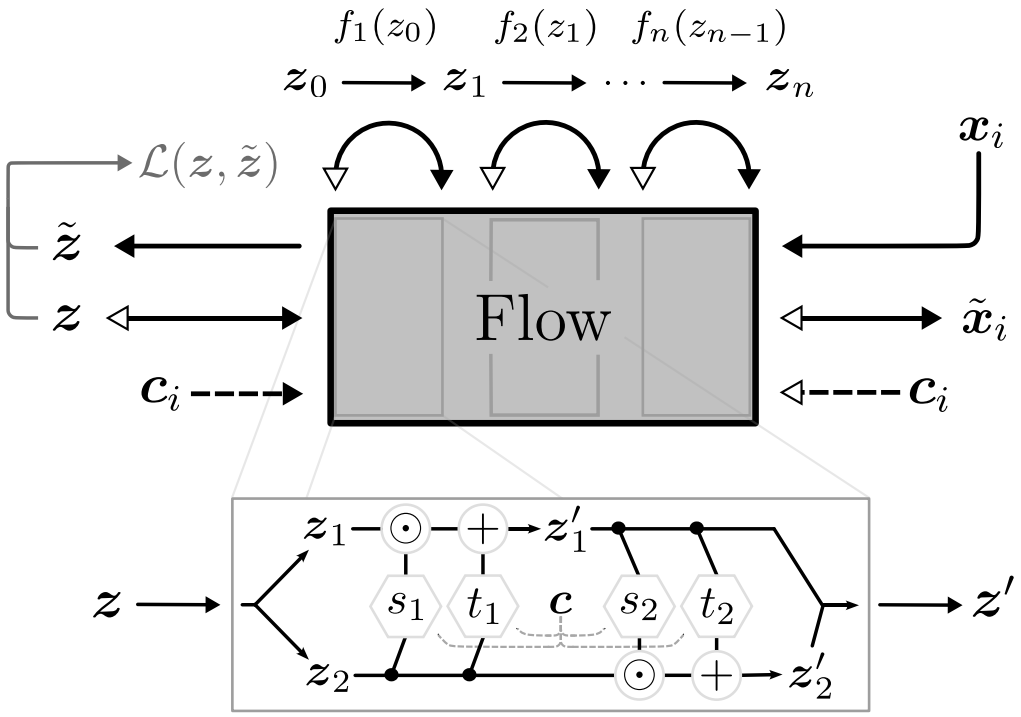
$$z'_1 = z_1 \odot \exp\left(s_1\left(z_2\right)\right) + t_1(z_2),$$
$$z'_2 = z_2 \odot \exp\left(s_2\left(z'_1\right)\right) + t_2(z'_1),$$

Scale transform

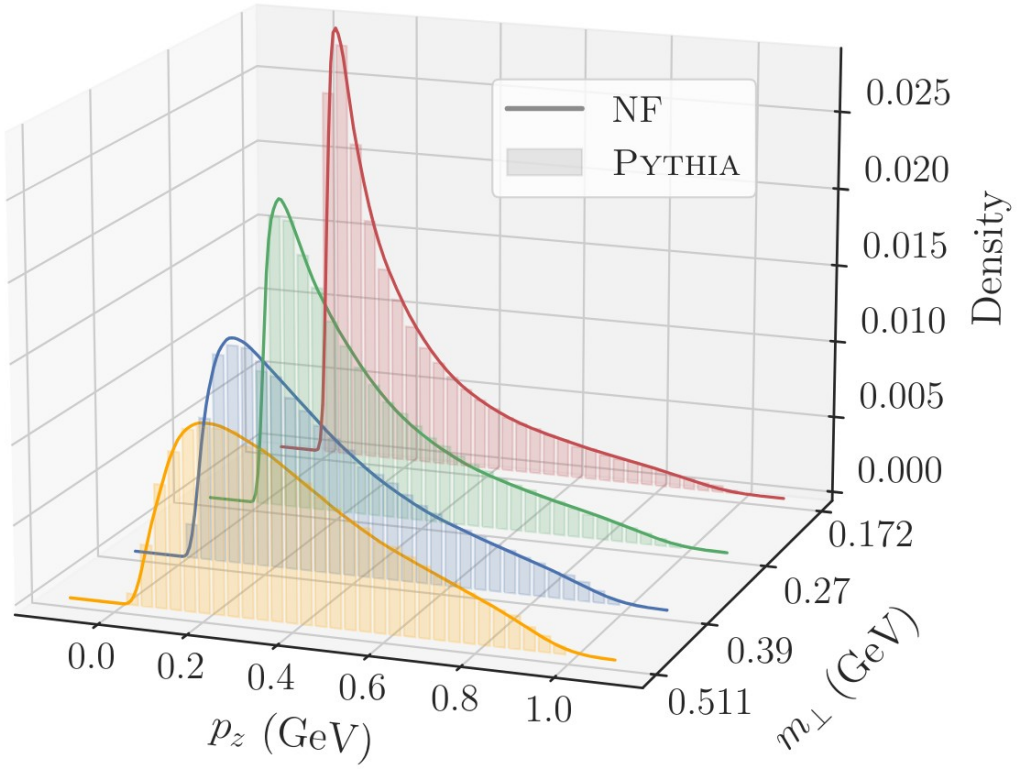
Translation transform

Inverse:

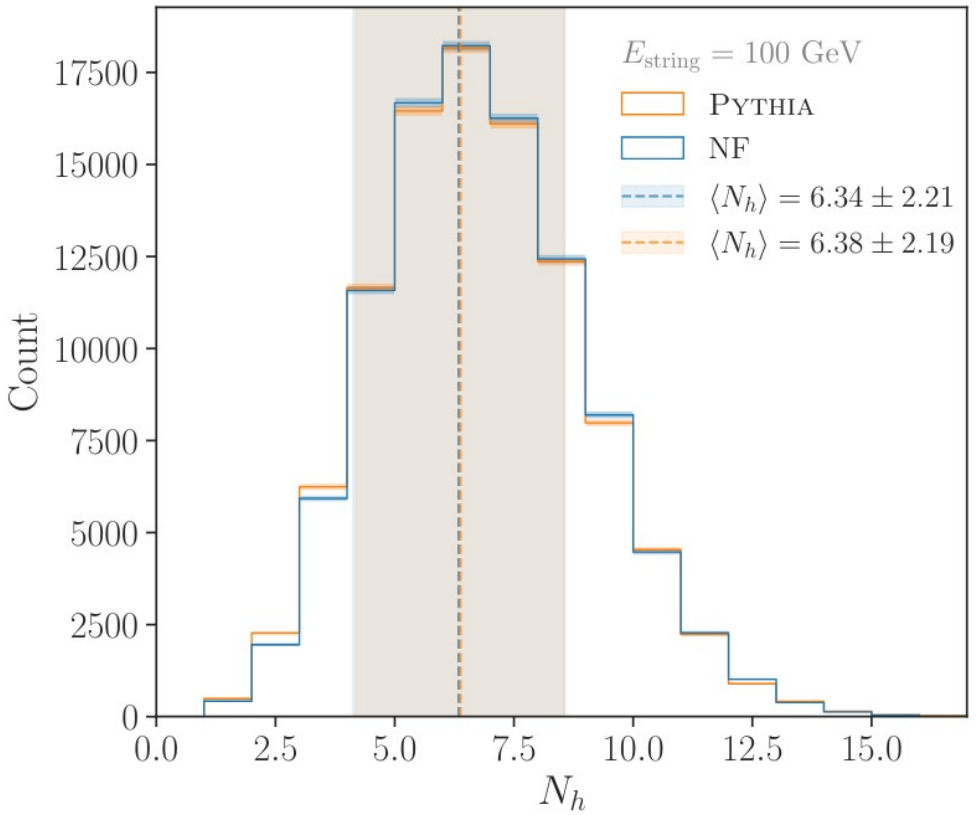
$$z_2 = \left(z'_2 - t\left(z'_1\right)\right) \odot \exp\left(-s\left(z'_1\right)\right),$$
$$z_1 = \left(z'_1 - t\left(z_2\right)\right) \odot \exp\left(-s\left(z_2\right)\right),$$

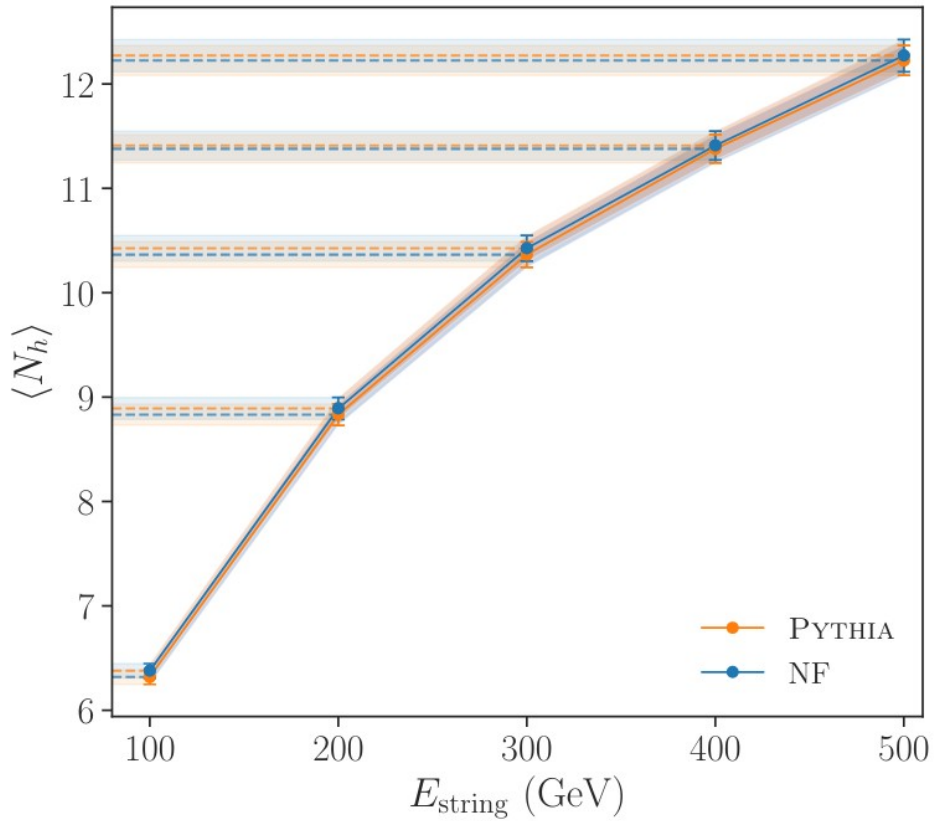


MAGIC: Base model



MAGIC: Base model





MAGIC: $q\bar{q}$

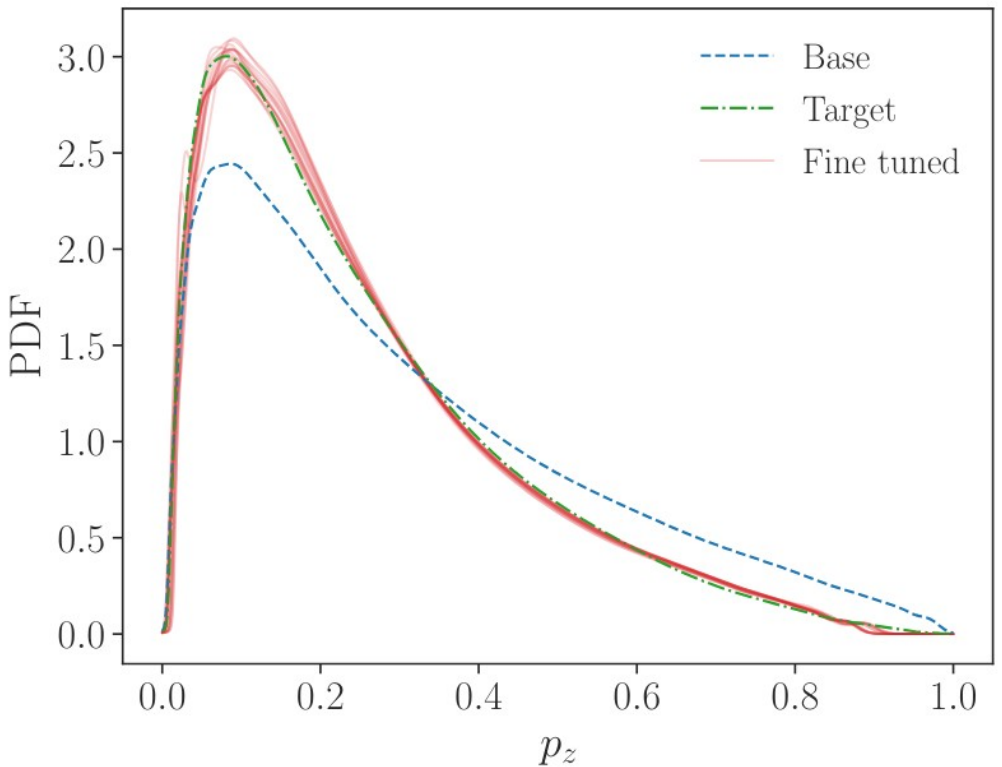
Use hadron multiplicity N as global event-level observable

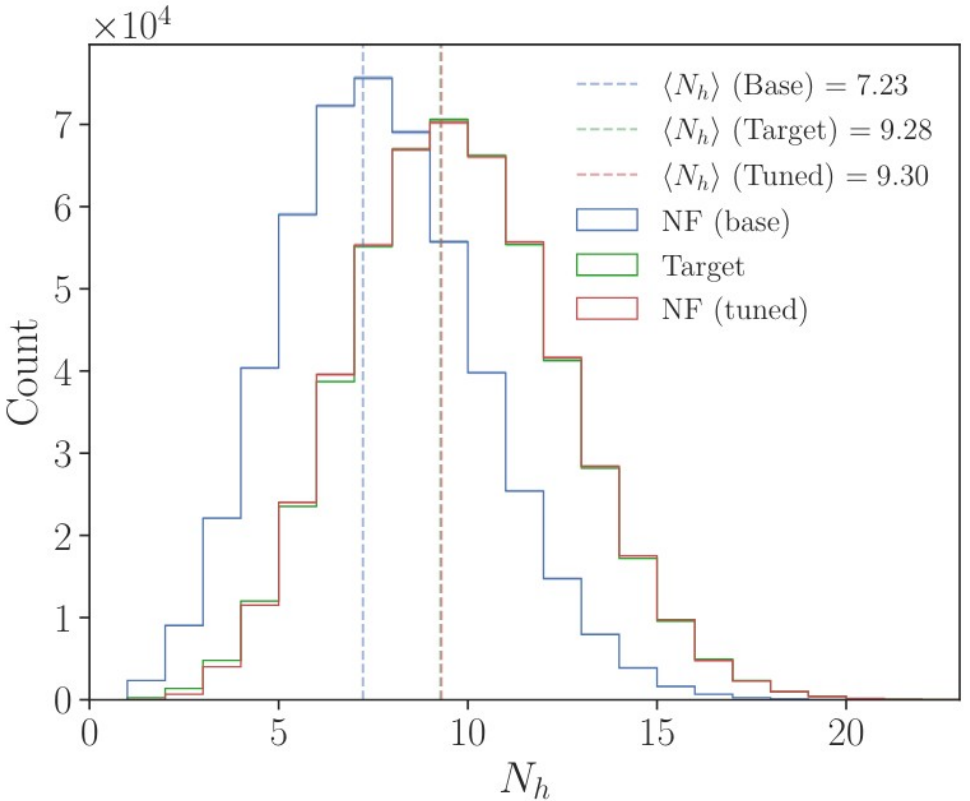
$$\boldsymbol{X} = \begin{pmatrix} \left[\{p_z^{h_1}, p_T^{h_1}\}, \{p_z^{h_2}, p_T^{h_2}\}, \{p_z^{h_3}, p_T^{h_3}\}\right]_1 \\ \left[\{p_z^{h_1}, p_T^{h_1}\}, \{p_z^{h_2}, p_T^{h_2}\}, \{p_z^{h_3}, p_T^{h_3}\}, \{p_z^{h_4}, p_T^{h_4}\}, \{p_z^{h_5}, p_T^{h_5}\}\right]_2 \\ \vdots \\ \left[\{p_z^{h_1}, p_T^{h_1}\}, \{p_z^{h_2}, p_T^{h_2}\}\right]_n \end{pmatrix}, \quad \boldsymbol{Y}^{\text{sim}} = \begin{pmatrix} N_1 = 3 \\ N_2 = 5 \\ \vdots \\ N_n = 2 \end{pmatrix}, \quad \boldsymbol{Y}^{\text{exp}} = \begin{pmatrix} N_1 = 7 \\ N_2 = 3 \\ \vdots \\ N_n = 2 \end{pmatrix}$$

Event weights: $\boldsymbol{w} = \begin{pmatrix} \prod_{i=1}^{N_1} w_i \\ \prod_{j=1}^{N_2} w_j \\ \vdots \\ \prod_{k=1}^{N_n} w_k \end{pmatrix}$ where $w_i = \dfrac{p_X^{F'}(p_z^{h_i}, p_T^{h_i})}{p_X^{F}(p_z^{h_i}, p_T^{h_i})}$,

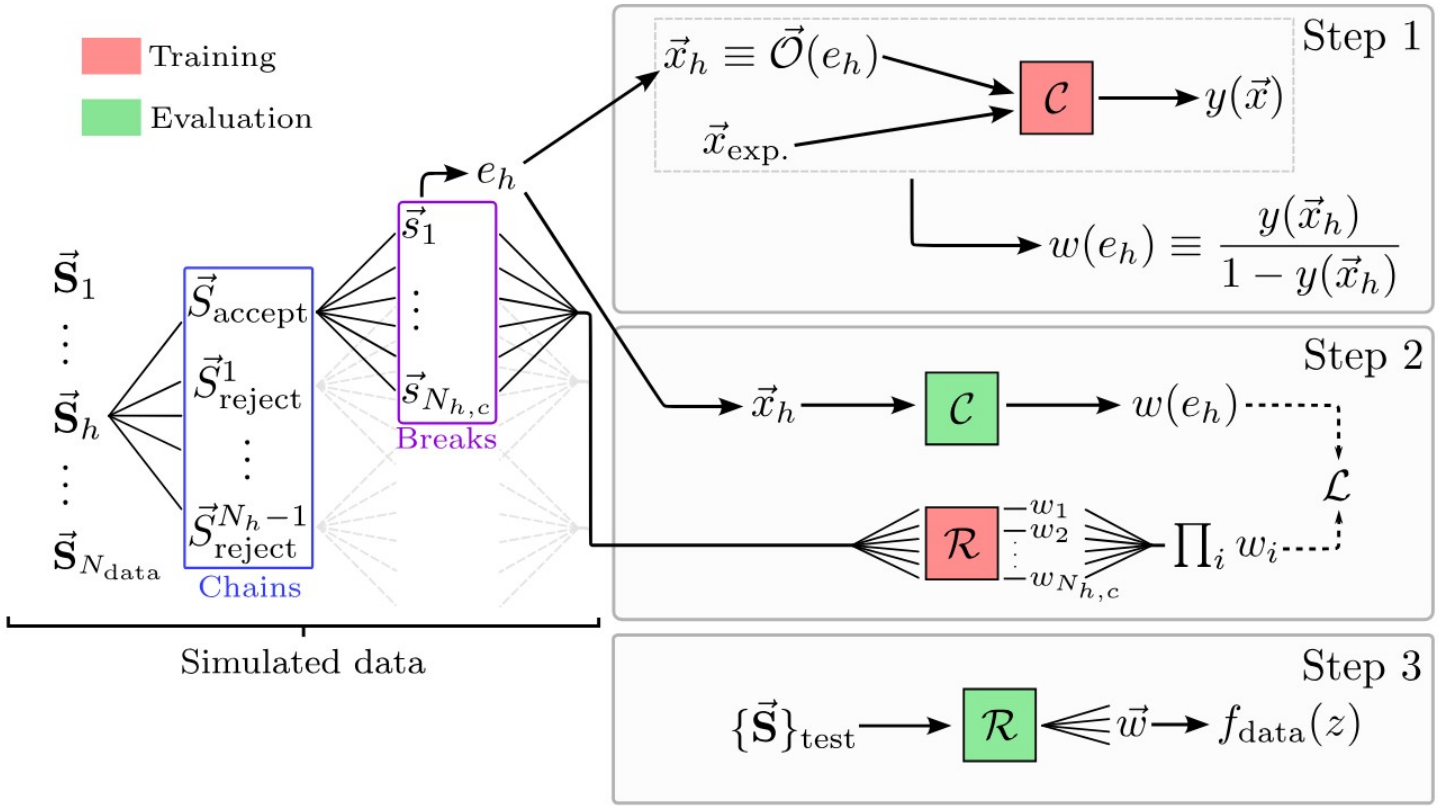
Loss (Earth mover's distance): $\mathcal{L}_{\text{EMD}}(\boldsymbol{Y}^{\text{sim}}, \boldsymbol{w}, \boldsymbol{Y}^{\text{exp}})$

MAGIC: $q\bar{q}$

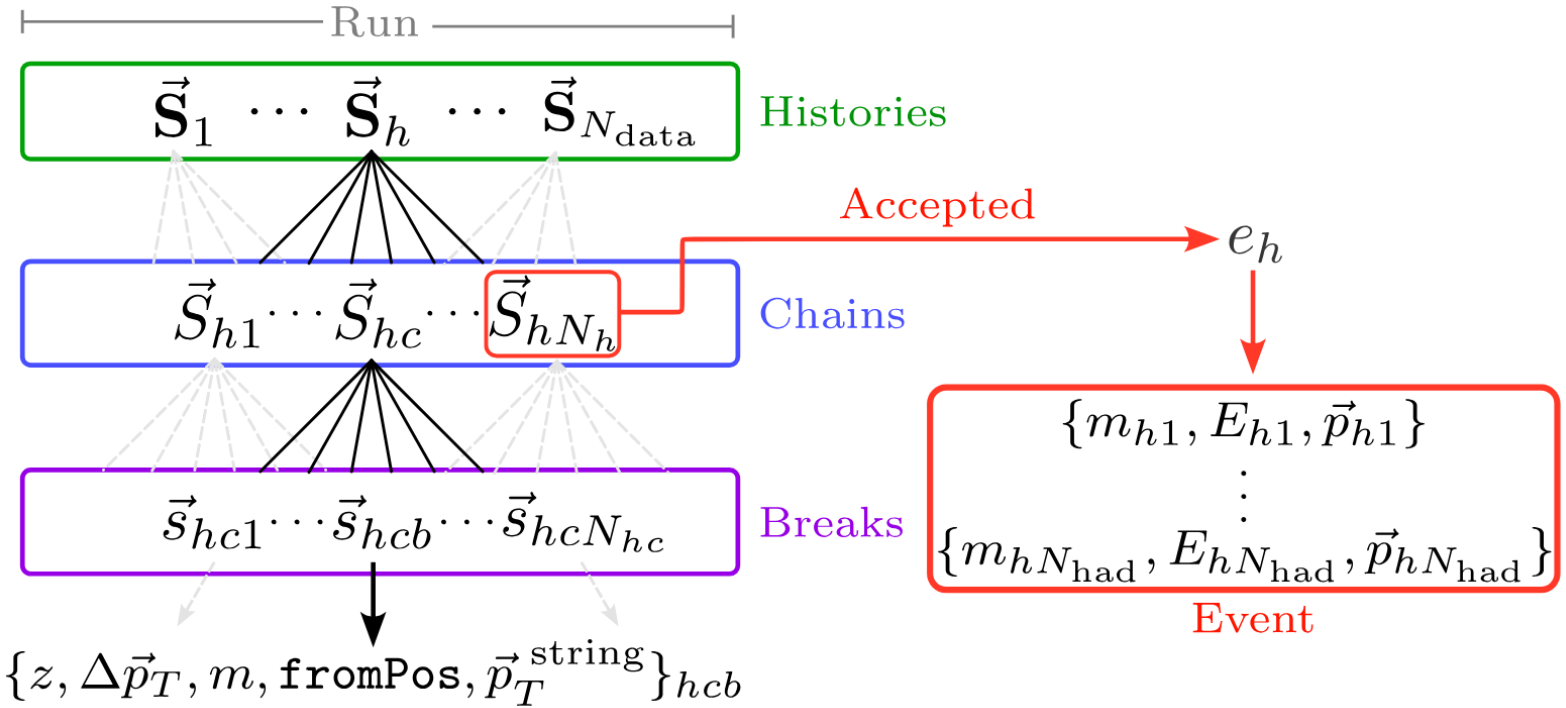

Base
Target
Fine tuned
PDF
p_z
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0


$\times 10^4$
$\langle N_h \rangle$ (Base) = 7.23
$\langle N_h \rangle$ (Target) = 9.28
$\langle N_h \rangle$ (Tuned) = 9.30
NF (base)
Target
NF (tuned)
Count
N_h
0
1
2
3
4
5
6
7
0
5
10
15
20

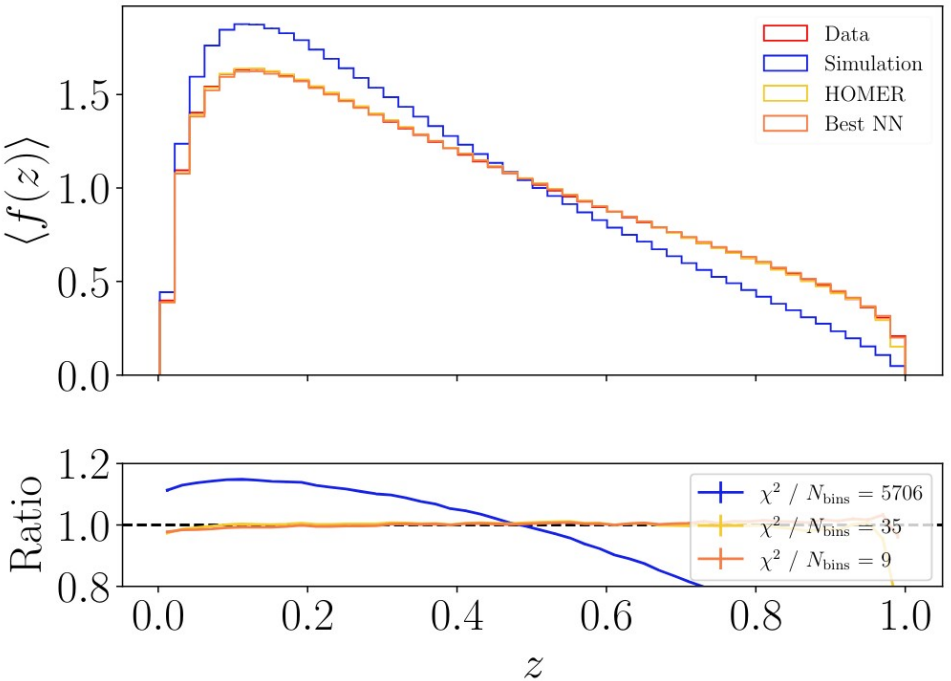
HOMER

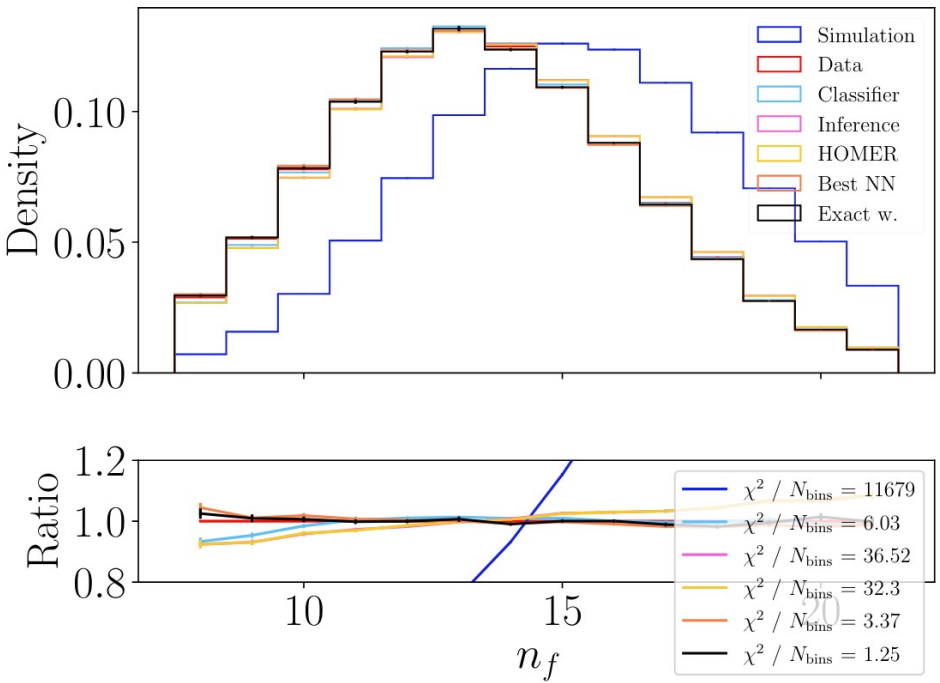

Training
Evaluation
$\vec{\mathbf{S}}_1$
$\vec{\mathbf{S}}_h$
$\vec{\mathbf{S}}_{N_{\text{data}}}$
$\vec{S}_{\text{accept}}$
$\vec{S}^1_{\text{reject}}$
$\vec{S}^{N_h-1}_{\text{reject}}$
Chains
e_h
$\vec{s}_1$
$\vec{s}_{N_{h,c}}$
Breaks
Simulated data
Step 1
$\vec{x}_h \equiv \vec{\mathcal{O}}(e_h)$
$\vec{x}_{\text{exp.}}$
$\mathcal{C}$
$y(\vec{x})$
$w(e_h) \equiv \frac{y(\vec{x}_h)}{1-y(\vec{x}_h)}$
Step 2
$\vec{x}_h$
$\mathcal{C}$
$w(e_h)$
$\mathcal{L}$
$\mathcal{R}$
w_1
w_2
$w_{N_{h,c}}$
$\prod_i w_i$
Step 3
$\{\vec{\mathbf{S}}\}_{\text{test}}$
$\mathcal{R}$
$\vec{w}$
$f_{\text{data}}(z)$

HOMER: $q\bar{q}$

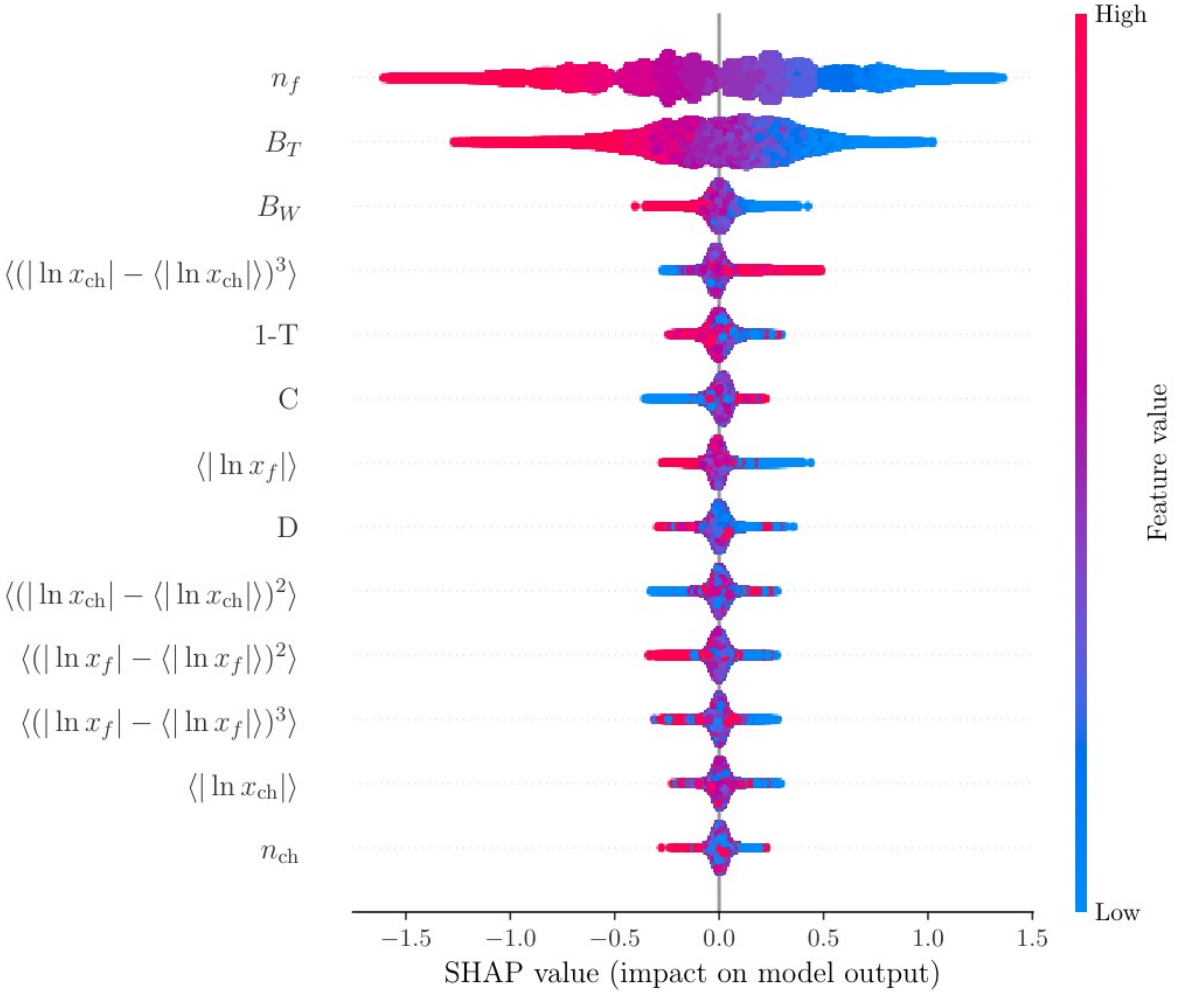

Run
$\vec{\mathbf{S}}_1 \cdots \vec{\mathbf{S}}_h \cdots \vec{\mathbf{S}}_{N_{\text{data}}}$
Histories
Accepted
e_h
$\vec{S}_{h1} \cdots \vec{S}_{hc} \cdots \vec{S}_{hN_h}$
Chains
$\{m_{h1}, E_{h1}, \vec{p}_{h1}\}$
$\{m_{hN_{\text{had}}}, E_{hN_{\text{had}}}, \vec{p}_{hN_{\text{had}}}\}$
Event
$\vec{s}_{hc1} \cdots \vec{s}_{hcb} \cdots \vec{s}_{hcN_{hc}}$
Breaks
$\{z, \Delta\vec{p}_T, m, \texttt{fromPos}, \vec{p}_T^{\,\text{string}}\}_{hcb}$

HOMER: $q\bar{q}$


Data
Simulation
HOMER
Best NN
$\langle f(z) \rangle$
Ratio
$\chi^2 / N_{\text{bins}} = 5706$
$\chi^2 / N_{\text{bins}} = 35$
$\chi^2 / N_{\text{bins}} = 9$
z


Simulation
Data
Classifier
Inference
HOMER
Best NN
Exact w.
Density
Ratio
$\chi^2 / N_{\text{bins}} = 11679$
$\chi^2 / N_{\text{bins}} = 6.03$
$\chi^2 / N_{\text{bins}} = 36.52$
$\chi^2 / N_{\text{bins}} = 32.3$
$\chi^2 / N_{\text{bins}} = 3.37$
$\chi^2 / N_{\text{bins}} = 1.25$
n_f

SHAP values


n_f
B_T
B_W
$\langle(|\ln x_{\rm ch}| - \langle|\ln x_{\rm ch}|\rangle)^3\rangle$
1-T
C
$\langle|\ln x_f|\rangle$
D
$\langle(|\ln x_{\rm ch}| - \langle|\ln x_{\rm ch}|\rangle)^2\rangle$
$\langle(|\ln x_f| - \langle|\ln x_f|\rangle)^2\rangle$
$\langle(|\ln x_f| - \langle|\ln x_f|\rangle)^3\rangle$
$\langle|\ln x_{\rm ch}|\rangle$
$n_{\rm ch}$
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
SHAP value (impact on model output)
High
Low
Feature value

Conclusions

- RSA: flexible and scalable approach to parameter estimation
  - Fully differentiable simulations
  - Uncertainty quantification, flavor tune
- Data-driven fragmentation function extraction:
  - HOMER and MAGIC offer as complementary methods
    - HOMER + gluons (2412.xxxxx), flavor, tackle real data soon?
      - Better observables needed
    - HOMER x MAGIC
    - Better likelihood models? (MAGIC$^+$)
  - Topology and simulation informed weights

https://uchep.gitlab.io/mlhad-docs/

Conclusions

- RSA: flexible and scalable approach to parameter estimation
  - Fully differentiable simulations
  - Uncertainty quantification, flavor tune
- Data-driven fragmentation function extraction:
  - HOMER and MAGIC offer as complementary methods
    - HOMER + gluons (2412.xxxxx), flavor, tackle real data soon?
      - Better observables needed
    - HOMER x MAGIC
    - Better likelihood models? (MAGIC^{+})
  - Topology and simulation informed weights

MLHAD

https://uchep.gitlab.io/mlhad-docs/

Thank you :)

Back-up

Stringy hadronization: overview

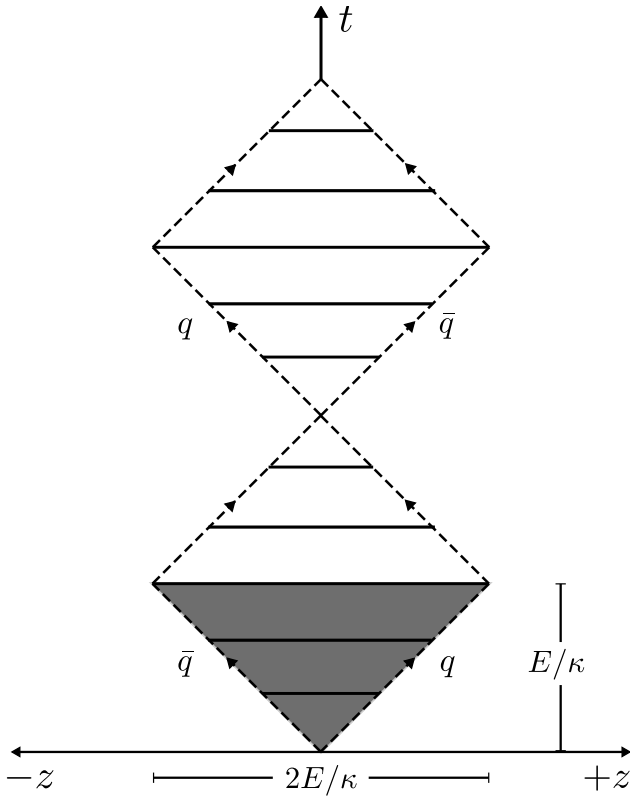
Consider the simplest hadronizing system:

A $q\bar{q}$ pair oriented along the z-axis, with equal and opposite momentum.

Treat this as a semi-classical system with potential

$$V(r) = \kappa r$$

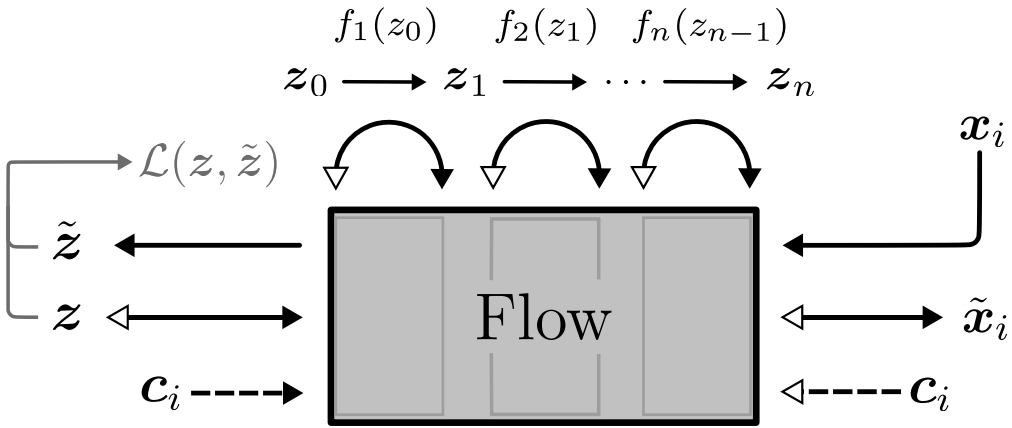
A color flux tube forms between the $q\bar{q}$ pair and in the absence of string breaks the system follows a 'yo-yo' motion



Invertible nerual networks (INN)

a.k.a normalizing flow

Learn a composition of n independent bijective transformations that relate a probability distribution $p_Z(z)$ on latent space Z to the target distribution $p_X(x)$ on target space X.



The probability distribution for the random variable x = f(z) is given by

$$p_X^f(\boldsymbol{x}) = p_Z(\boldsymbol{z})|\det J_f(\boldsymbol{z})|^{-1}$$

$$J_f = \partial f / \partial x$$

For n iterative transformations:

$$p_X^F(\boldsymbol{x}) = p_Z(\boldsymbol{z}_0) \prod_{i=1}^{n} |\det J_{f_i}(\boldsymbol{z}_{i-1})|^{-1}$$

Invertible nerual networks (INN)

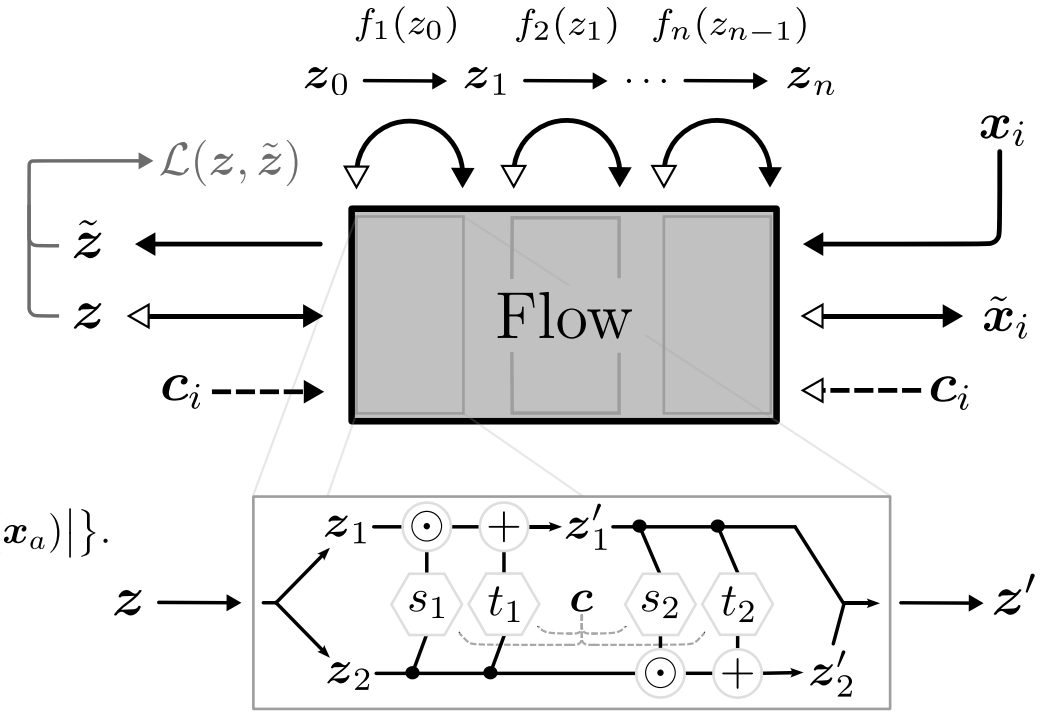
a.k.a normalizing flow

$$p_X^F(\boldsymbol{x}) = p_Z(\boldsymbol{z}_0) \prod_{i=1}^{n} |\det J_{f_i}(\boldsymbol{z}_{i-1})|^{-1}$$

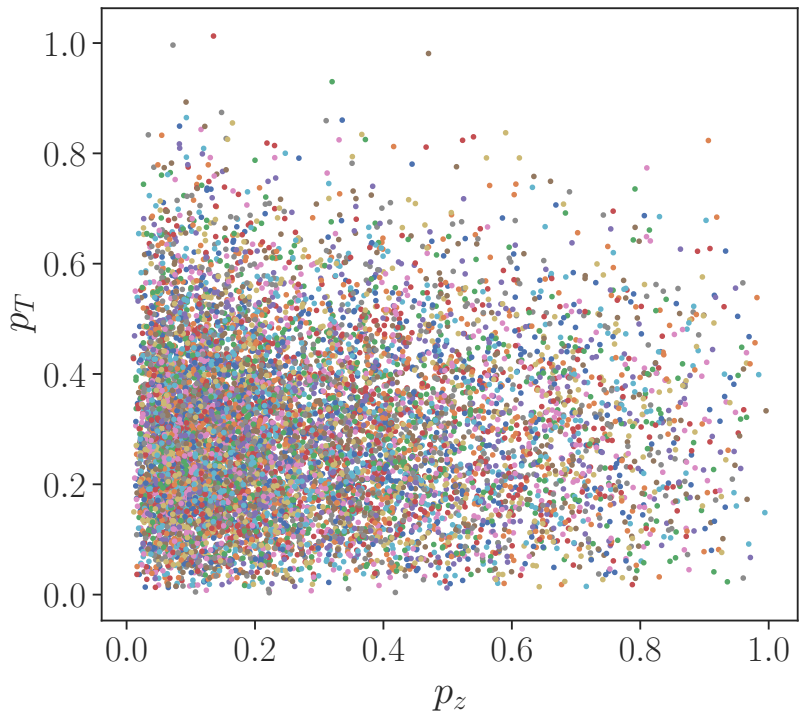
Train with the negative log likelihood:

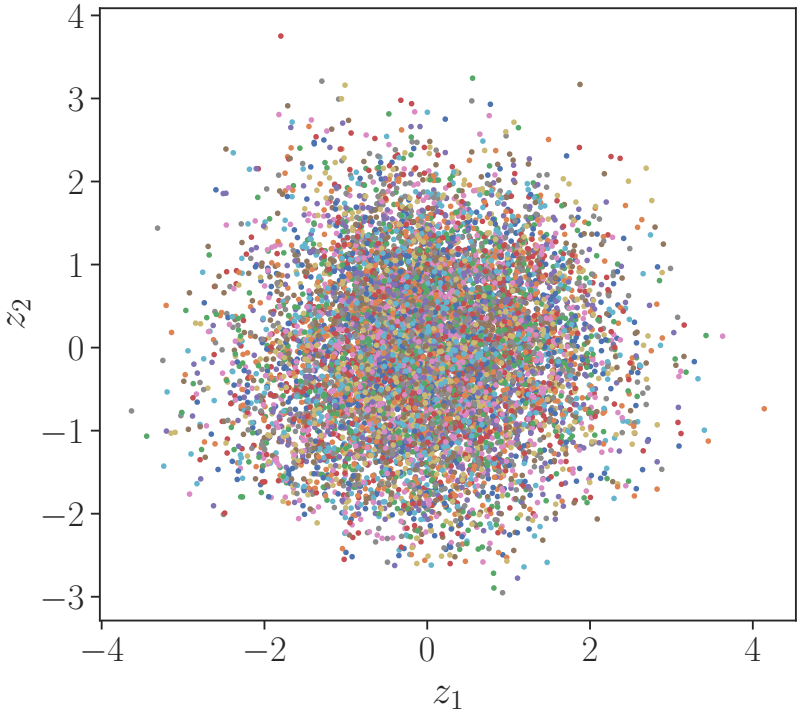
$$\mathcal{L}_{\mathrm{NF}} = -\sum_{a=1}^{N} \log p_X^F(\boldsymbol{x}_a; \boldsymbol{\theta}, \boldsymbol{c}_a)$$

$$= \sum_{a=1}^{N} \Big\{ -\log p_Z\big(F^{-1}(\boldsymbol{x}_a; \boldsymbol{\theta}, \boldsymbol{c}_a)\big) + \log \Big| \det J_{F^{-1}}(\boldsymbol{x}_a) \Big| \Big\}.$$

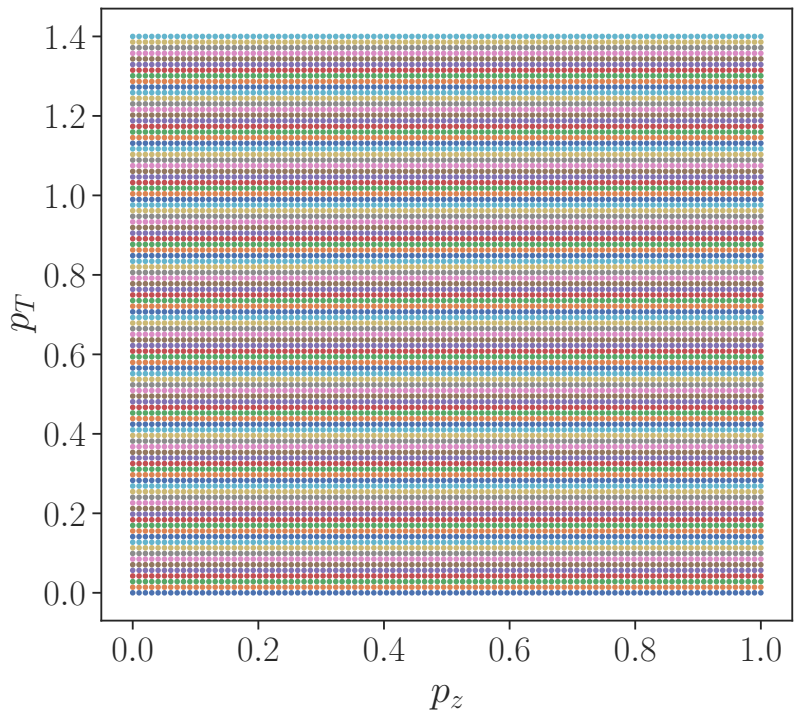


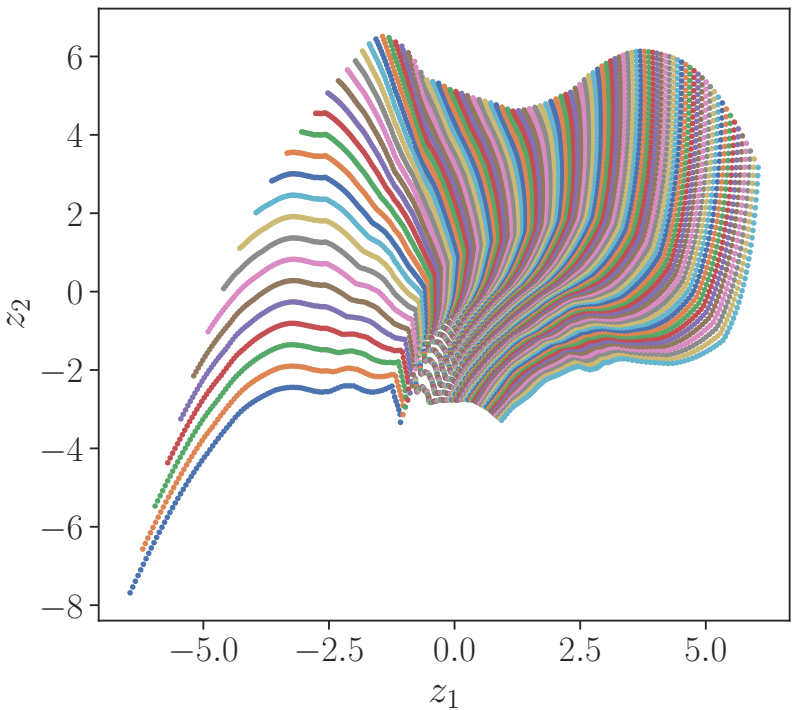
INN learned mapping





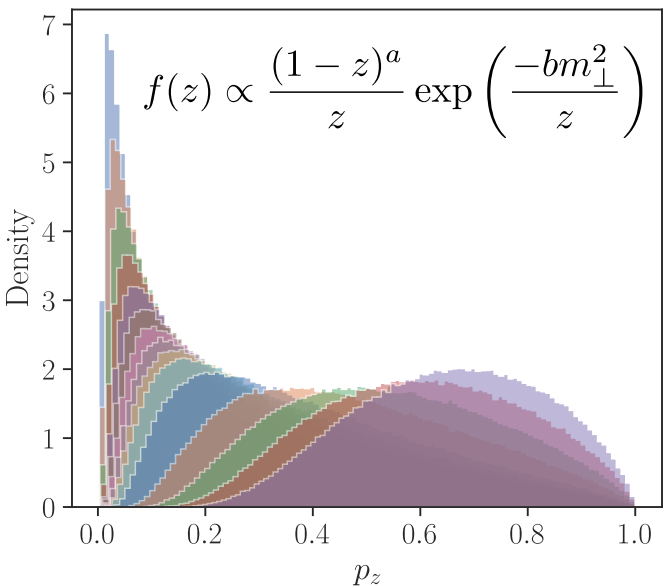
INN learned mapping

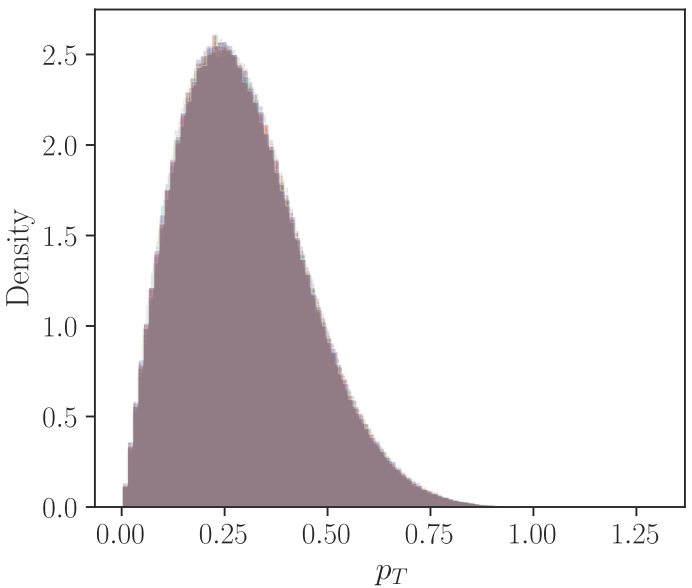




Training data

The implementation of full hadronization event using single emission kinematics requires an independent p_T sampling followed by a p_T-dependent sampling of p_z sampling (due to the dependence of f(z) of transverse mass).





Generate Pythia $q\bar{q}$ → h events, first with no p_T kicks at different values of the hadron mass, record p_z. Generate events again, with kicks turned on, record p_T.