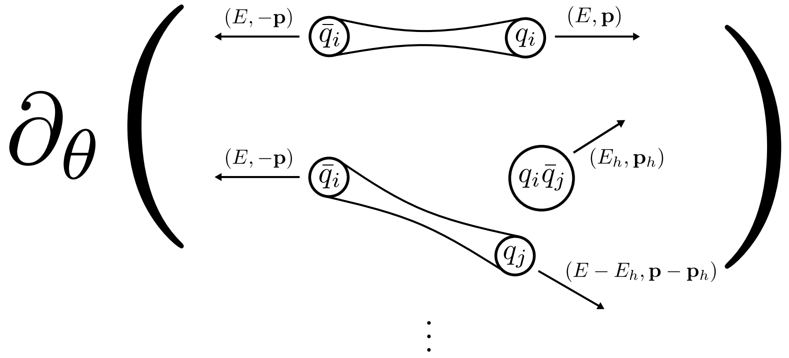
The journey towards differentiable hadronization models

University of Alabama
HEP Seminar
October 17th, 2025

Tony Menzo
Postdoc @ University of Alabama & Fermilab



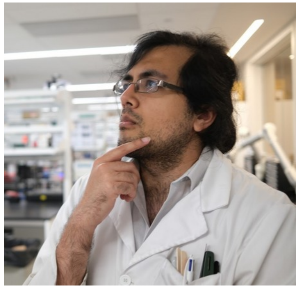
Based upon works in 2203.04983, 2308.13459, 2311.09296, 2410.06342, 2411.02194, 2505.00142, 2503.05667, 2509.03592, 2511.xxxxx

MLHAD

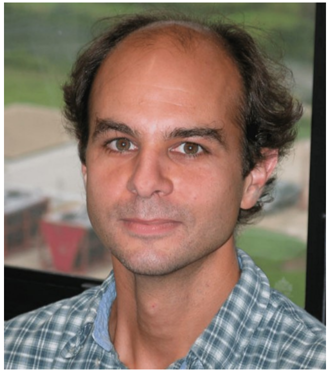

Jure Zupan


Phil Ilten


Manuel Szewc

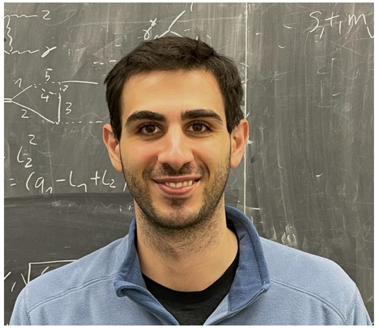

Rikab Gambhir


Christian Bierlich

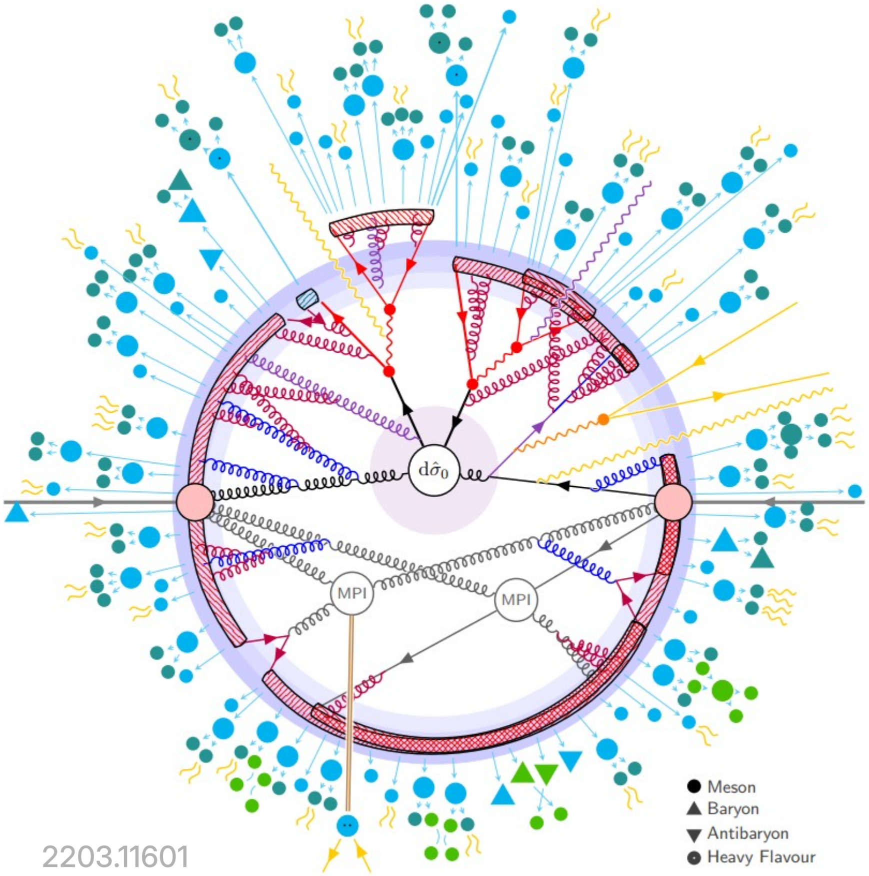

Steve Mrenna


Ahmed Youssef


Michael Wilkinson


Ben Assi

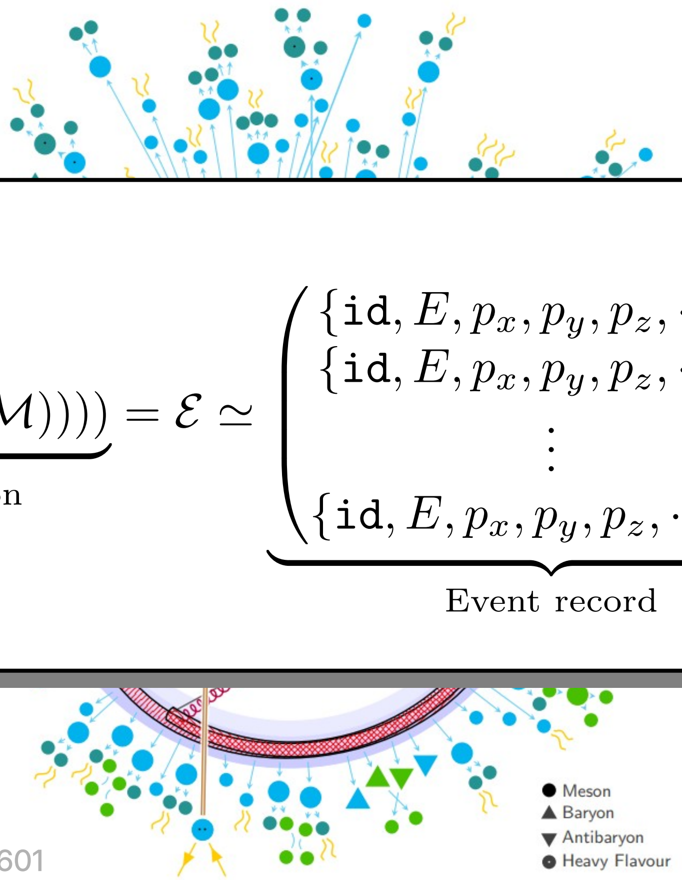
Monte Carlo Event generators



2203.11601

Monte Carlo Event generators

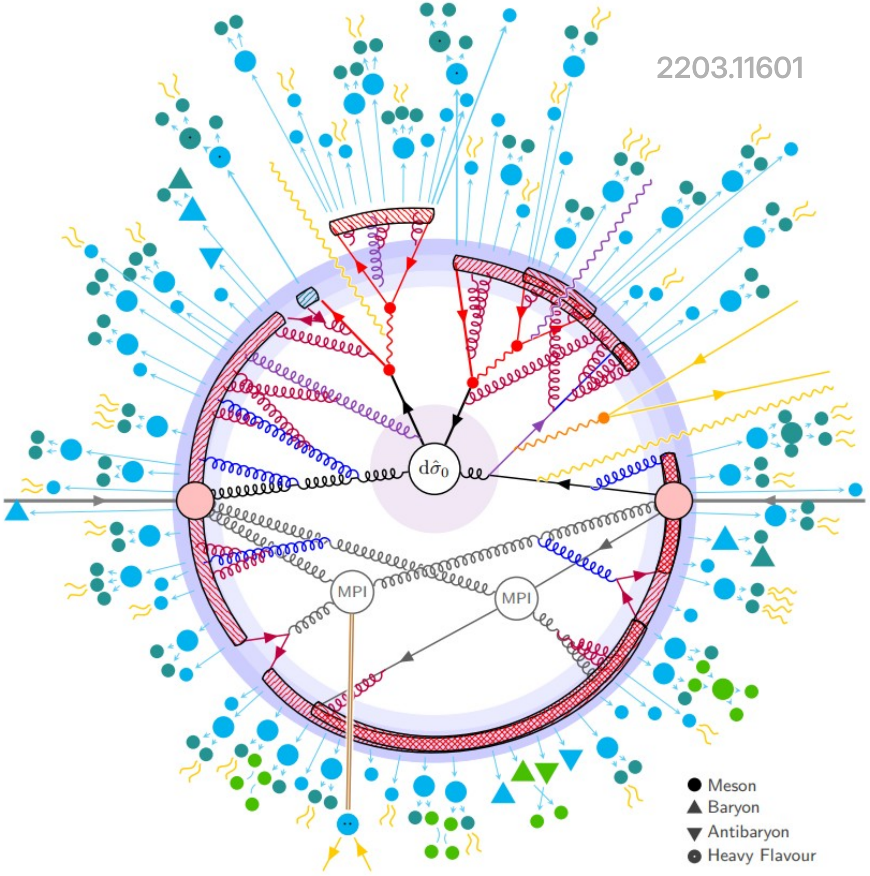
$$\mathcal{G} : \underbrace{\mathcal{S}(\mathcal{D}(\mathcal{H}(\mathcal{P}(\mathcal{M}))))}_{\text{Simulation}} = \mathcal{E} \simeq \underbrace{\begin{pmatrix} \{\texttt{id}, E, p_x, p_y, p_z, \cdots\}_1 \\ \{\texttt{id}, E, p_x, p_y, p_z, \cdots\}_2 \\ \vdots \\ \{\texttt{id}, E, p_x, p_y, p_z, \cdots\}_N \end{pmatrix}}_{\text{Event record}} .$$



2203.11601

Hadronization

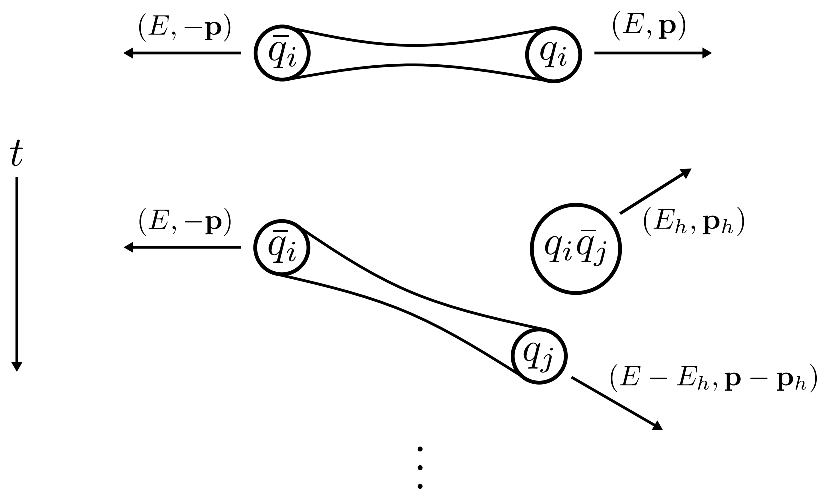
The conversion of a partonic system consisting of quarks, anti-quarks, and gluons ($q\bar{q}, qg\bar{q}, gg, qqq, qq\bar{q}\bar{q}, qqqgg, \ldots$) to a final state consisting of hadrons ($h_1, h_2, \ldots, h_n$).



Phenomenological models

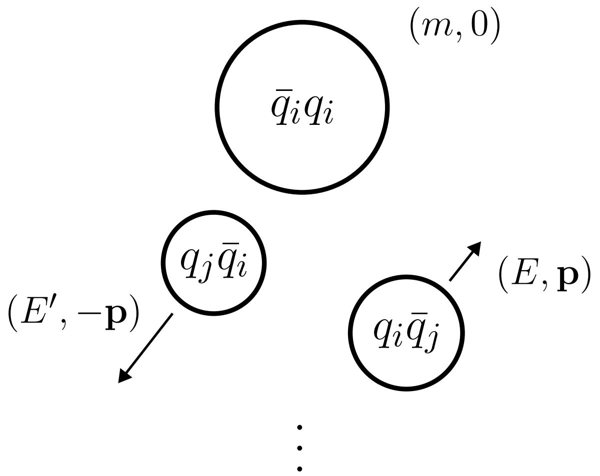
Lund string model

(used in Pythia)



Cluster model

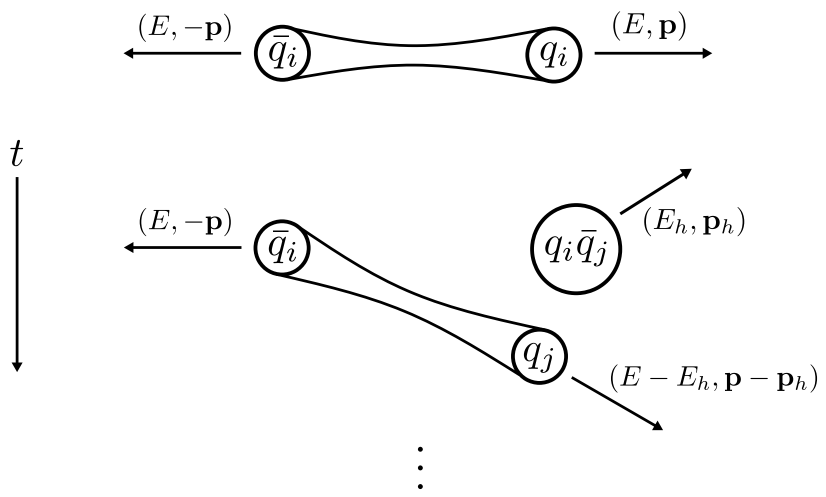
(used in Herwig, Sherpa)



Phenomenological models

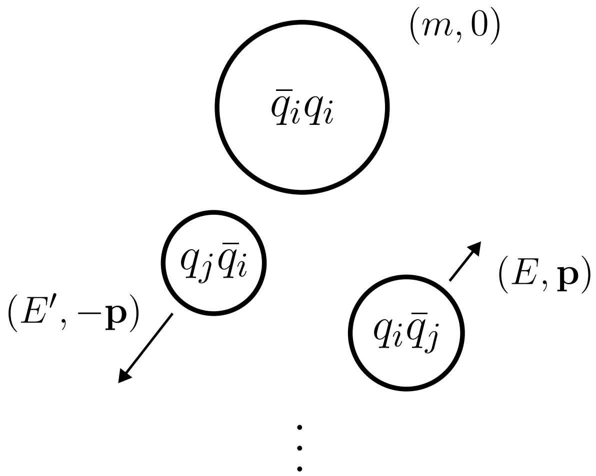
Lund string model

(used in Pythia)



Cluster model

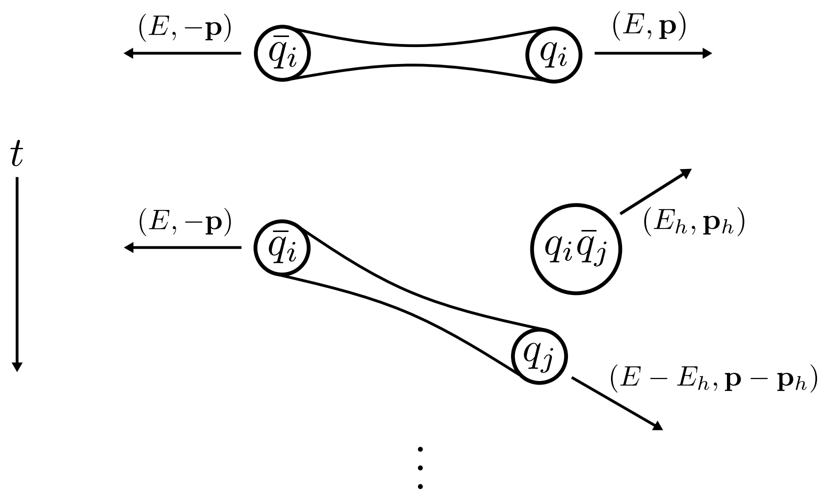
(used in Herwig, Sherpa)



Phenomenological models

Lund string model

(used in Pythia)



Cluster model

(used in Herwig, Sherpa)

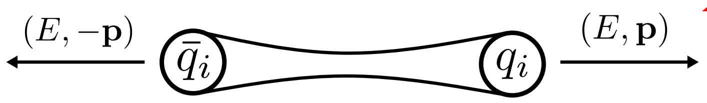
HadML collaboration

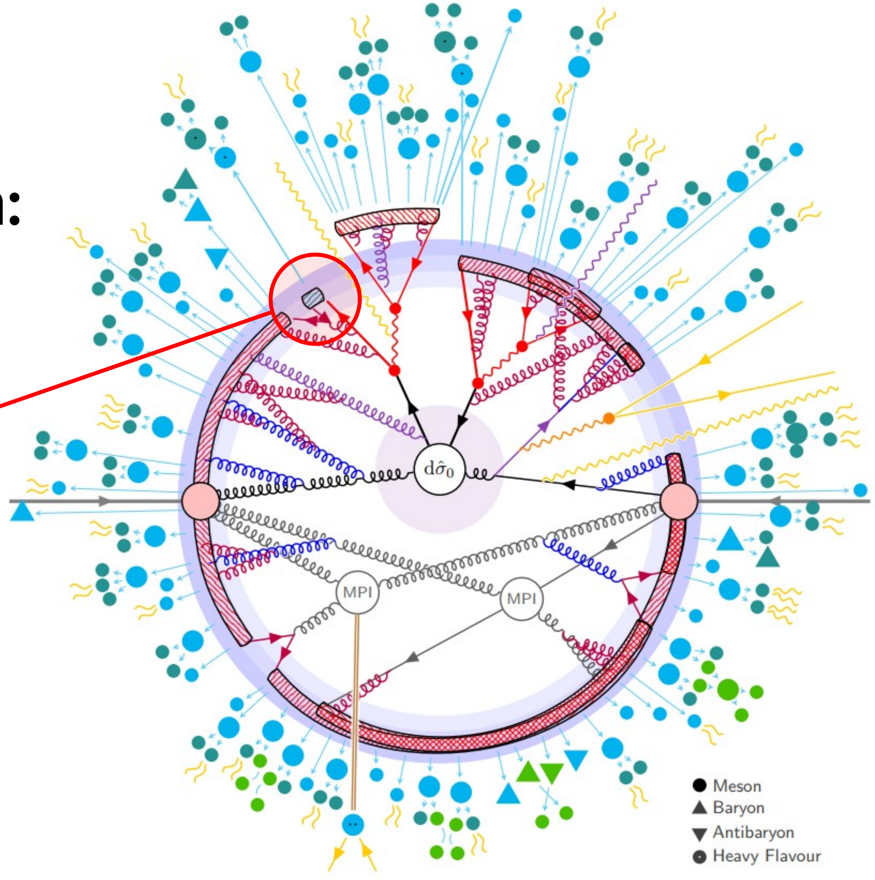
For ML methods applied to the cluster model:

2203.12660, 2305.17169, 2312.08453

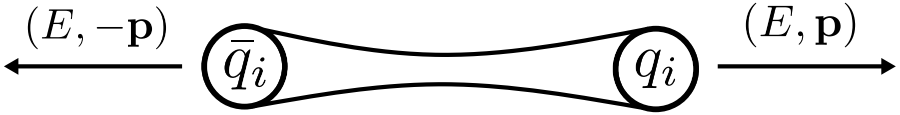
⋮

The "atomic" hadronizing system:
Quark-antiquark ($q\bar{q}$)





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



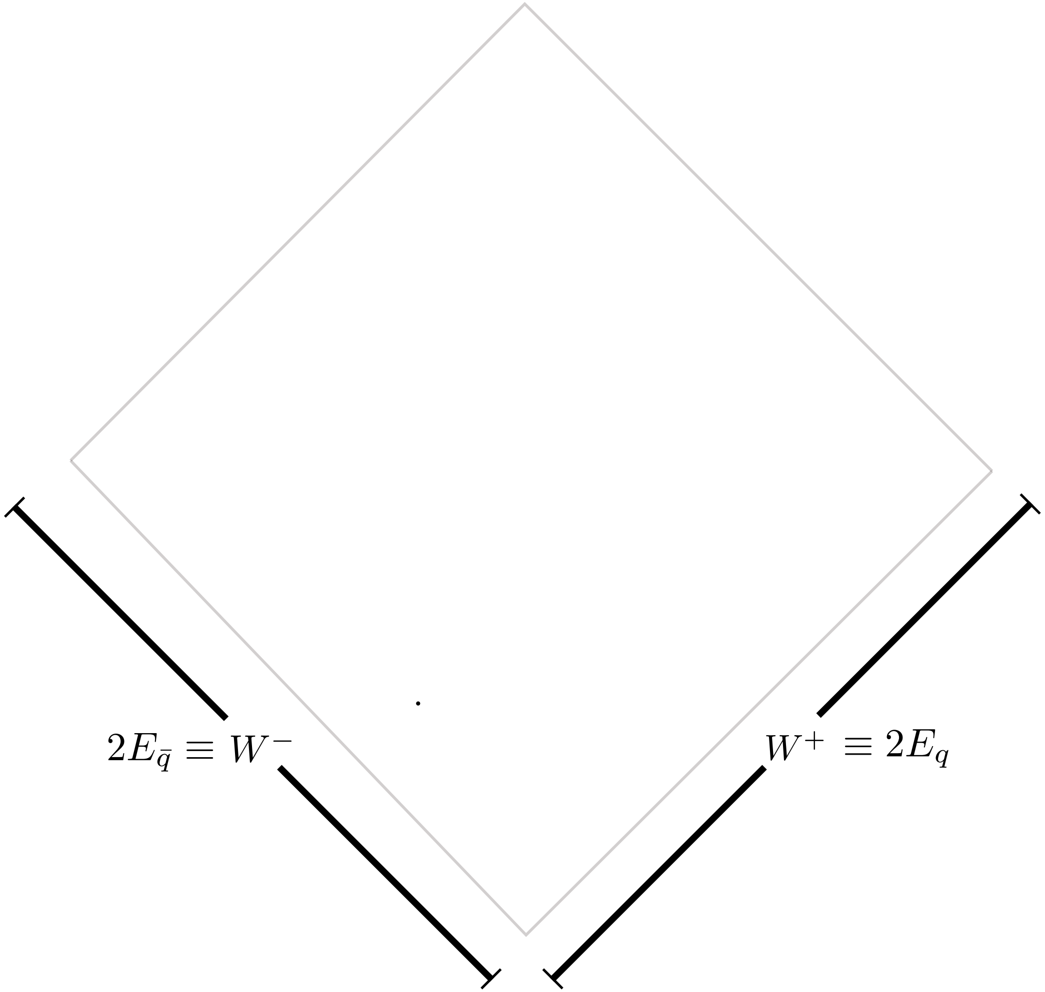
1) Randomly select one of the string ends
2) Sample string break flavor
3) Sample transverse momentum of new quarks

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

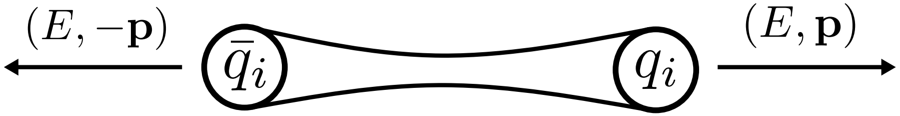
4) Sample longitudinal momentum fraction of new hadron

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

5) Repeat steps 1-4



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



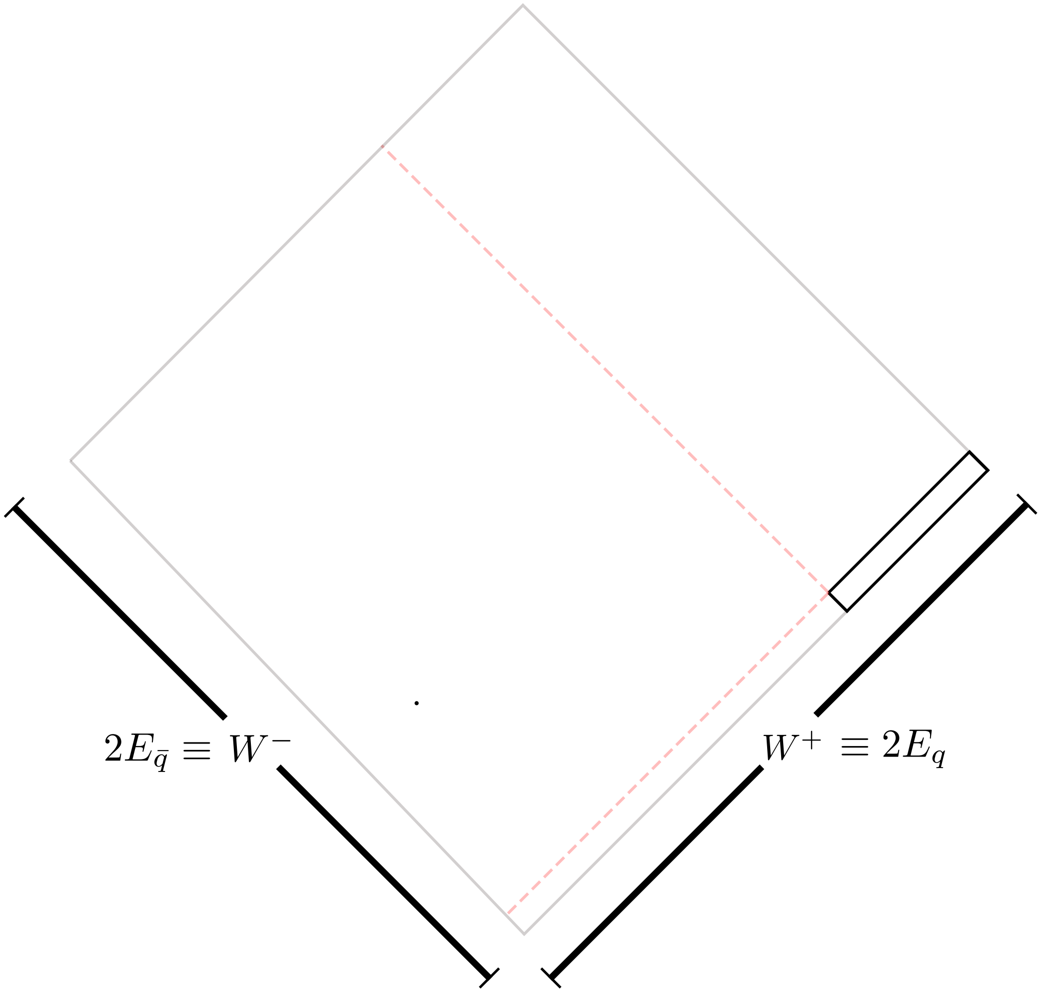
1) Randomly select one of the string ends
2) Sample string break flavor
3) Sample transverse momentum of new quarks

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

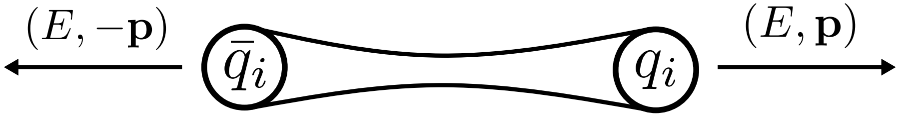
4) Sample longitudinal momentum fraction of new hadron

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

5) Repeat steps 1-4



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



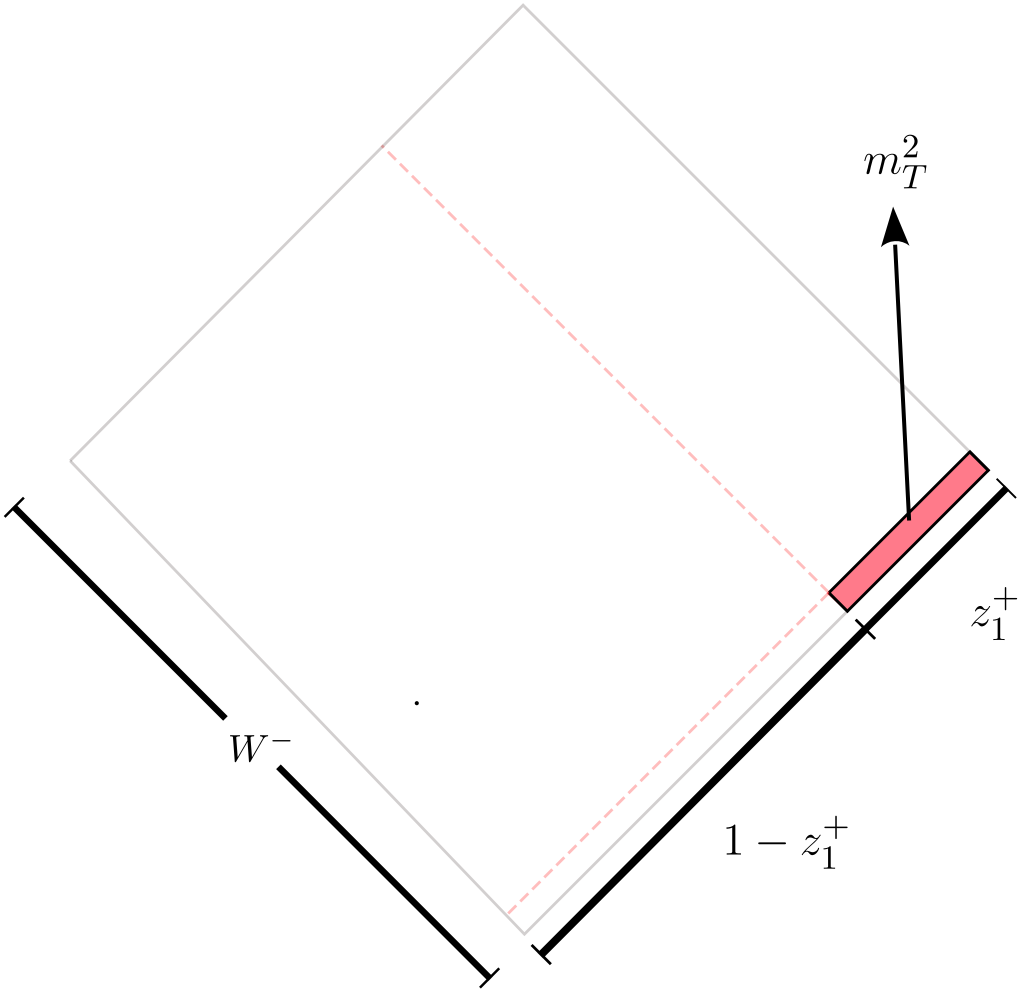
1) Randomly select one of the string ends
2) Sample string break flavor
3) Sample transverse momentum of new quarks

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

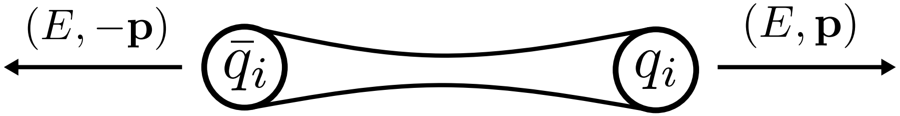
4) Sample longitudinal momentum fraction of new hadron

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

5) Repeat steps 1-4



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



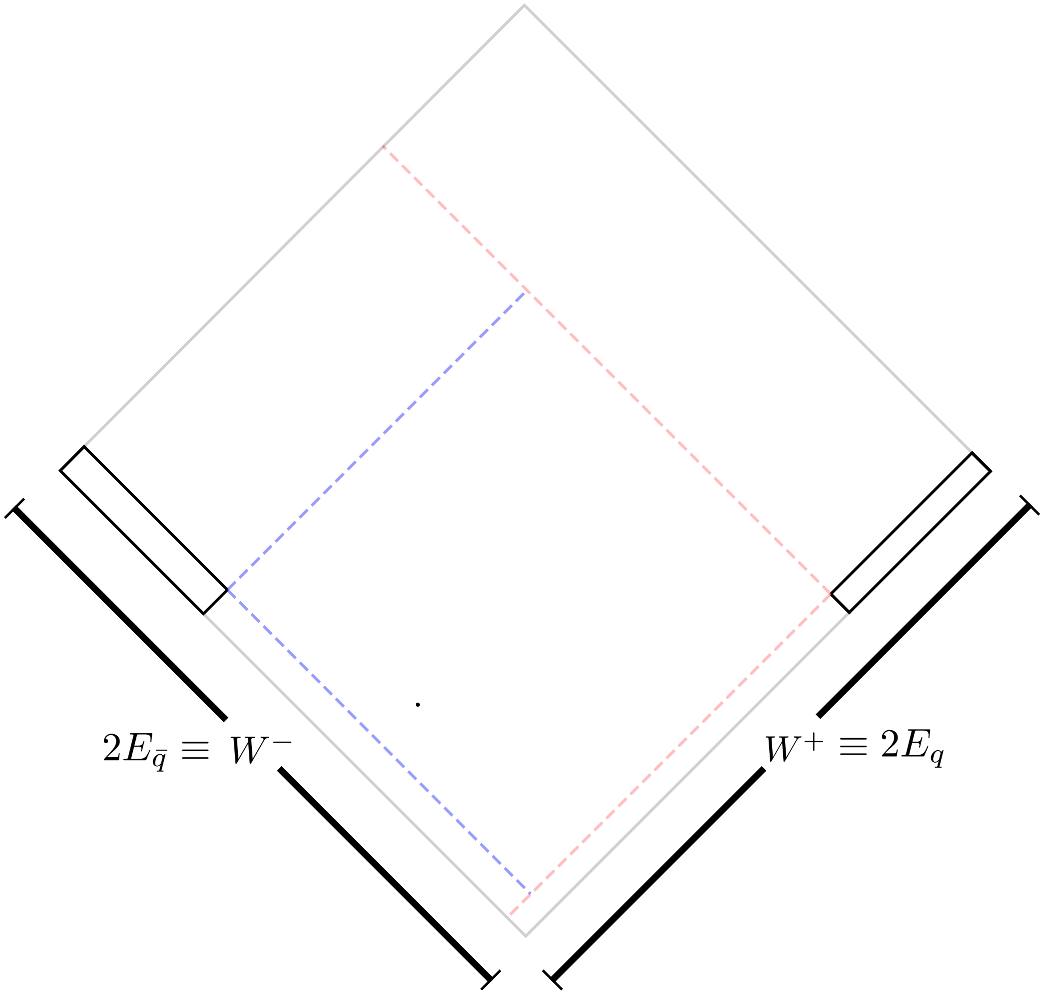
1) Randomly select one of the string ends
2) Sample string break flavor
3) Sample transverse momentum of new quarks

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

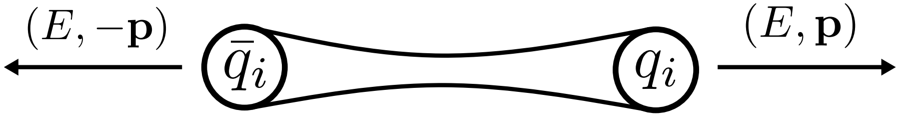
4) Sample longitudinal momentum fraction of new hadron

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

5) Repeat steps 1-4



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



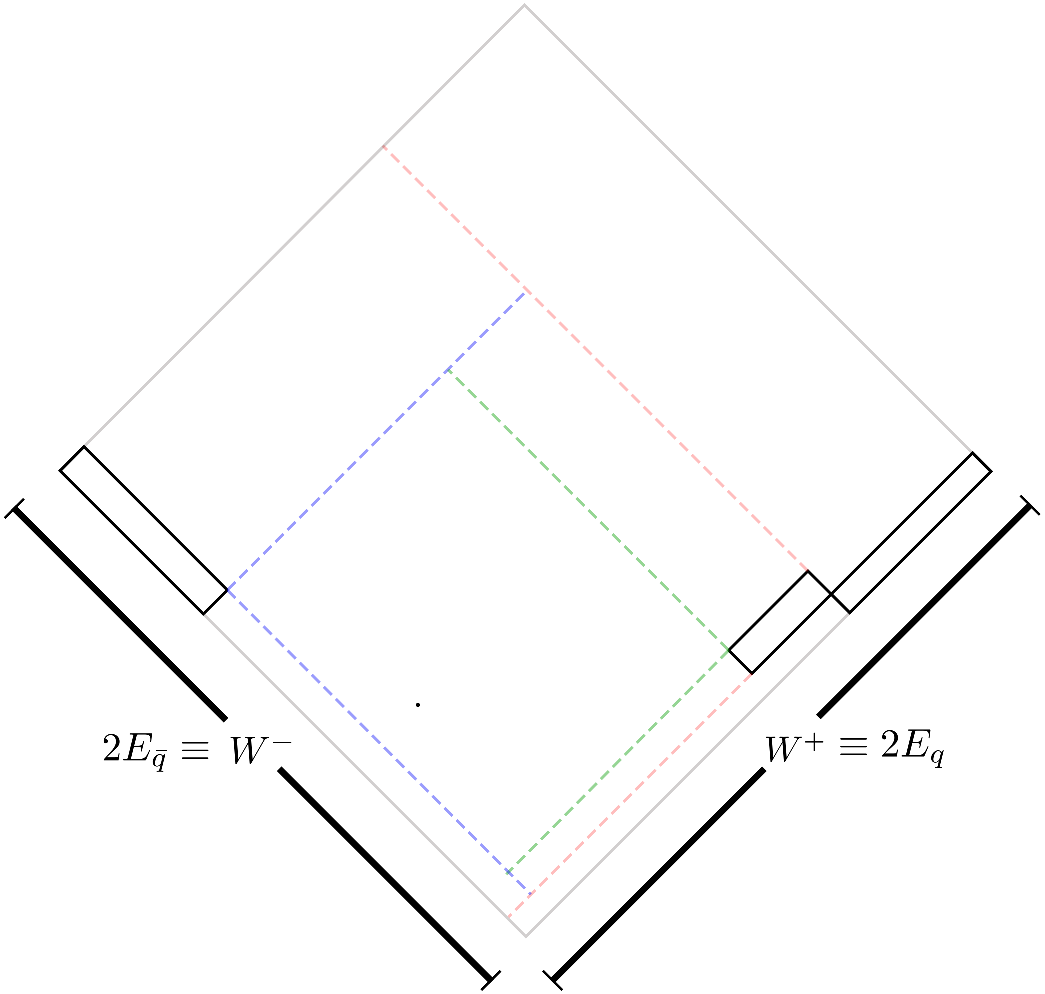
1) Randomly select one of the string ends
2) Sample string break flavor
3) Sample transverse momentum of new quarks

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

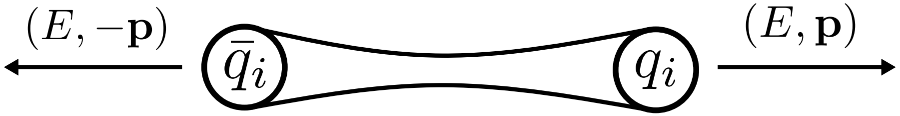
4) Sample longitudinal momentum fraction of new hadron

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

5) Repeat steps 1-4



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



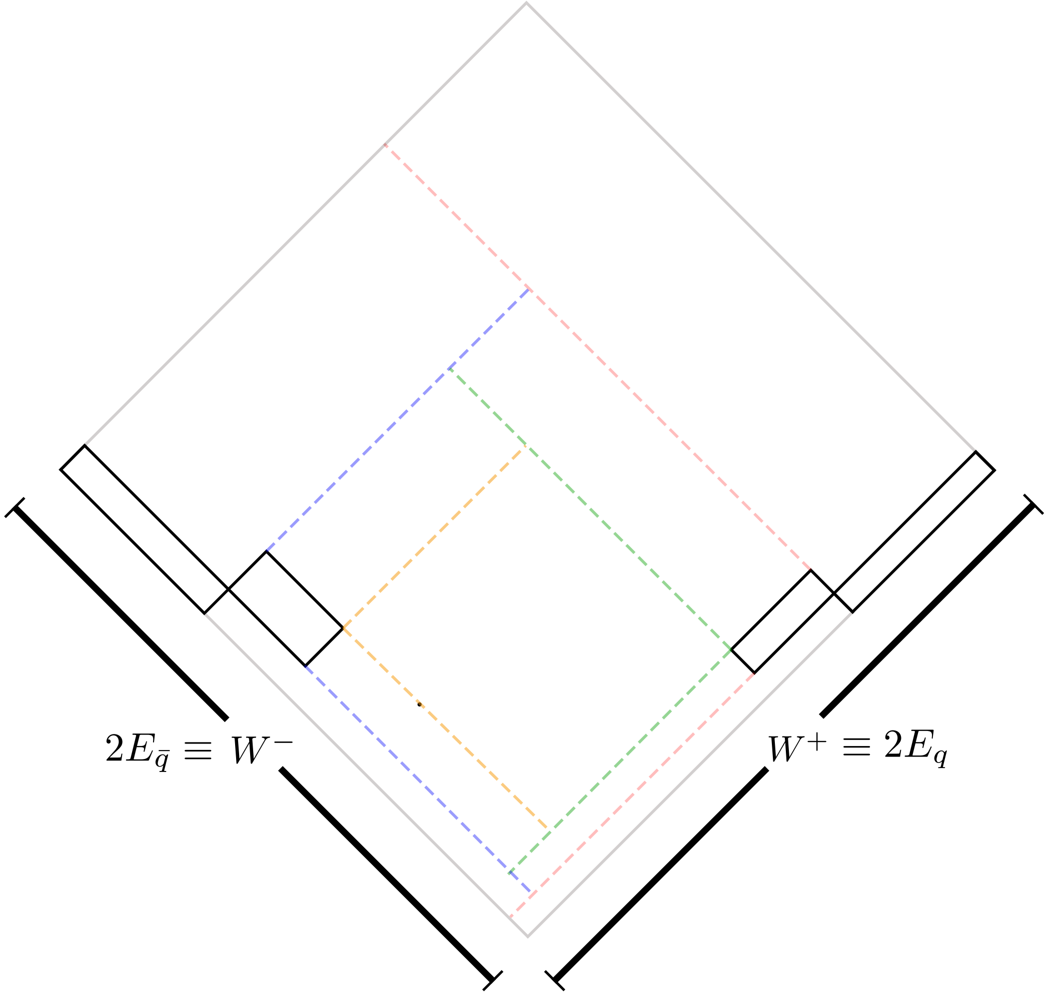
1) Randomly select one of the string ends
2) Sample string break flavor
3) Sample transverse momentum of new quarks

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

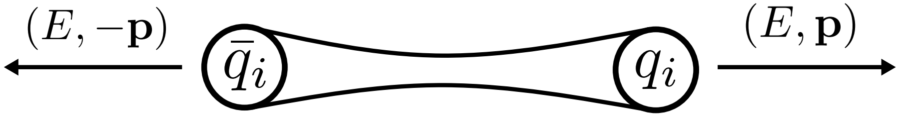
4) Sample longitudinal momentum fraction of new hadron

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

5) Repeat steps 1-4



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



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

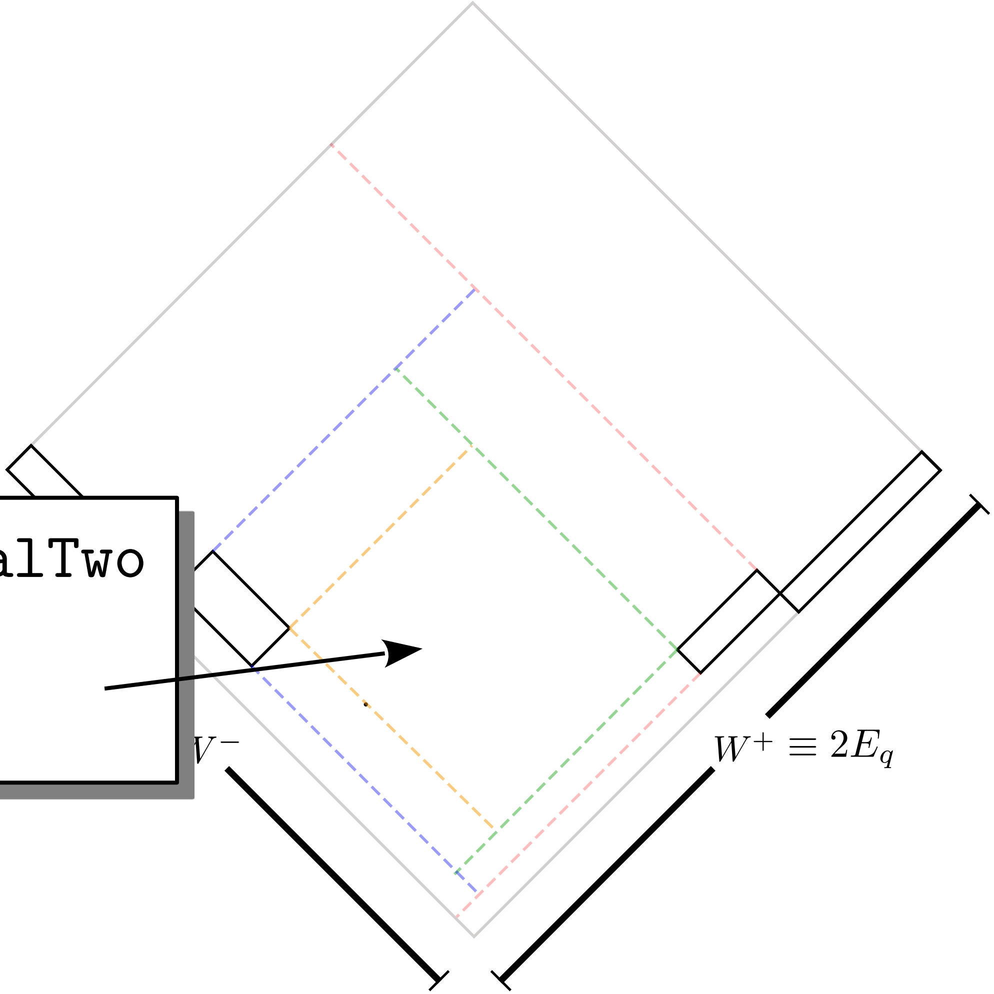
$\mathcal{P}($

4) Samp
new h

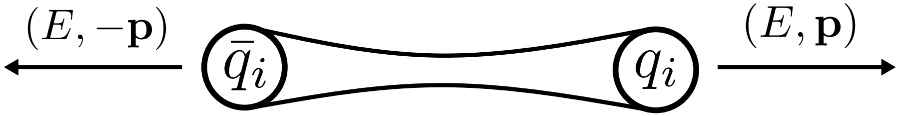
$f(z) \propto \frac{}{z} \exp\left(\frac{}{z}\right), \quad z = \frac{}{2E}$

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$\bar{q}$)



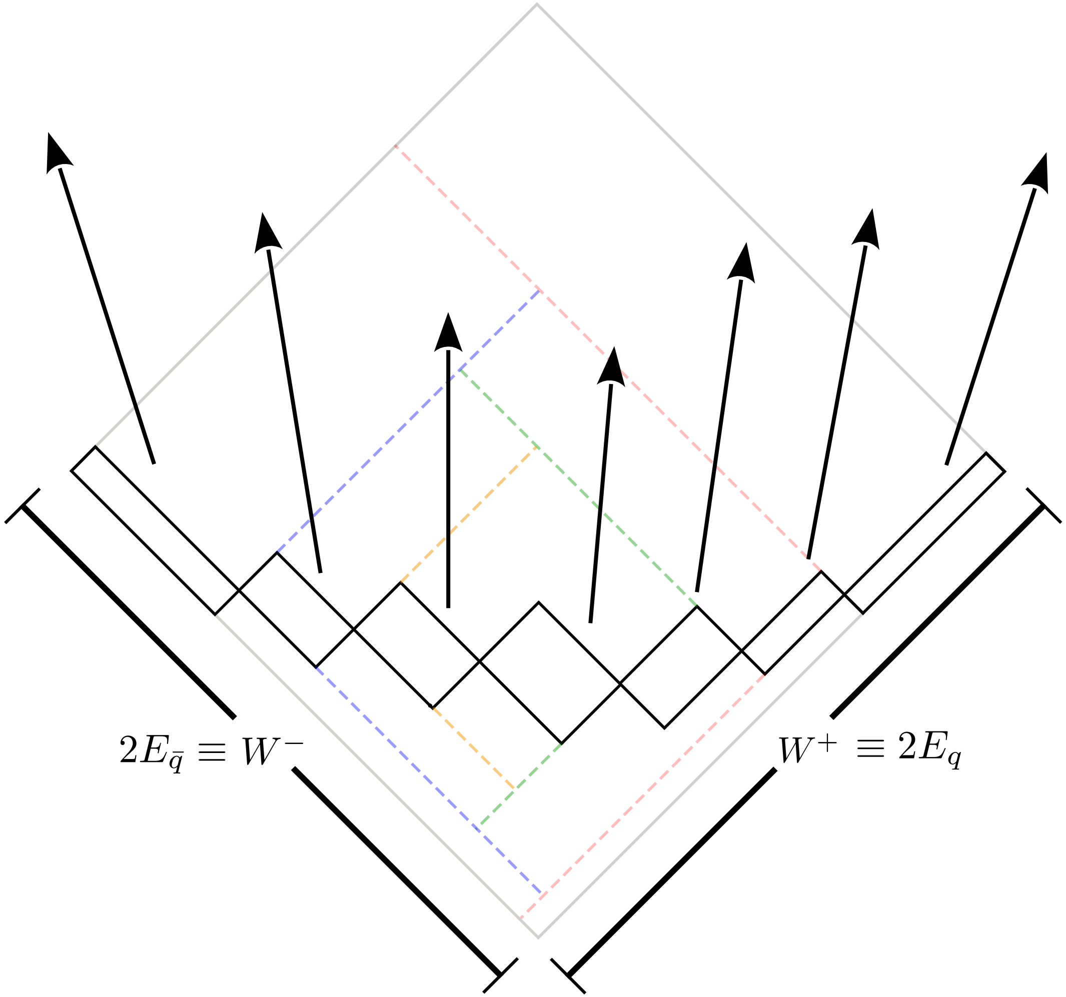
1) Randomly select one of the string ends
2) Sample string break flavor
3) Sample transverse momentum of new quarks

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

4) Sample longitudinal momentum fraction of new hadron

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

5) Repeat steps 1-4



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

$(E, -\mathbf{p})$

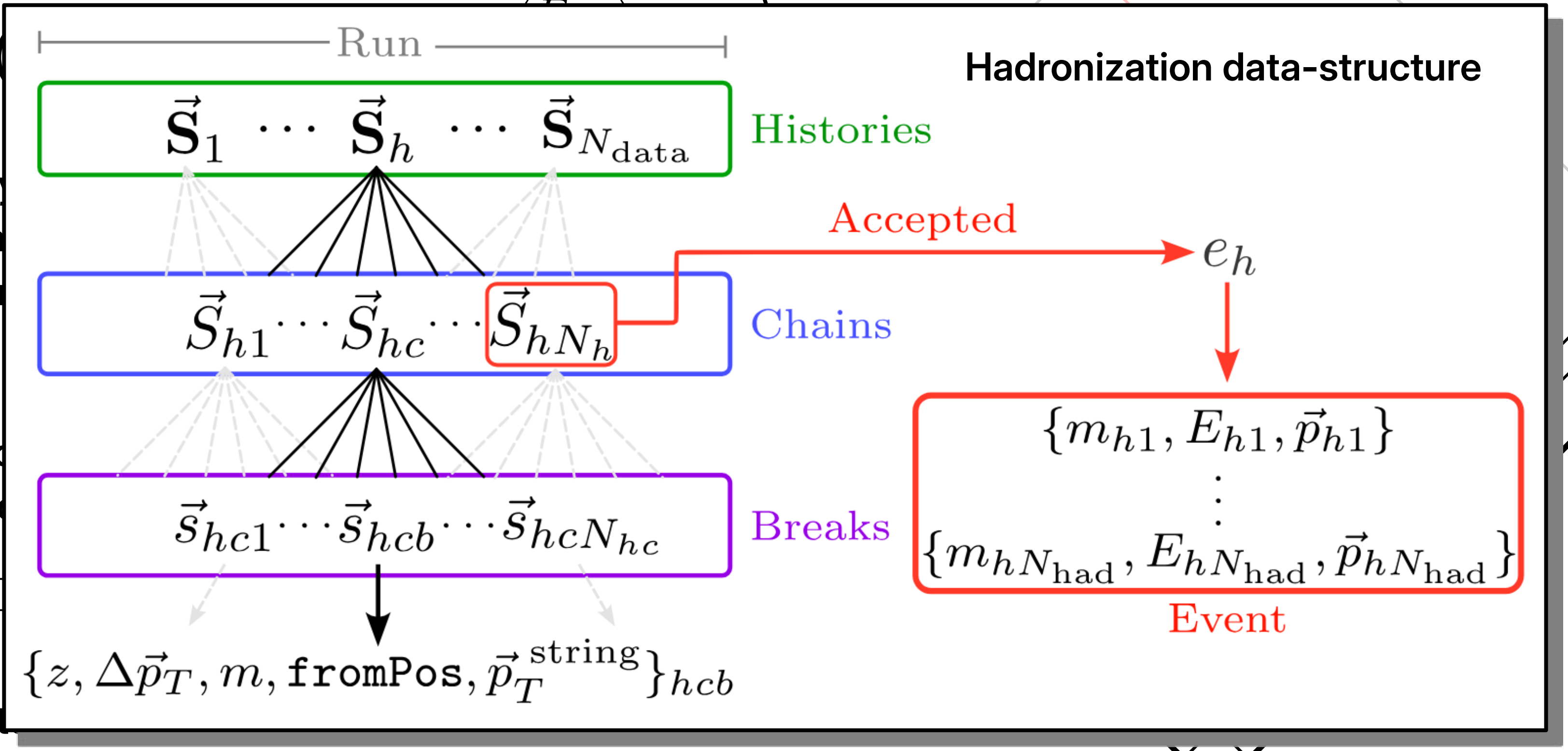
1) Randomly
2) Sample n
3) Sample t

$\mathcal{P}(p_x,$

4) Sample lo
   new hadr

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

5) Repeat st



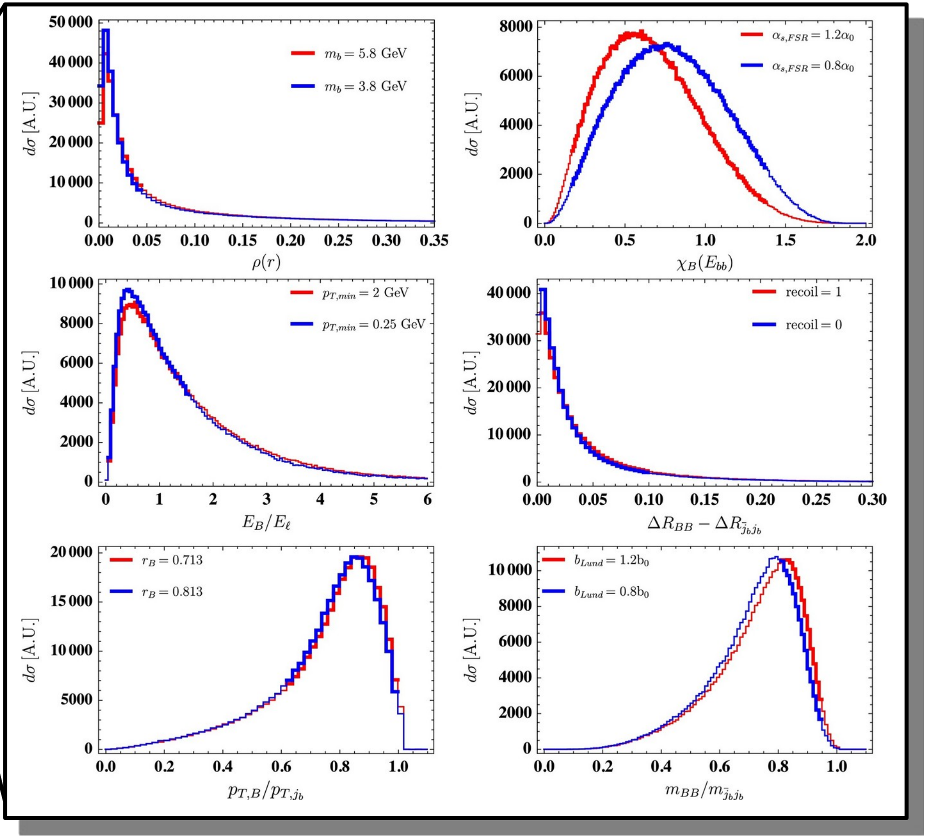
$2E_q$

Why?

Precision (exclusive) measurements are sensitive to hadronization!

- Uncertainty reduction
  - Top quark mass measurement (r_b)
  - e^+e^- determination of α_s

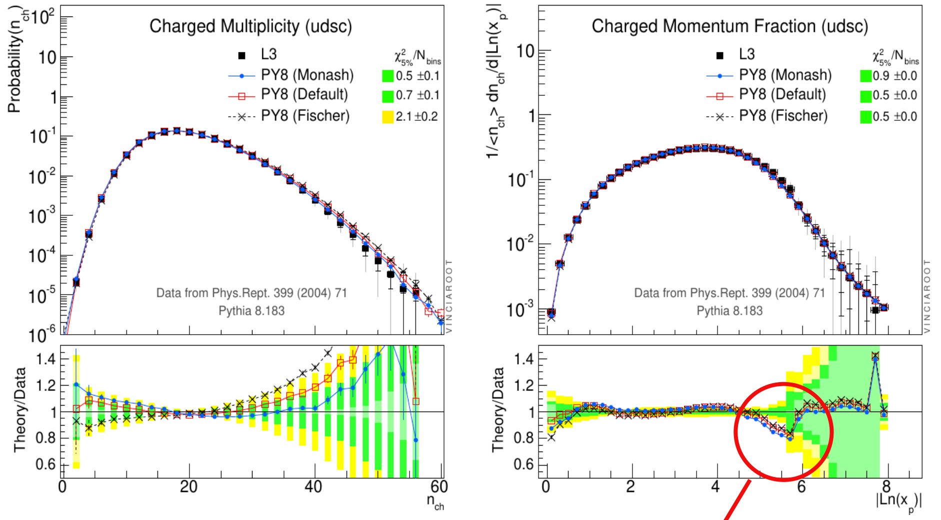
- Mis-modeling bias
  - High-multiplicity events
  - Tuning discrepancies



Why?

Precision (exclusive) measurements are sensitive to hadronization!

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



Monash tune - 1404.5630

Cannot be improved by retuning

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

    New observables must be measured by experiments = **time + $$$**

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

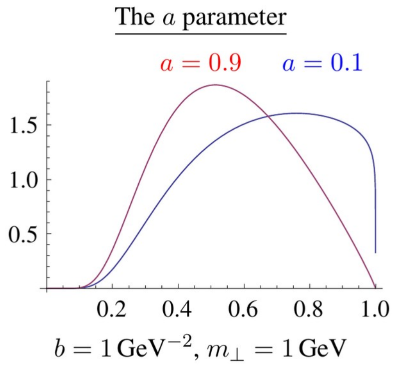
    New observables must be measured by experiments = **time + $$$**

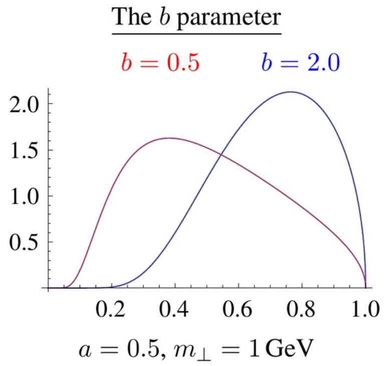
*or some combination of all three

Data-driven fine tuning

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.



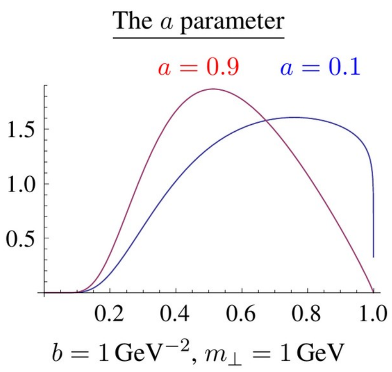


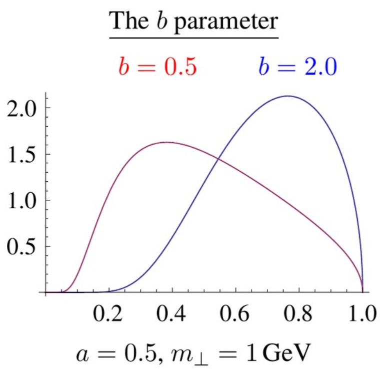
+ ε

Data-driven fine tuning

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.



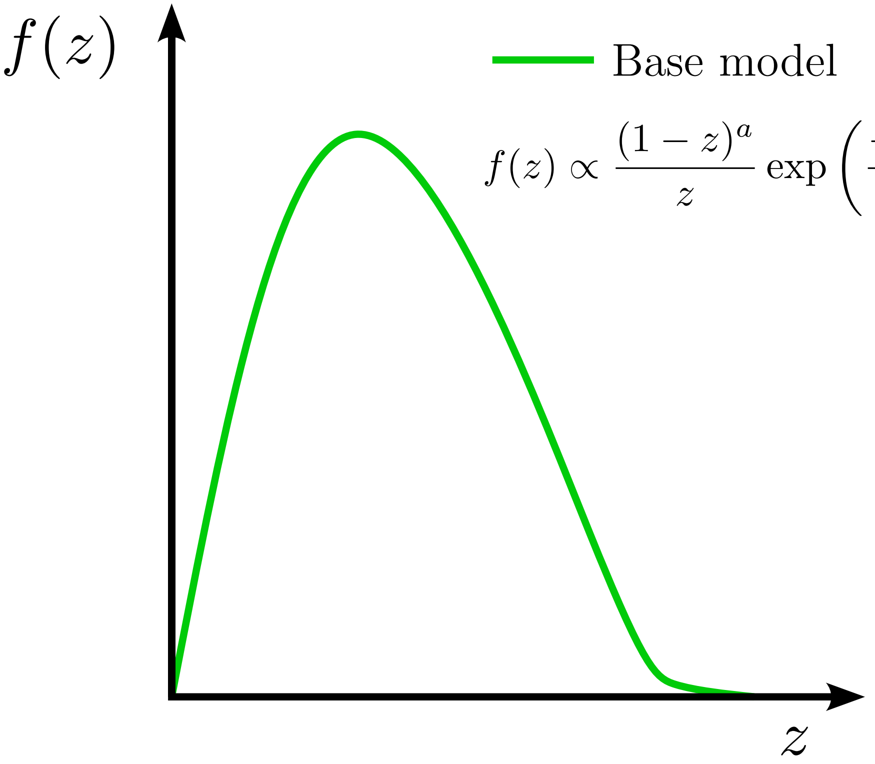


+ ε

Machine learning offers a nice framework to tackle this problem

MLHAD efforts: big picture

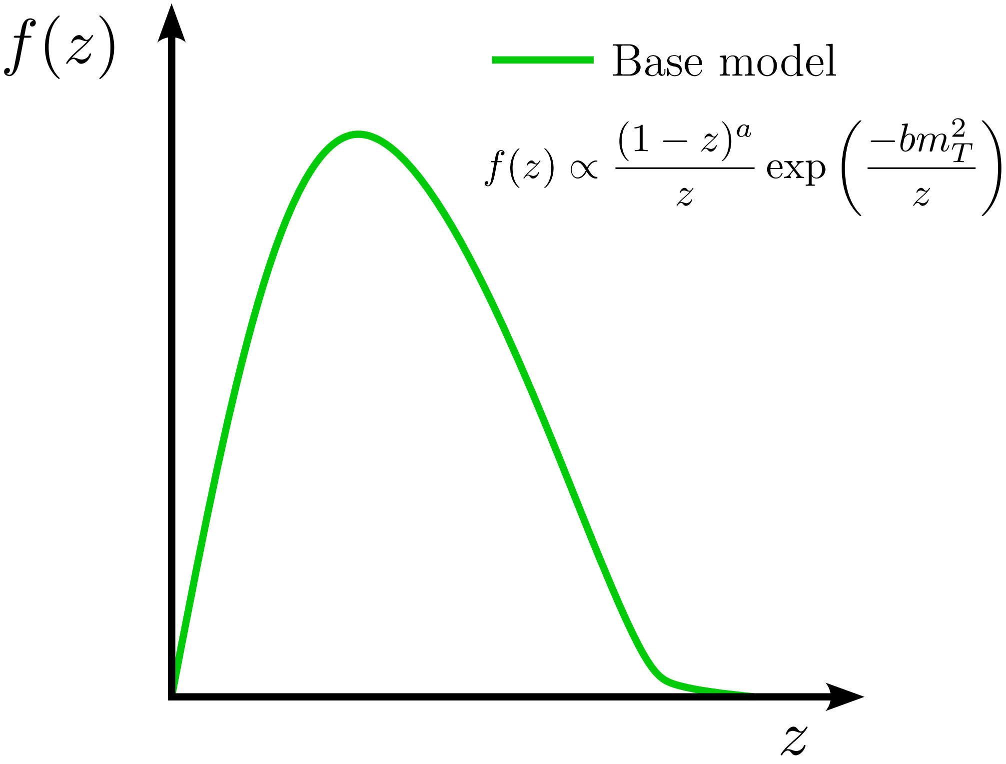
Solving the "inverse problem of hadronization"

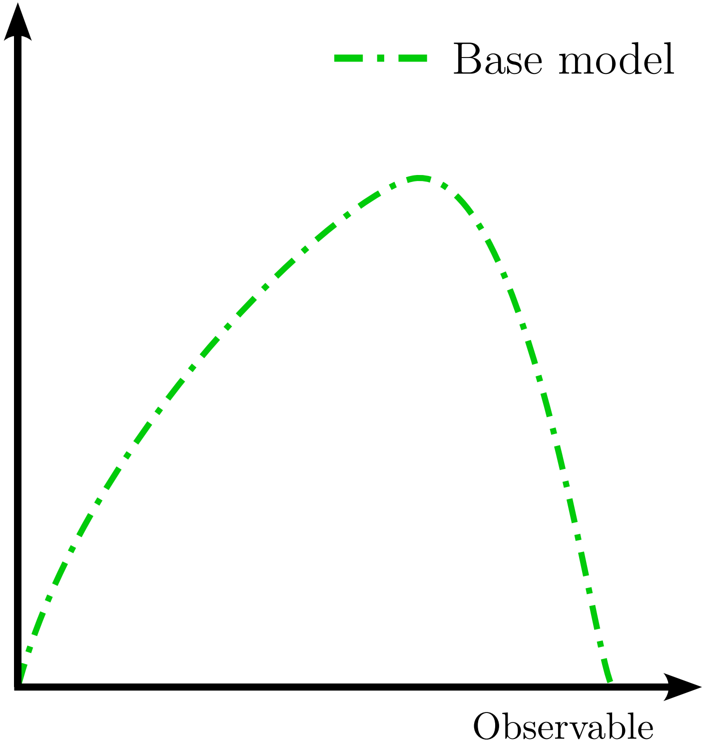


$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_T^2}{z}\right)$$

MLHAD efforts: big picture

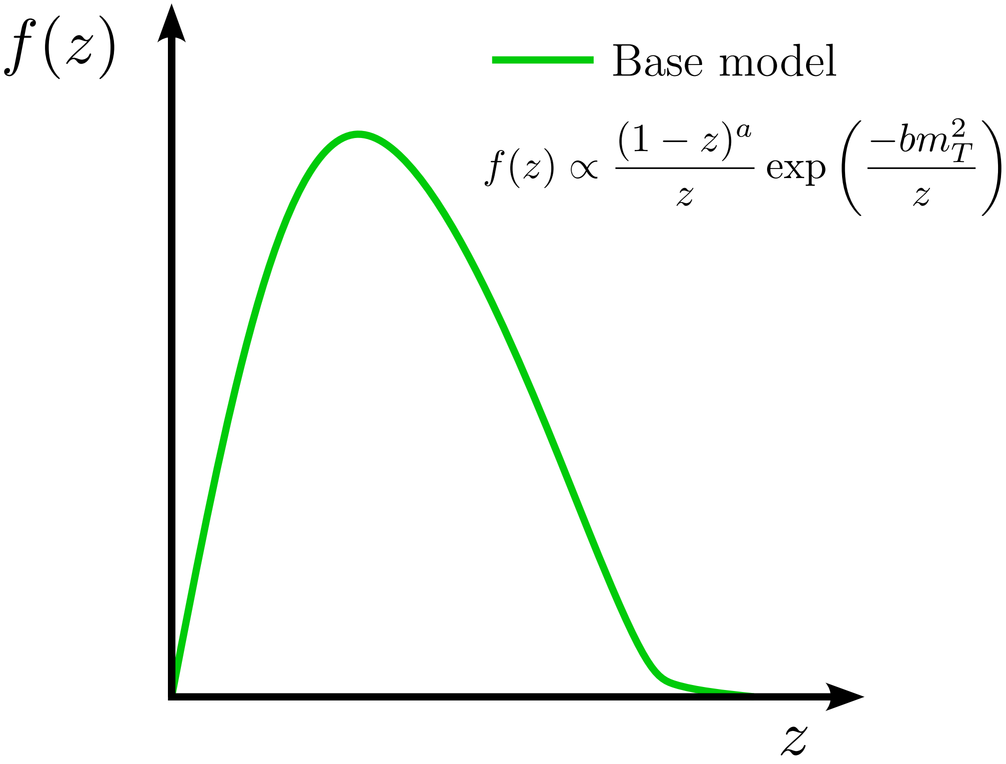
Solving the "inverse problem of hadronization"

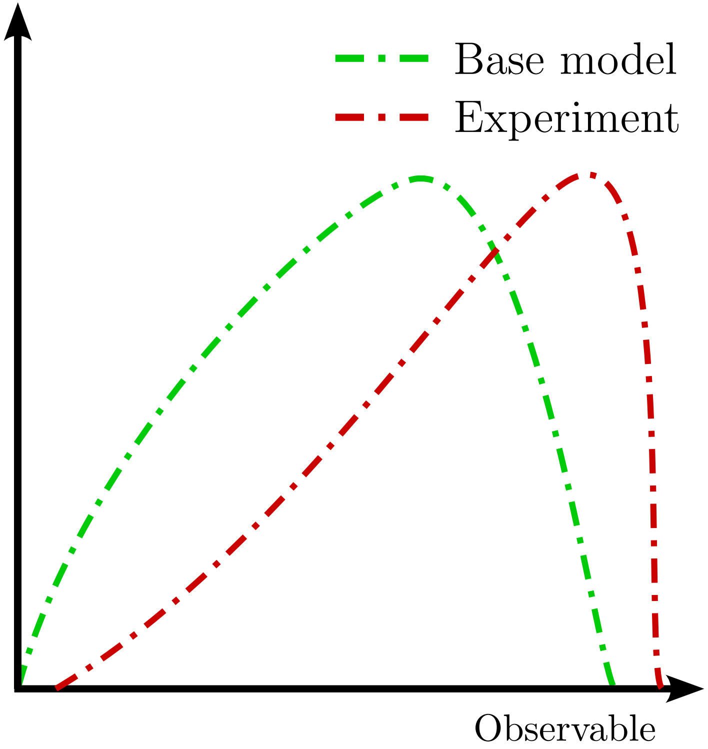




MLHAD efforts: big picture

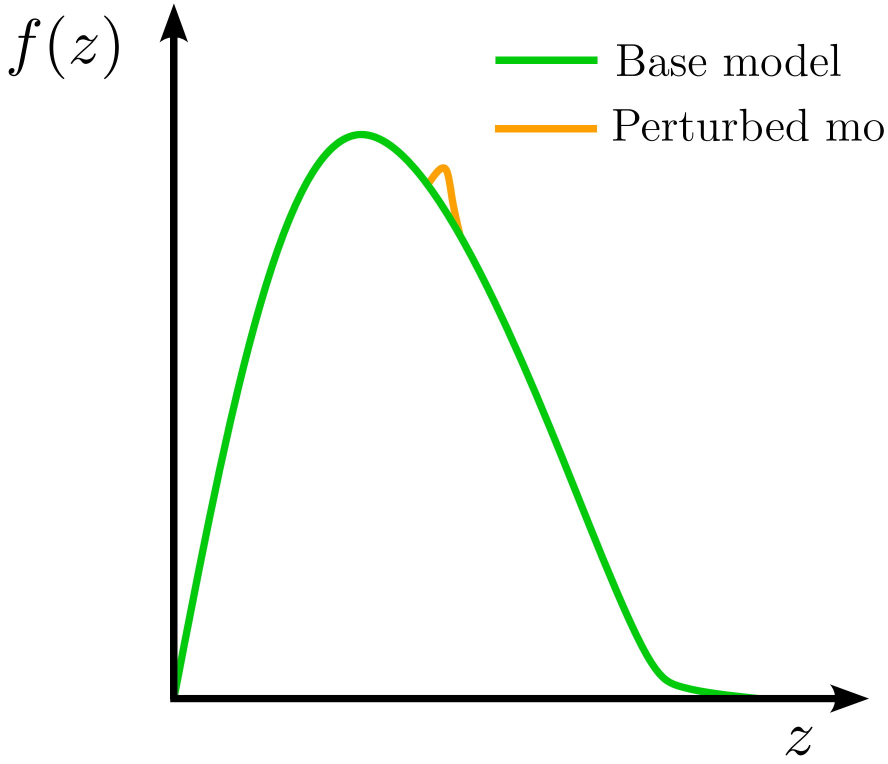
Solving the “inverse problem of hadronization”

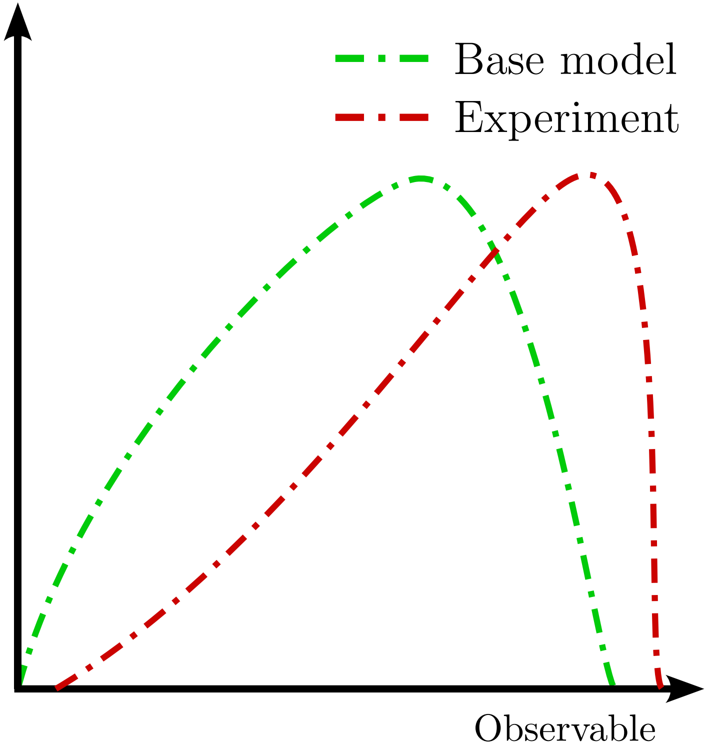




MLHAD efforts: big picture

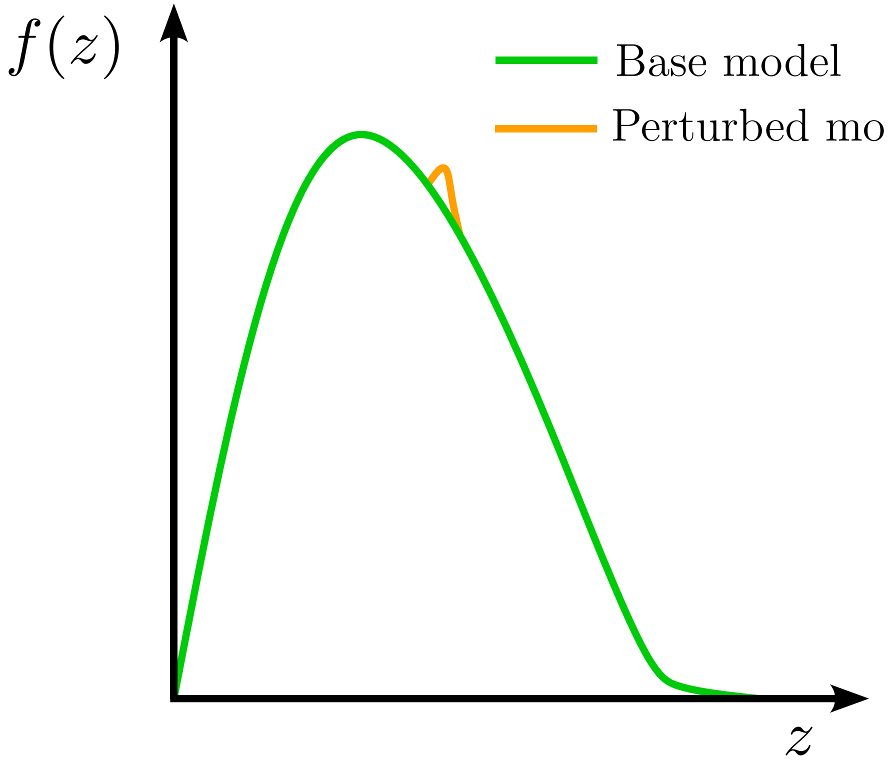
Solving the "inverse problem of hadronization"

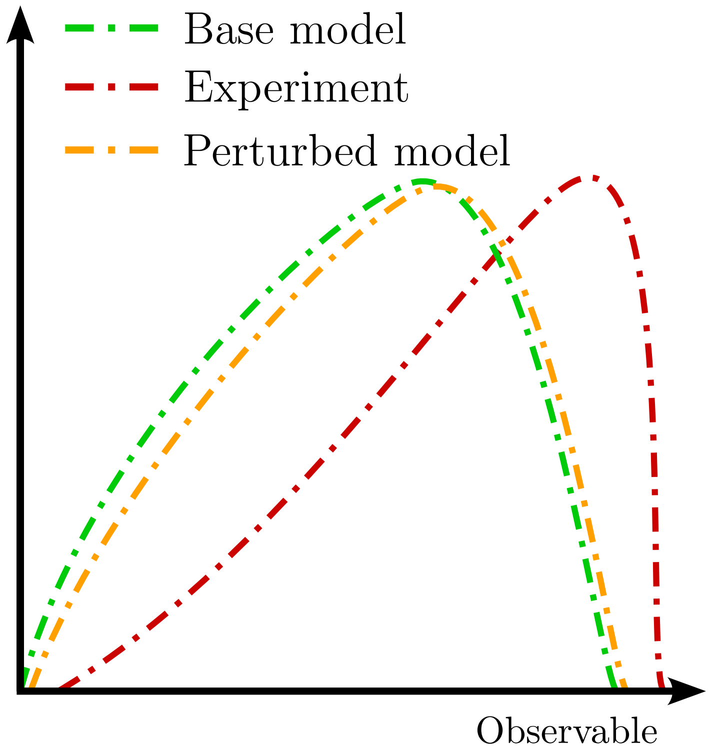




MLHAD efforts: big picture

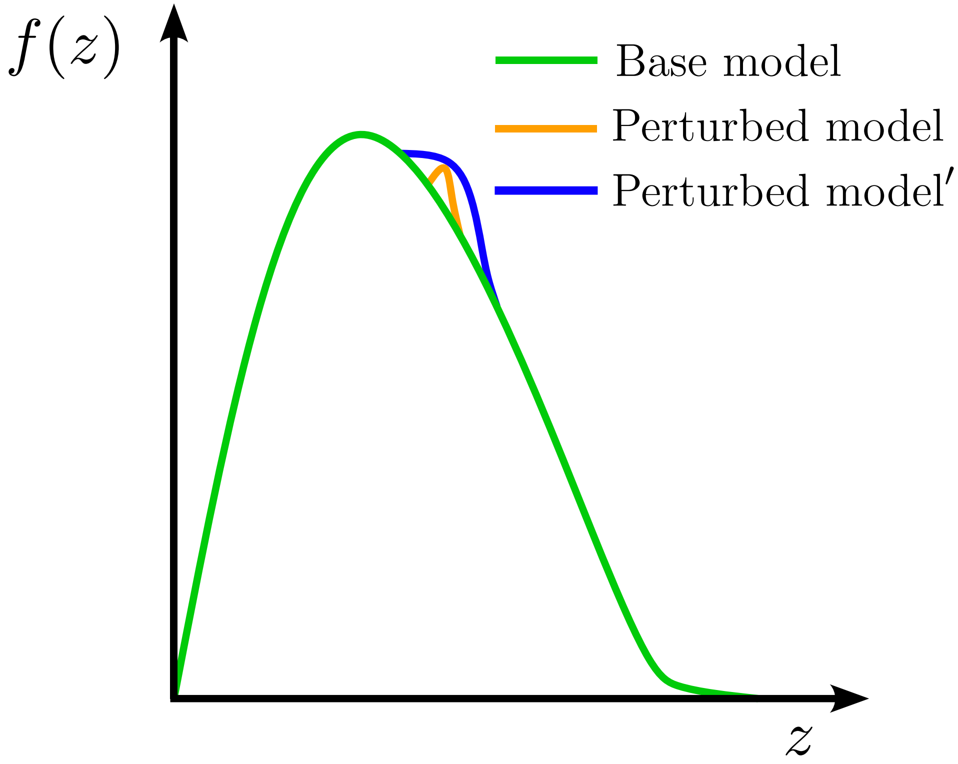
Solving the "inverse problem of hadronization"

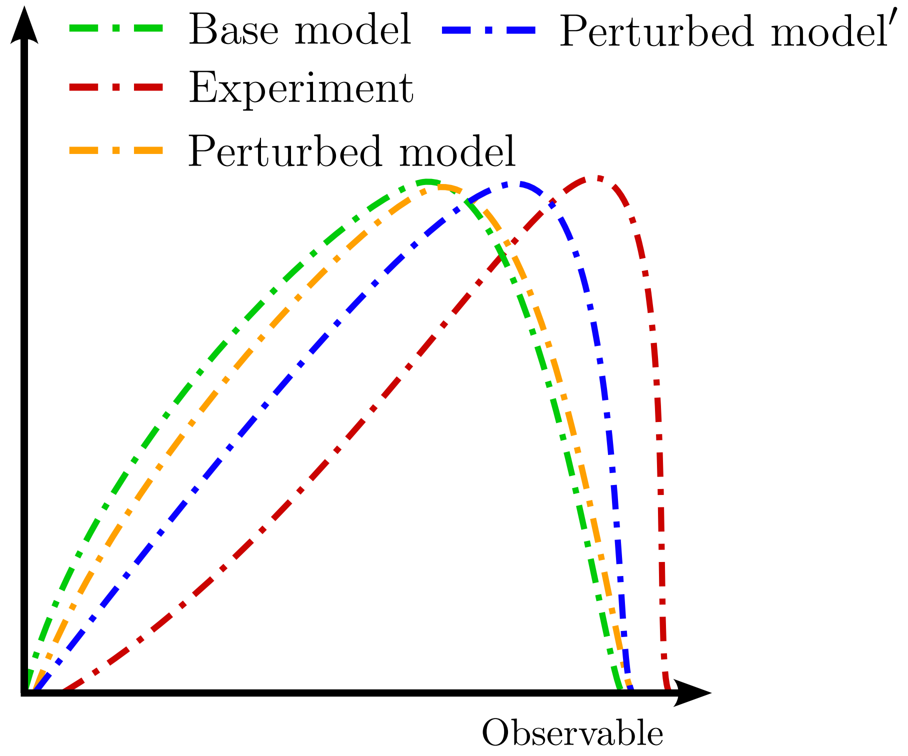




MLHAD efforts: big picture

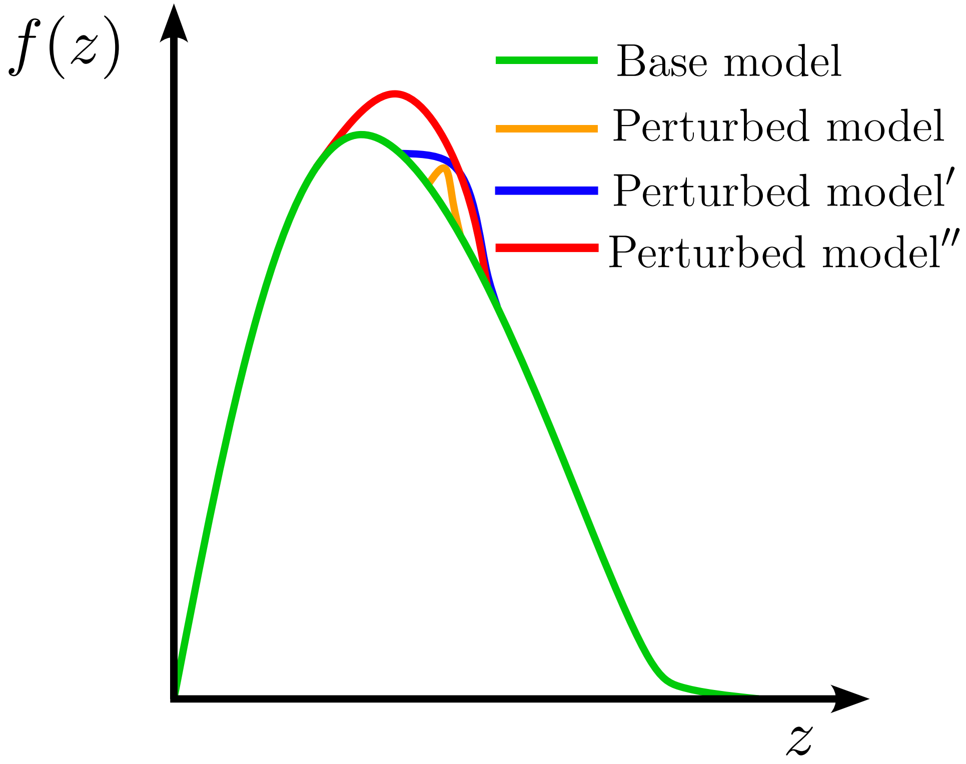
Solving the "inverse problem of hadronization"

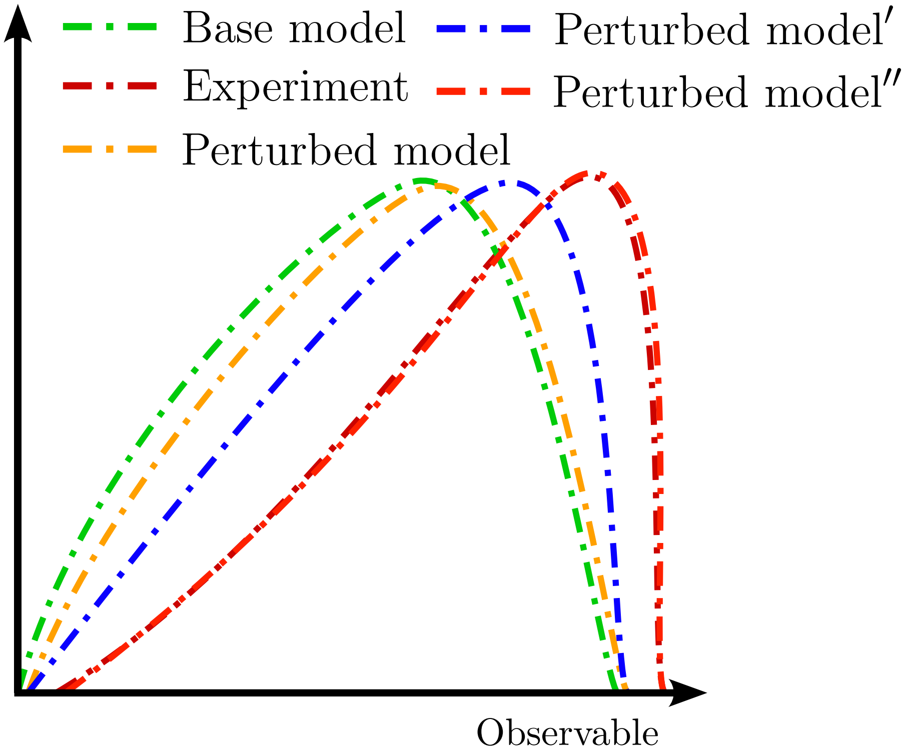




MLHAD efforts: big picture

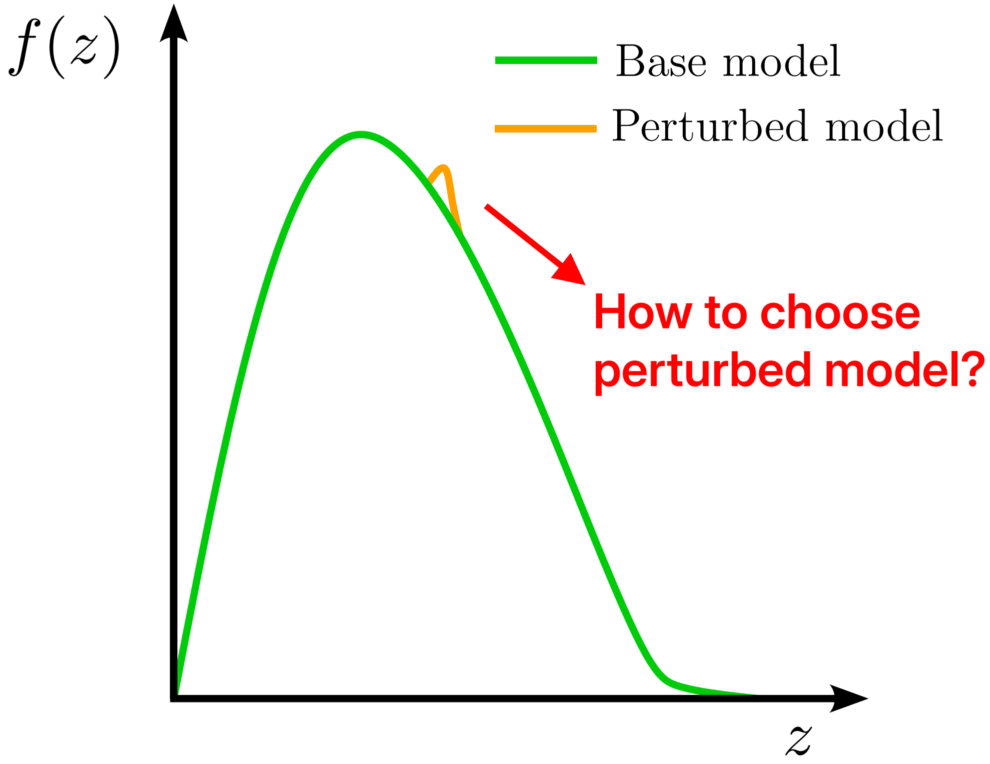
Solving the "inverse problem of hadronization"





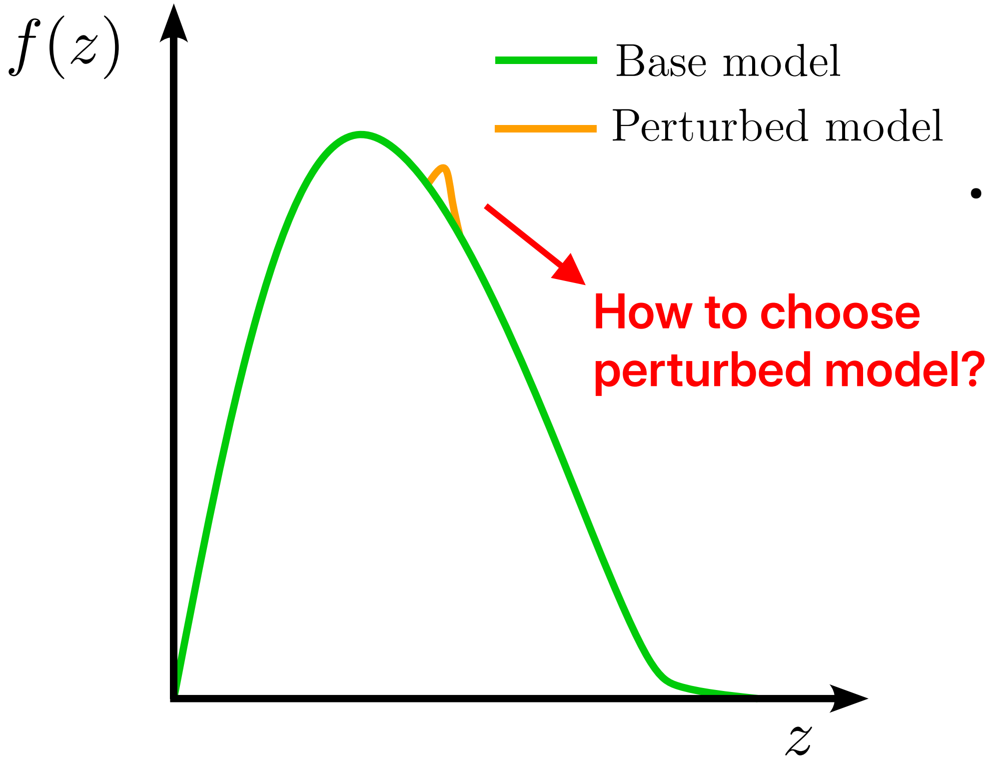
MLHAD efforts: big picture

Solving the "inverse problem of hadronization"



MLHAD efforts: big picture

Solving the “inverse problem of hadronization”



~~ML~~ vs ML

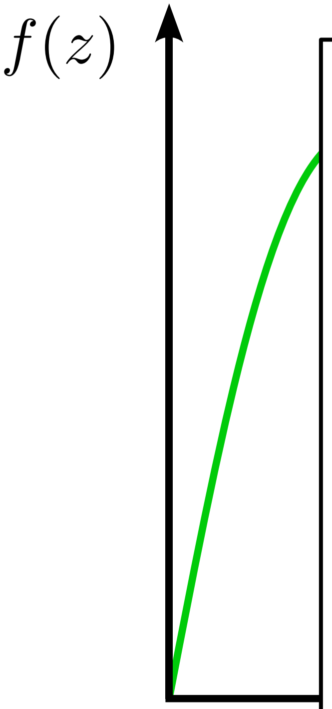
- Vary Lund parameters (traditional “manual”/semi-”manual” tuning)

- ML-based (data-driven) fragmentation function
  - MAGIC, HOMER

- Hybrid – keep Lund fragmentation function, promote Lund parameters to differentiable objects
  - Rejection sampling with autodifferentiation (RSA)

MLHAD efforts: big picture

Solving the "inverse problem of hadronization"



ML

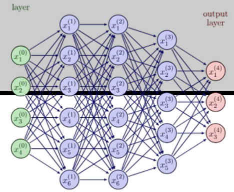
- ML-based (data-driven) fragmentation function
  - MAGIC, HOMER

eep Lund
tion function,
und parameters
tiable objects
on sampling with
erentiation (RSA)

Machine learning

Form of universal function/distribution approximation.

Inputs → Model $\equiv f(\vec{x};\theta)$ → Prediction ↕ Data



The “reweighting revolution”

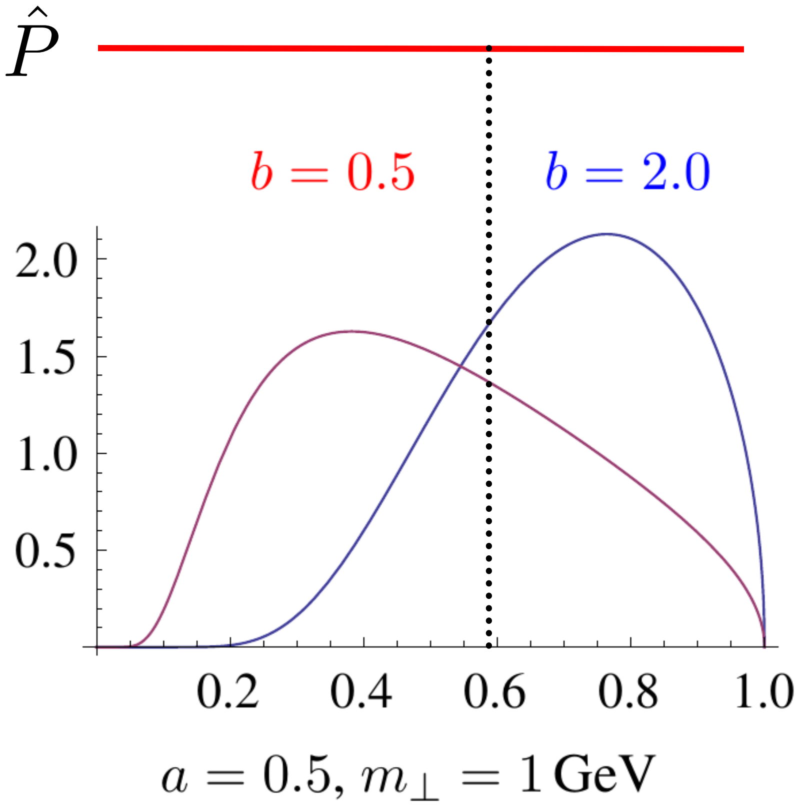
2308.13459 ,2411.02194, 2505.00142, 26xx.xxxxx

In collaboration w/ MLHAD

+

Nick Heller, Ben Nachman, Andrzej Siodmok

Kinematic reweighting (2308.13459)



Rejection sampling:

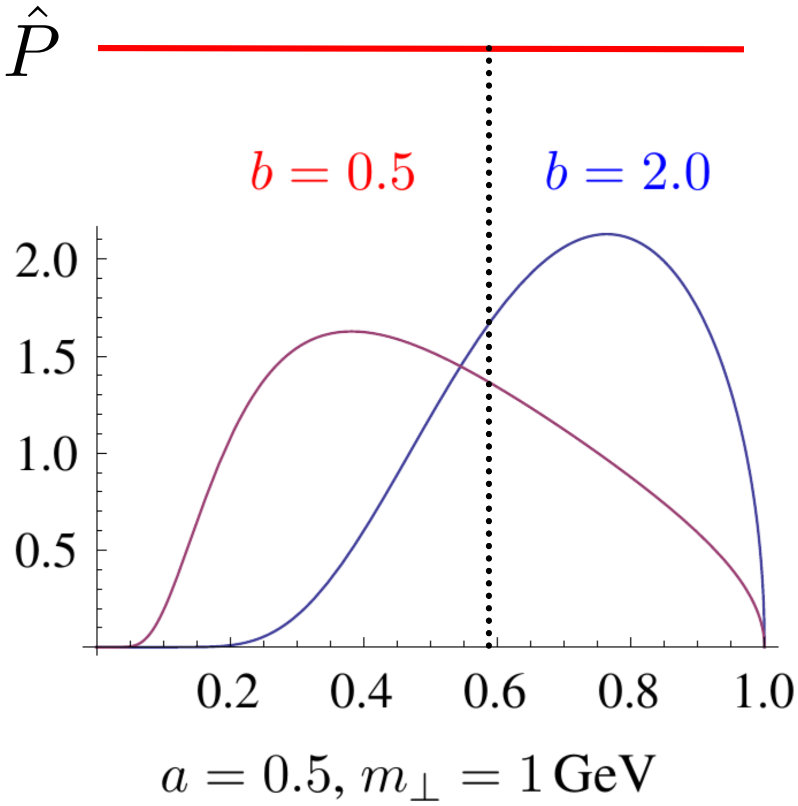
Acceptance probability: $P_{\mathrm{accept}} = \dfrac{p(z,\theta)}{\hat{P}}$

Rejection probability: $P_{\mathrm{reject}} = 1 - P_{\mathrm{accept}}$

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

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

Kinematic reweighting (2308.13459)



Rejection sampling:

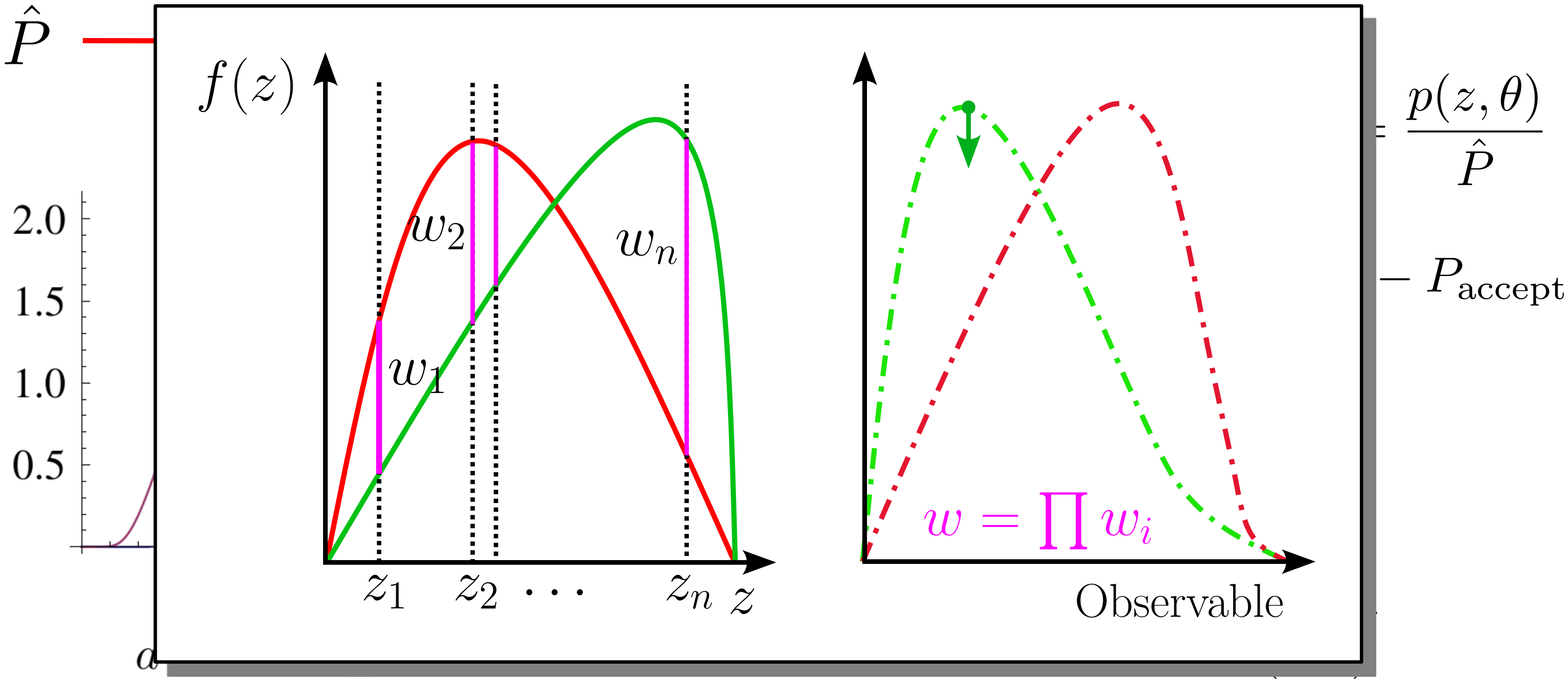
Acceptance probability: $P_{\mathrm{accept}} = \dfrac{p(z,\theta)}{\hat{P}}$

Rejection probability: $P_{\mathrm{reject}} = 1 - P_{\mathrm{accept}}$

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

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

Kinematic reweighting (2308.13459)



Kinematic reweighting (2308.13459)

$\hat{P}$

Rejection sampling:

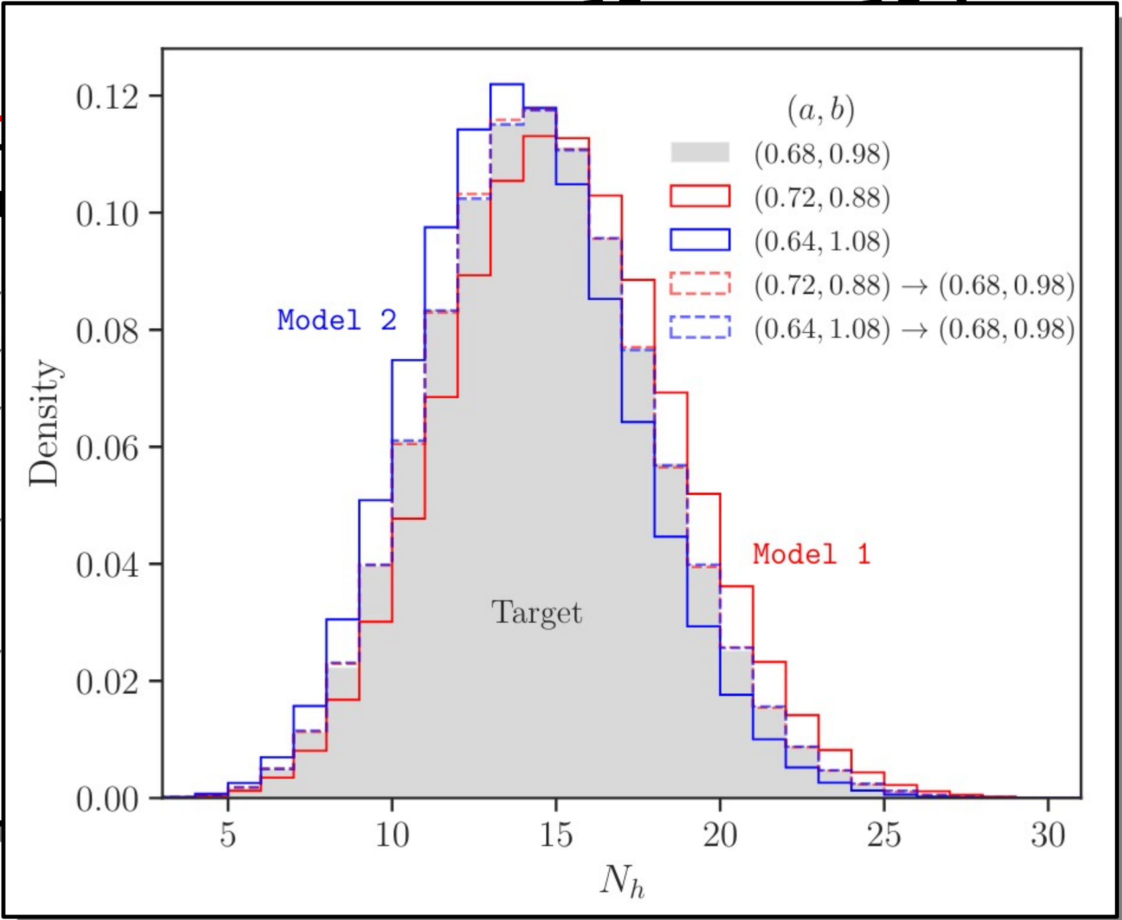
Data-structure:

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

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

Kinematic reweighting (2308.13459)

Data-structu



Kinematic reweighting (2308.13459)

$\hat{P}$

Rejection sampling:

Data-structure:

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

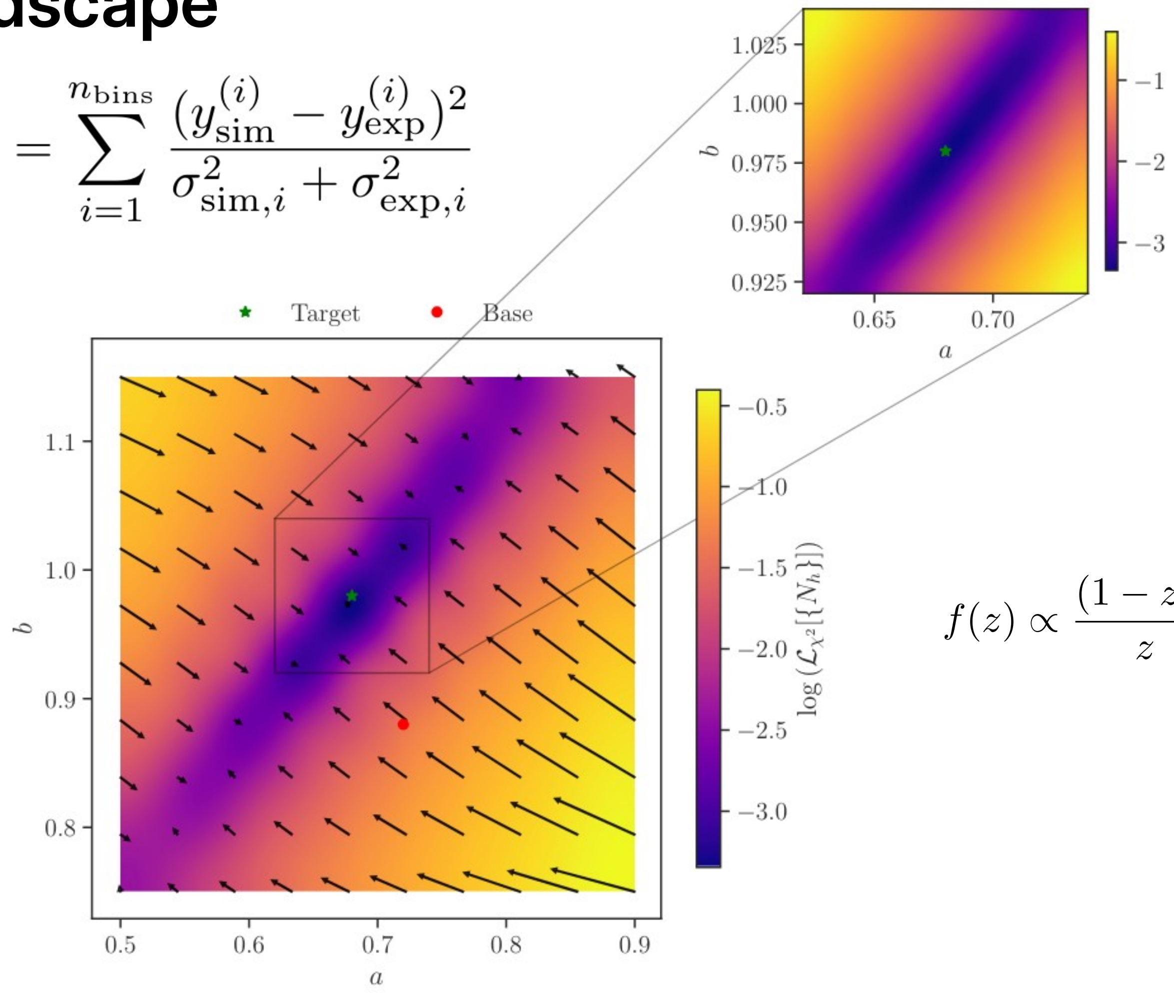
Differentiable! → RSA

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

$a = 0.5,\ m_\perp = 1\,\text{GeV}$ $\qquad$ $P = p(z,\theta)$

χ^2-loss landscape

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

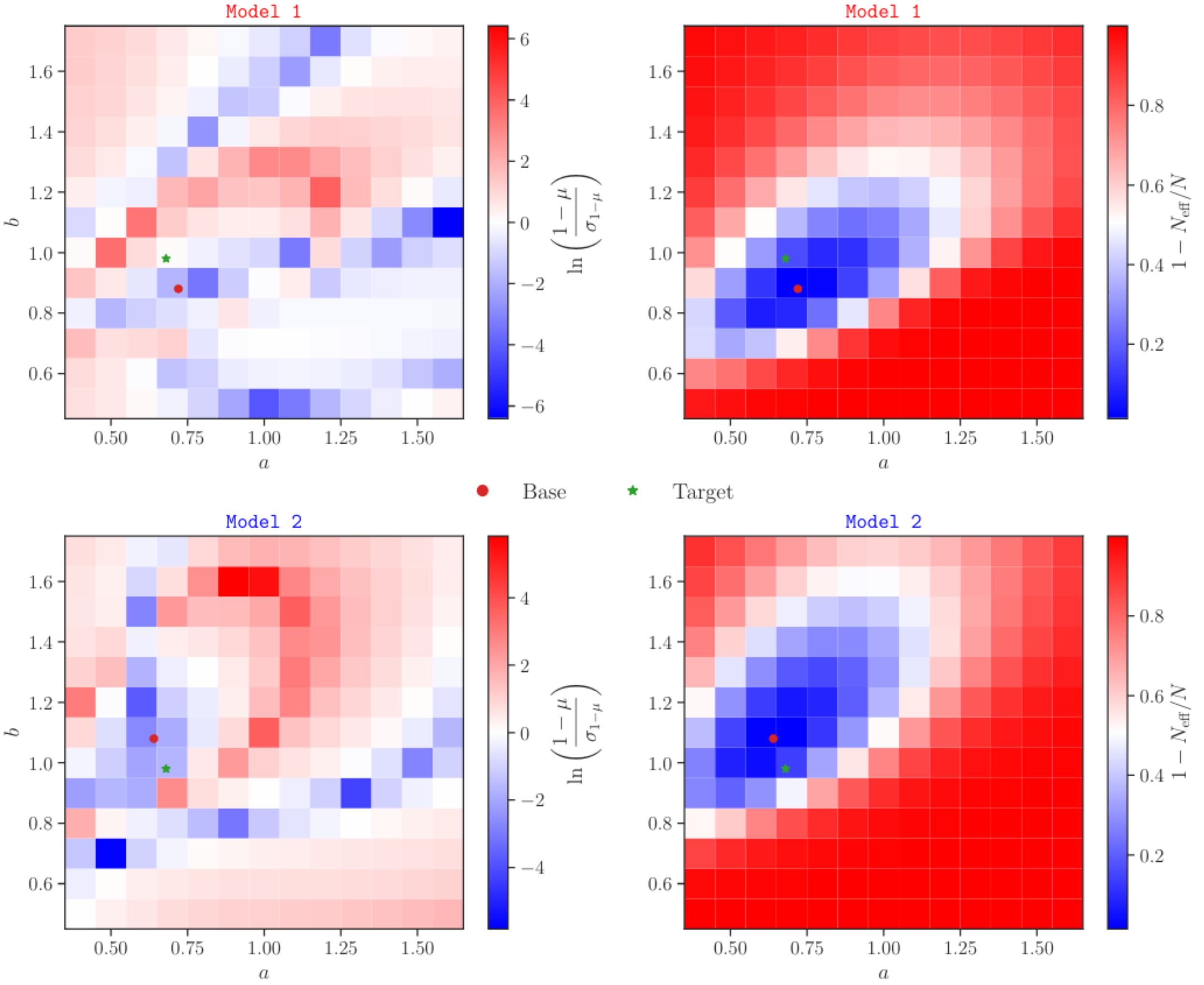


$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(\frac{-bm_T^2}{z}\right)$$

No free lunch

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

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



ML-based observables

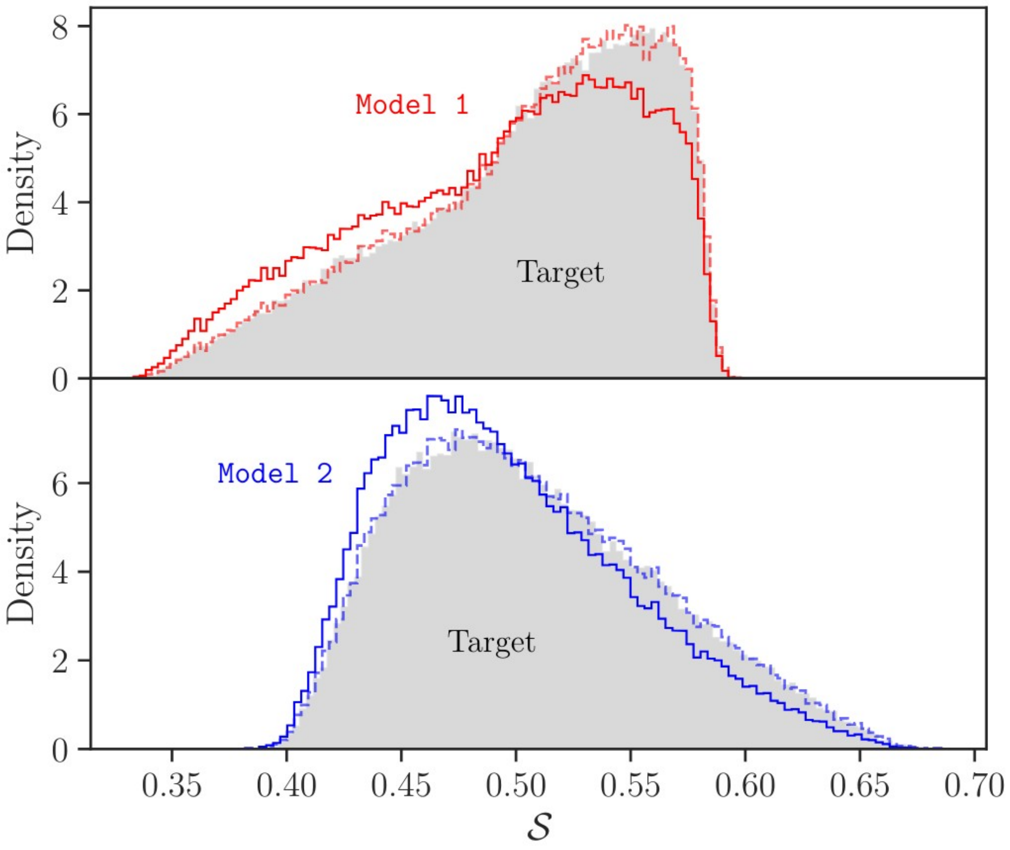
- Train a deepsets classifier to distinguish simulation from experiment
  - Takes full event information as input

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

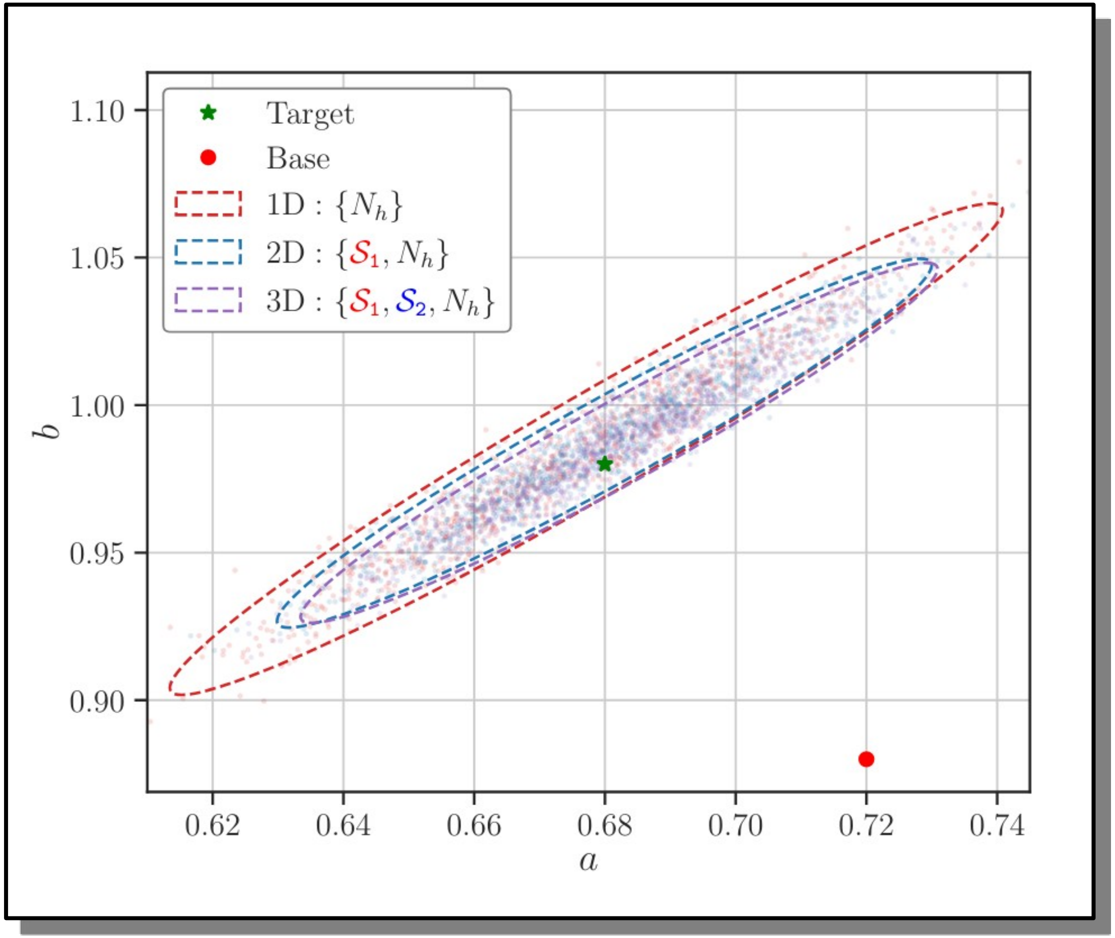
$(E, p_x, p_y, p_z)_2$

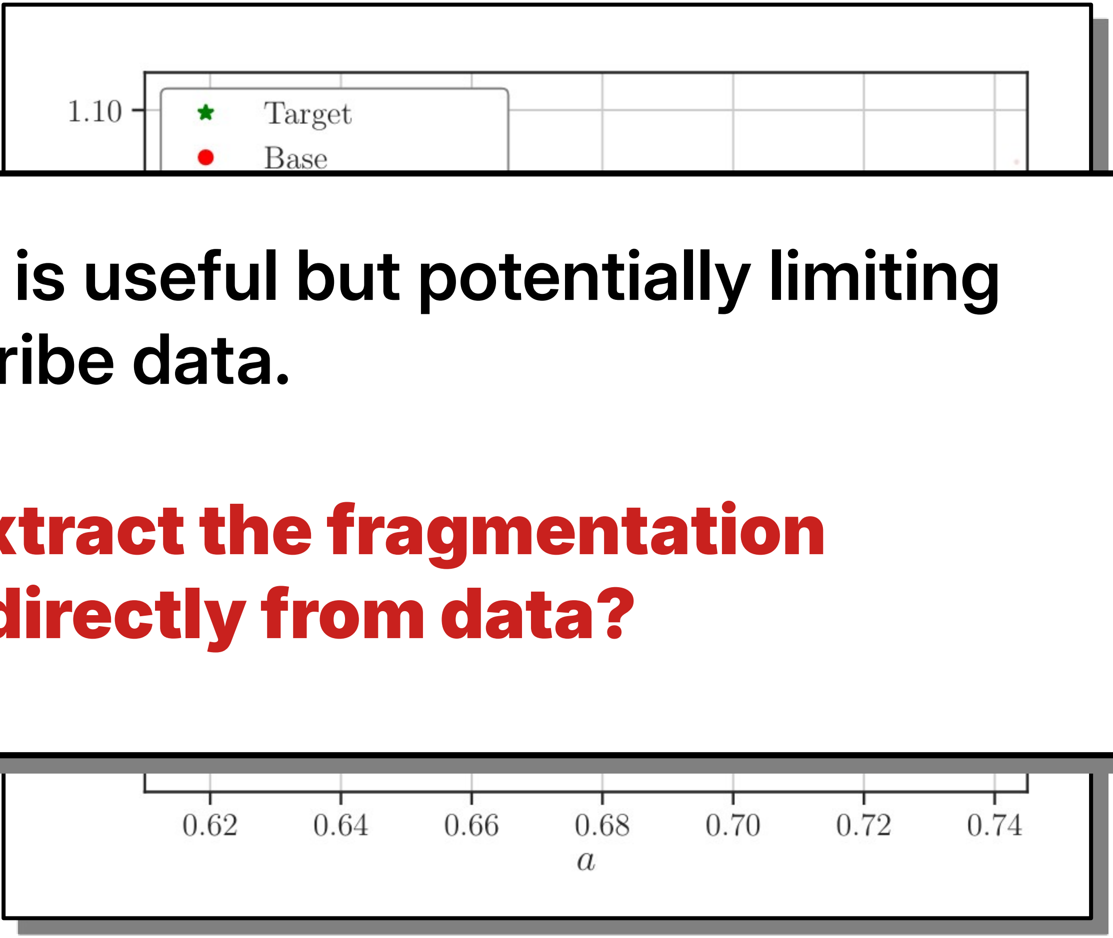
...

Use trained classifier output (score) as an observable



Classifier score with full event info improves tuning convergence





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?

Microscopic Alterations Generated from IR Collections

\+ (2311.09296, 25xx.xxxx)

Histories and Observables for Monte Carlo Event Reweighting

(2410.06342, 2503.05667, 2509.03592)

MLHAD

MAGIC

(2311.09296, 25xx.xxxx)

+

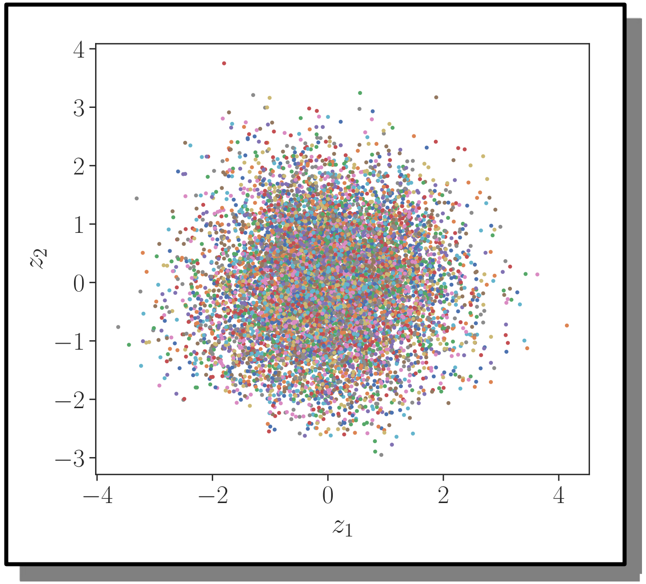
HOMER

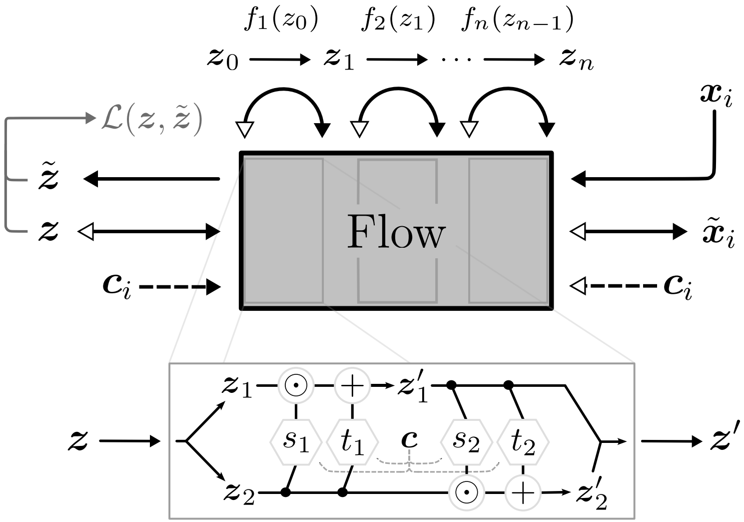
(2410.06342, 2503.05667, 2509.03592)

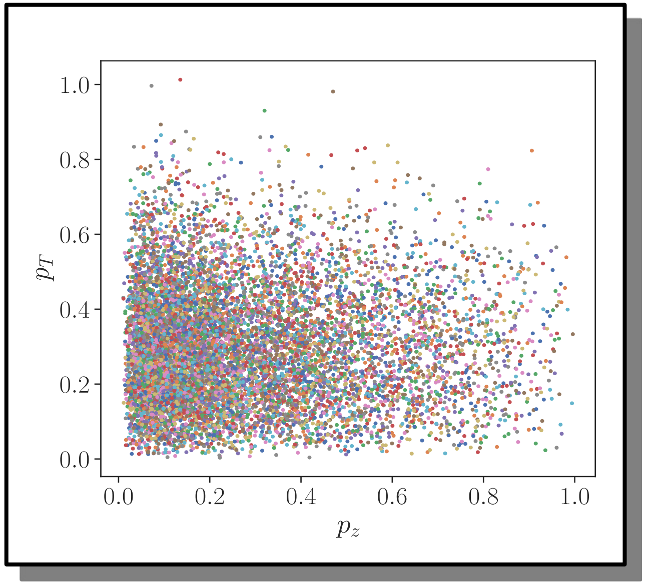
MLHAD

MAGIC: Invertible nerual networks (INN)

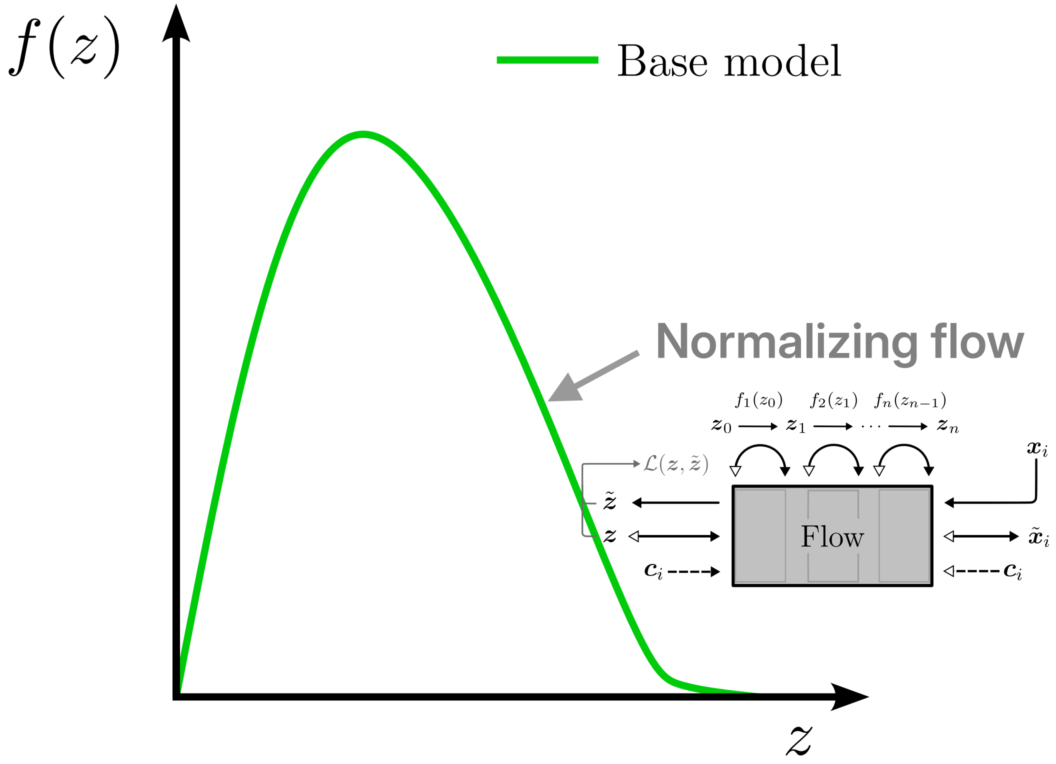
a.k.a normalizing flow

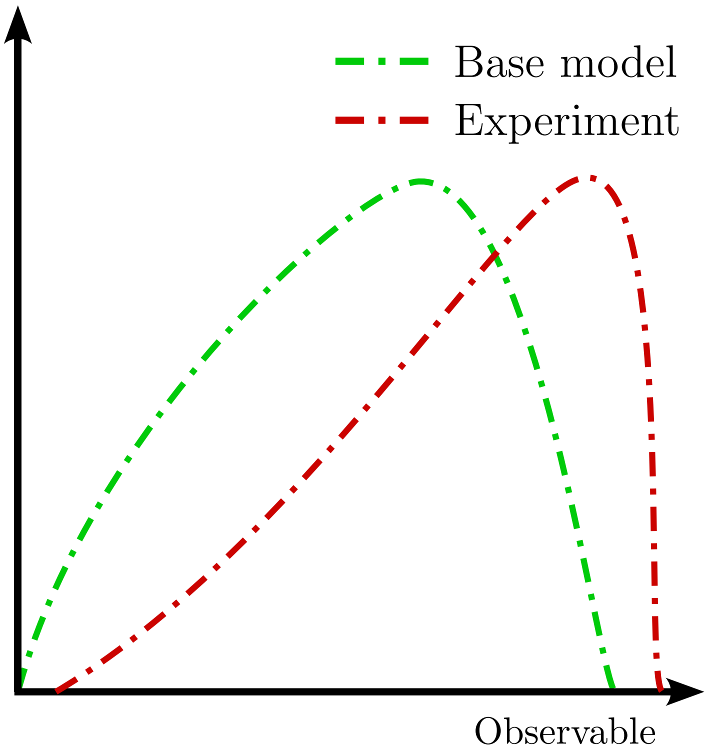




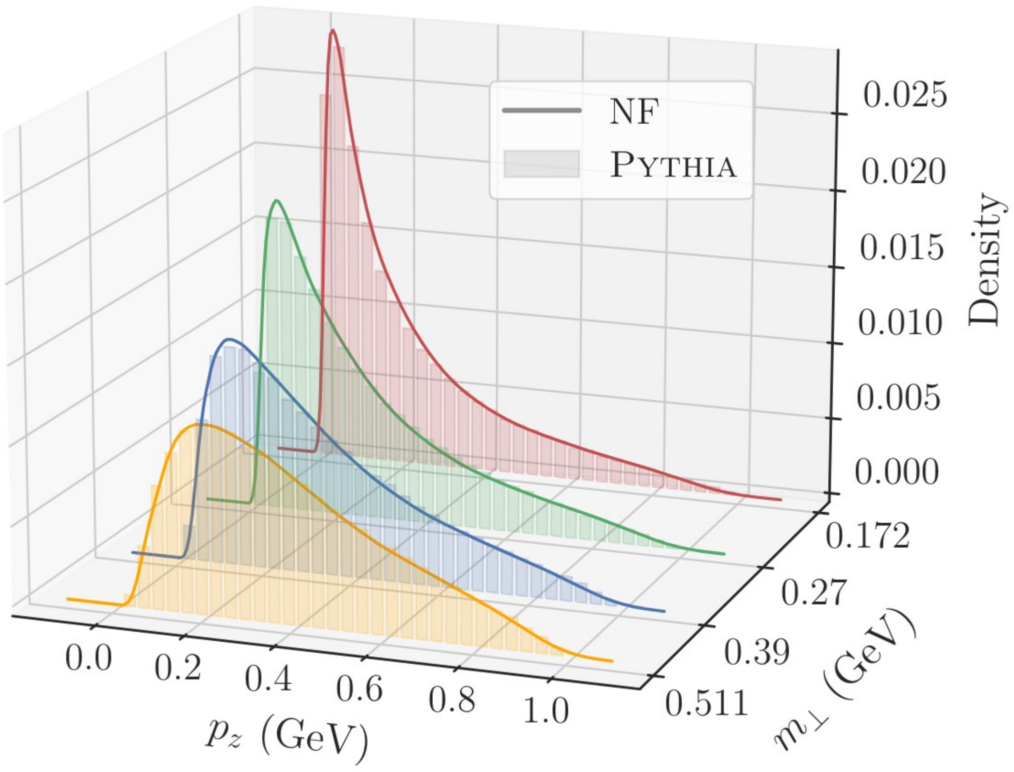


MLHAD efforts: MAGIC

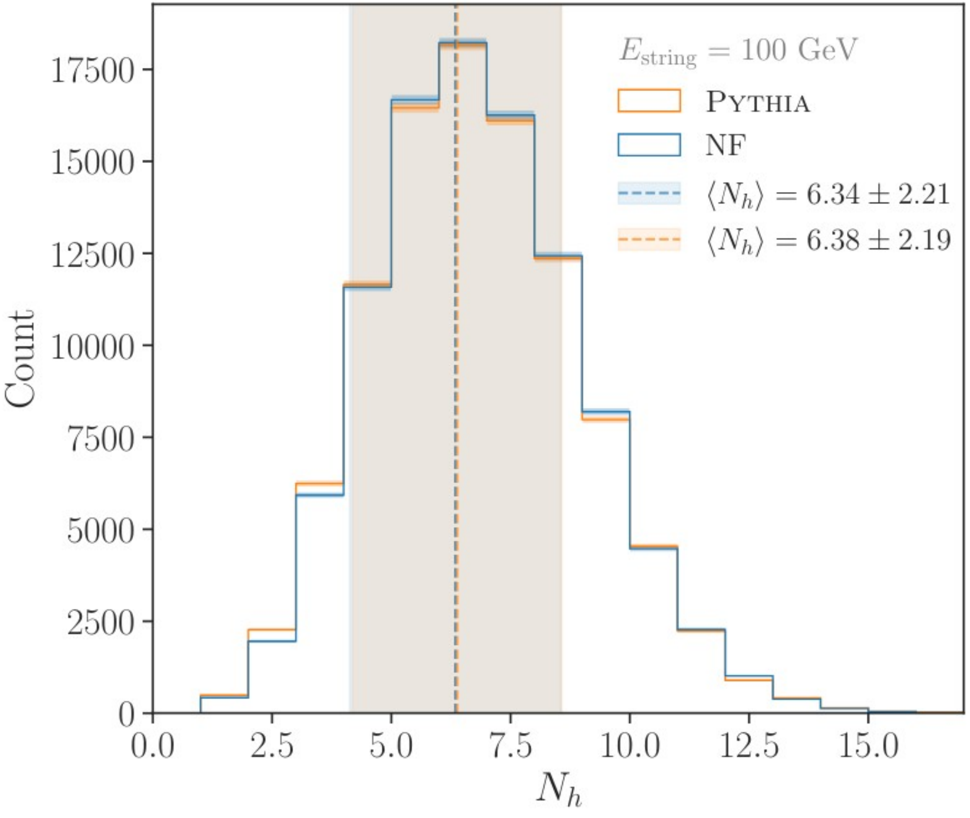


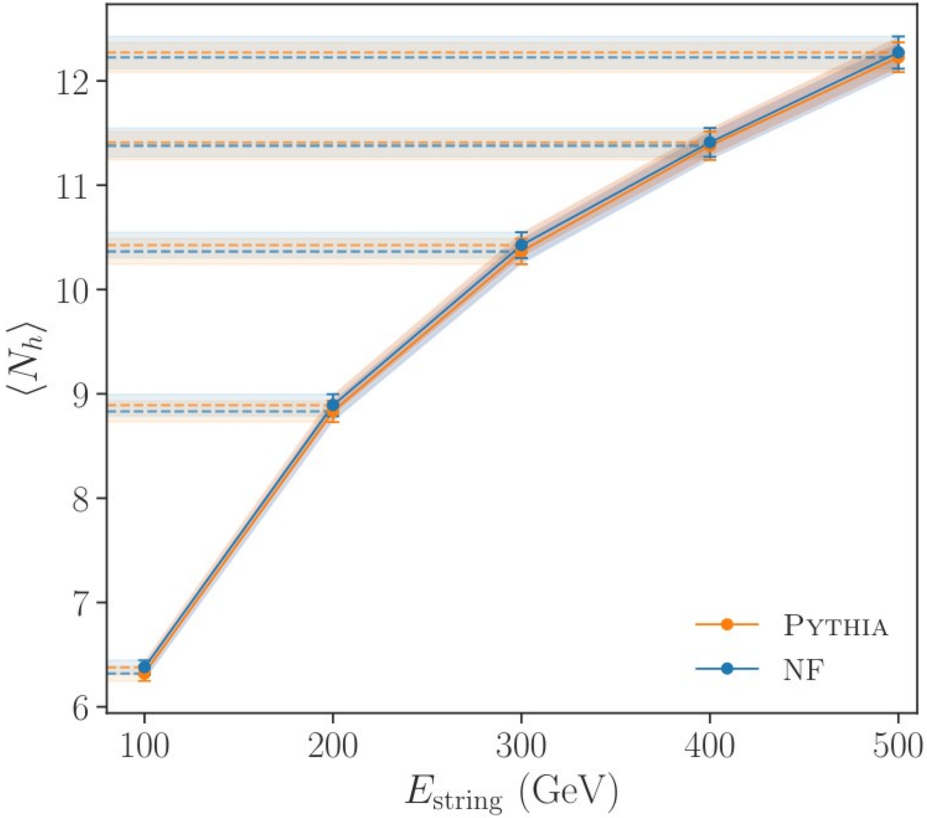


MAGIC: Base model

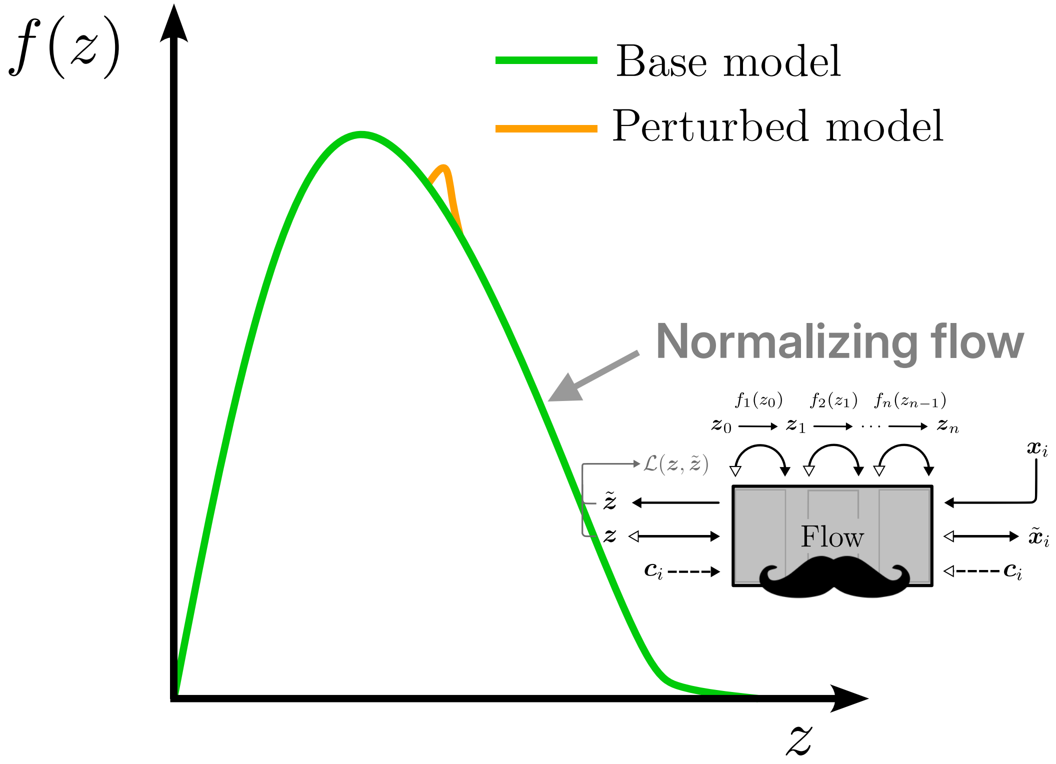


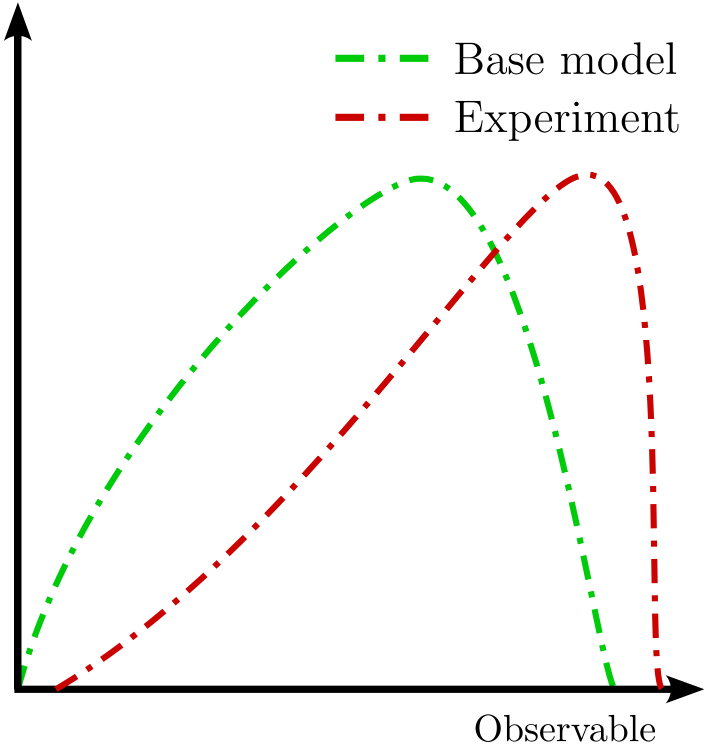
MAGIC: Base model



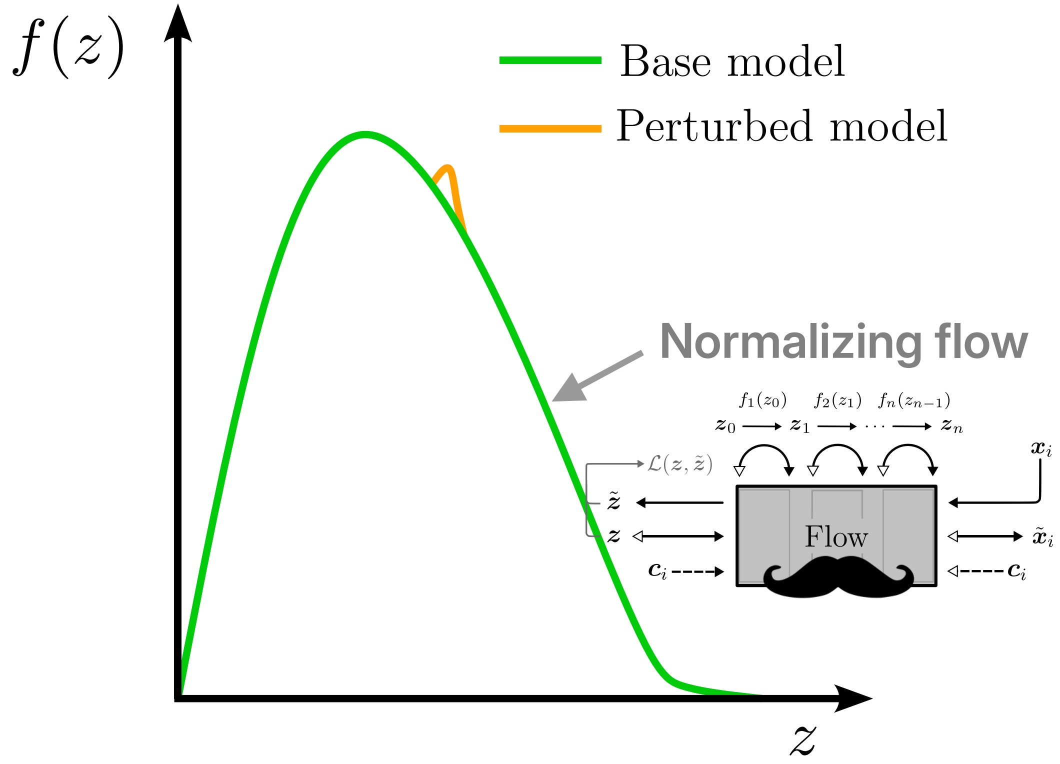


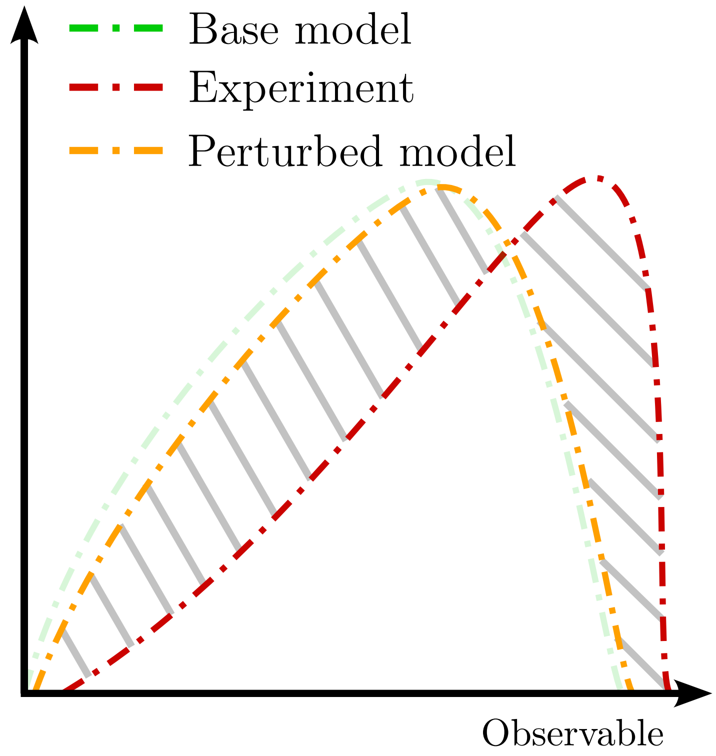
MLHAD efforts: MAGIC



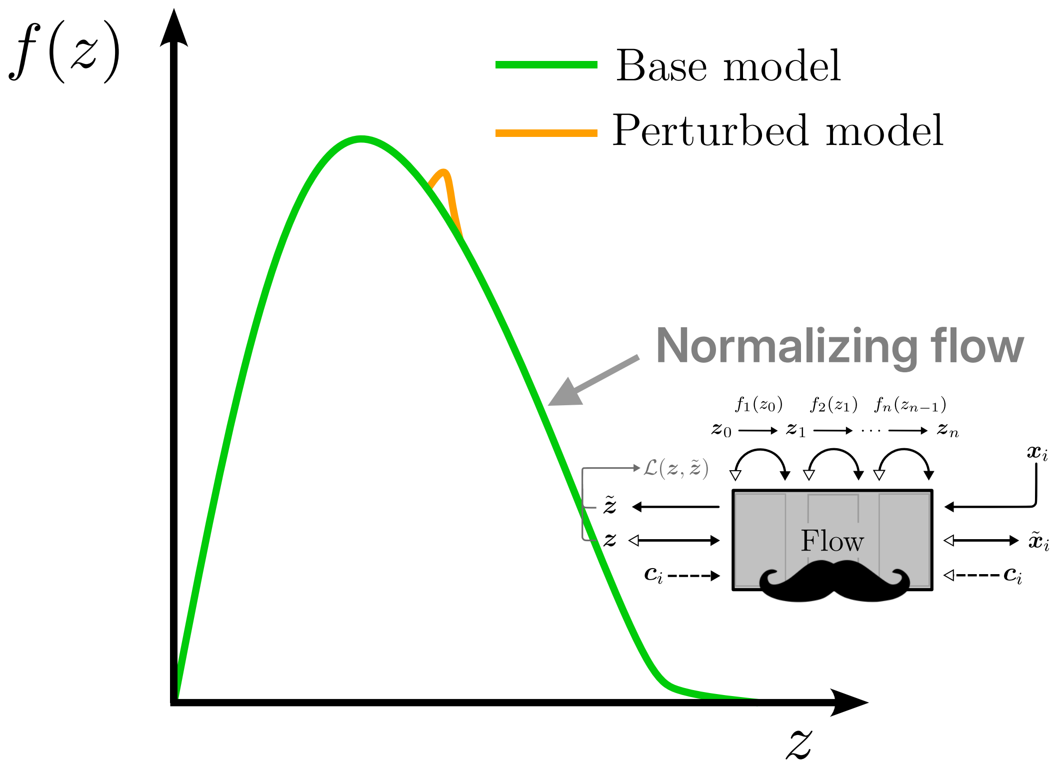


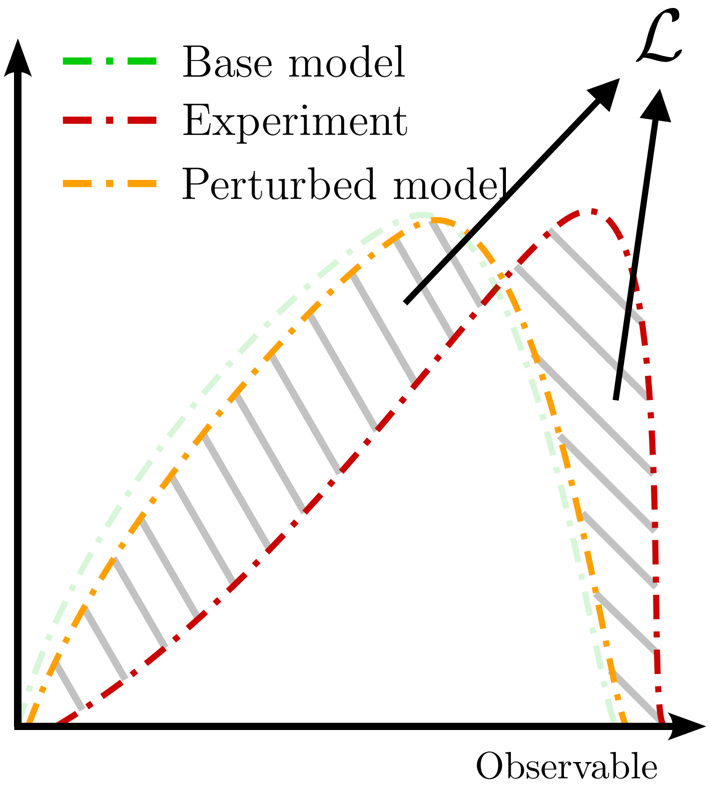
MLHAD efforts: MAGIC



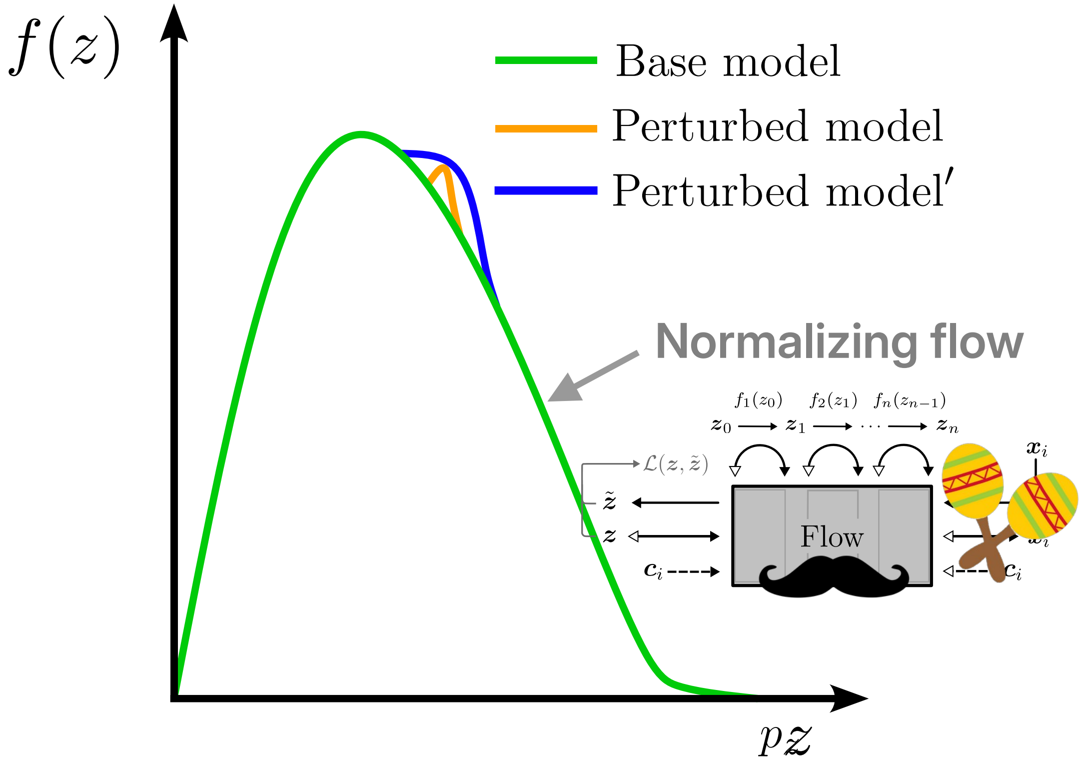


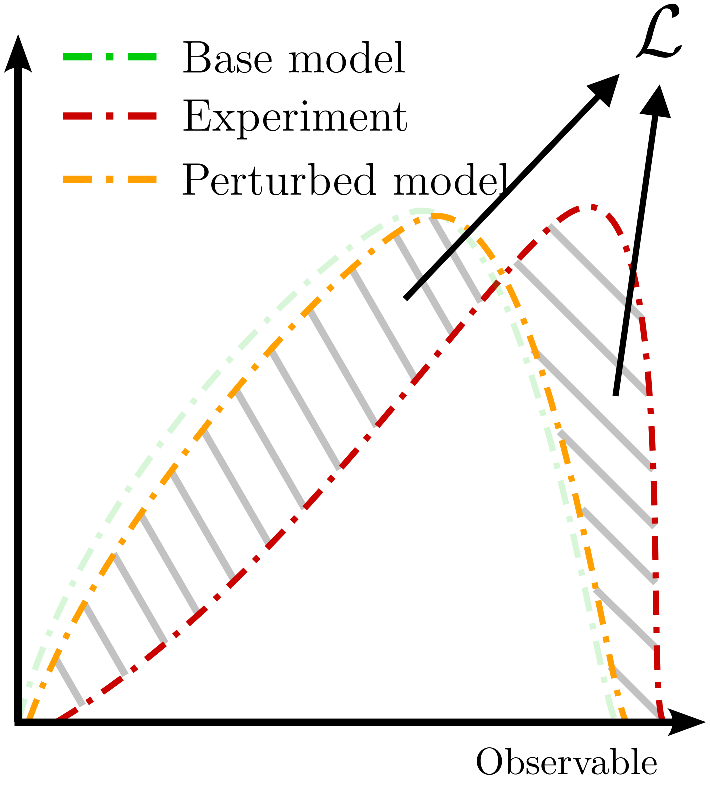
MLHAD efforts: MAGIC



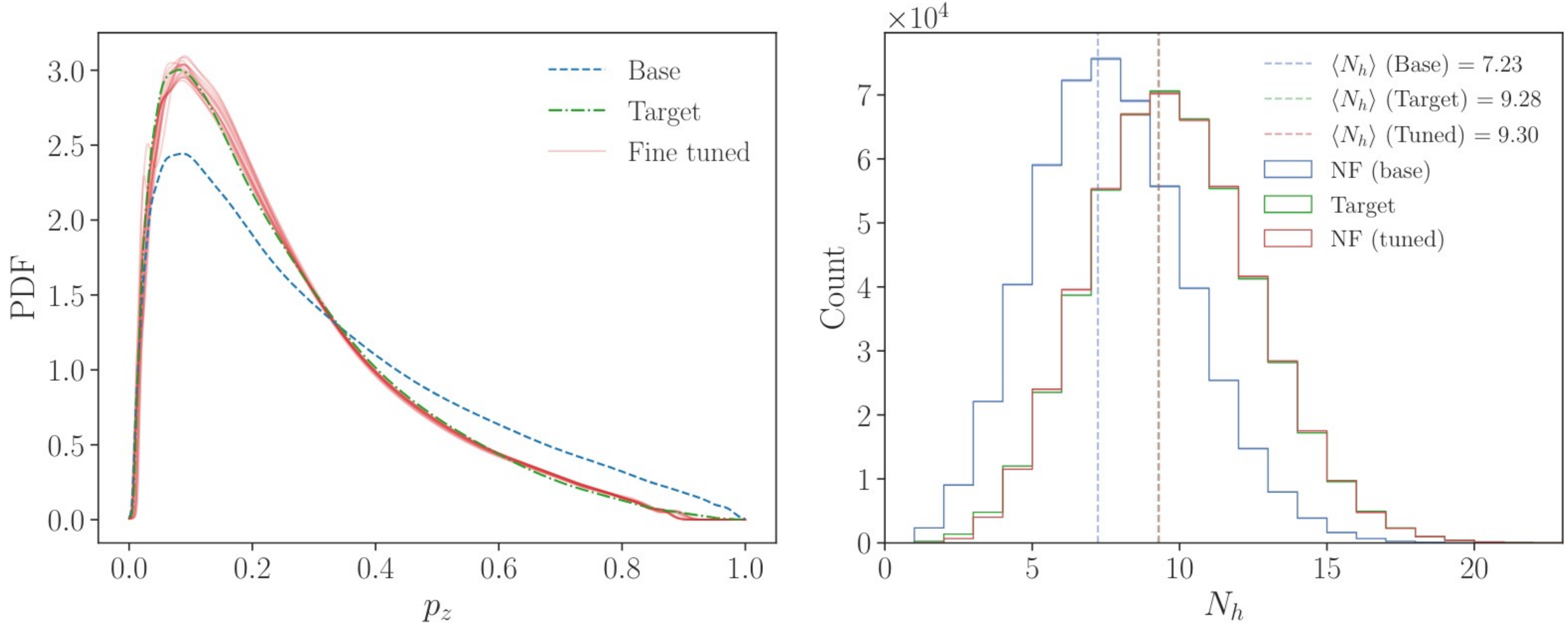


MLHAD efforts: MAGIC





MAGIC: $q\bar{q}$



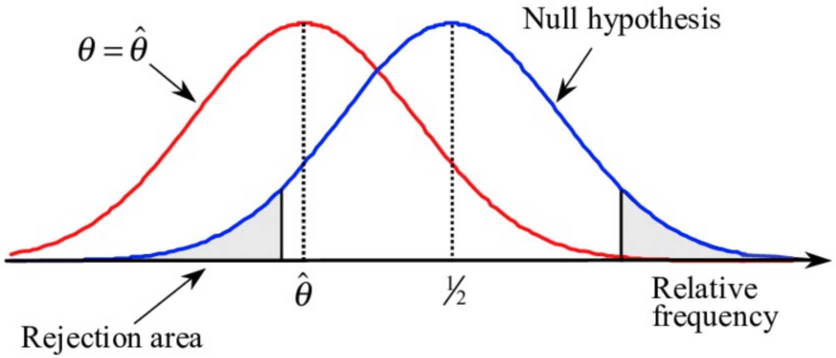
3.0
2.5
2.0
1.5
1.0
0.5
0.0
PDF
Base
Target
Fine tuned
0.0
0.2
0.4
0.6
0.8
1.0
p_z
$\times 10^4$
7
6
5
4
3
2
1
0
Count
$\langle N_h \rangle$ (Base) = 7.23
$\langle N_h \rangle$ (Target) = 9.28
$\langle N_h \rangle$ (Tuned) = 9.30
NF (base)
Target
NF (tuned)
0
5
10
15
20
N_h

HOMER

$$\text{argmin}_f L[f] = \frac{p(x \mid \theta_0)}{p(x \mid \theta_1)} = \mathcal{L}(x)$$

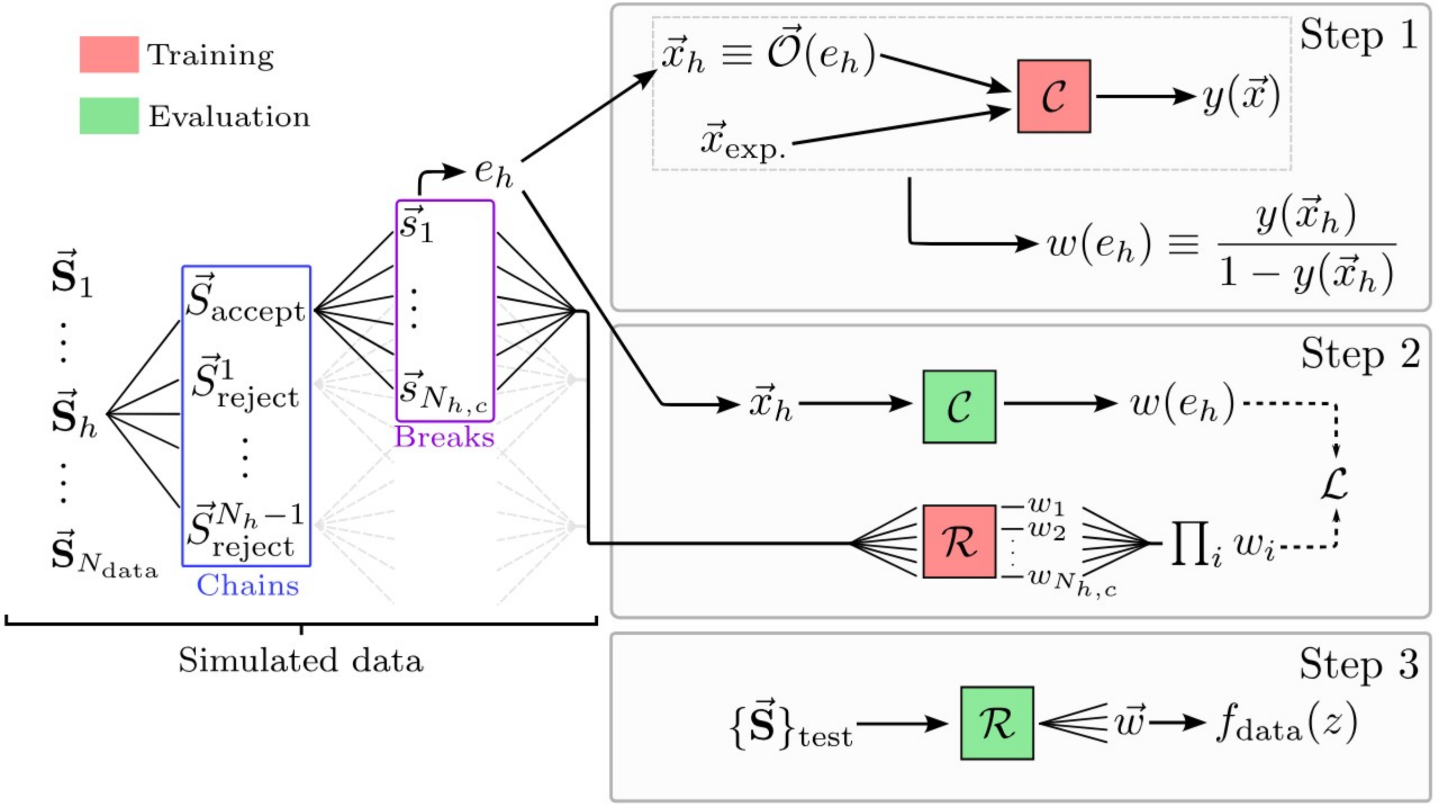
Minimize binary cross-entropy

$$f(x) = \frac{p(x \mid \theta_0)}{p(x \mid \theta_0) + p(x \mid \theta_1)}$$



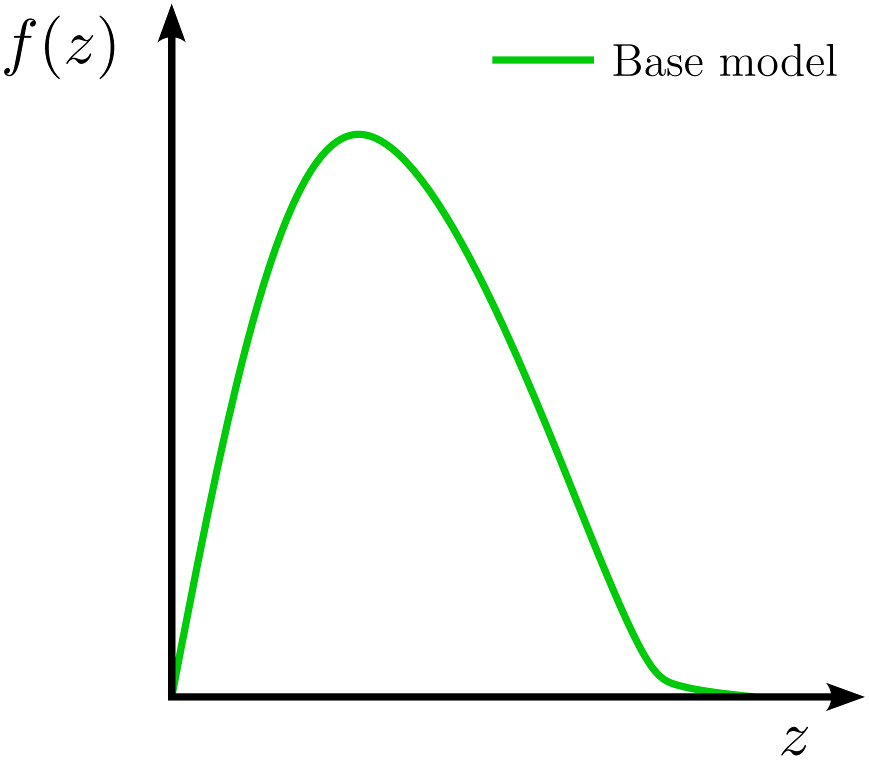
$$\frac{f(x)}{1-f(x)} = \frac{\frac{p(x|\theta_0)}{p(x|\theta_0)+p(x|\theta_1)}}{1-\frac{p(x|\theta_0)}{p(x|\theta_0)+p(x|\theta_1)}} = \frac{p(x \mid \theta_0)}{p(x \mid \theta_0) + p(x \mid \theta_1) - p(x \mid \theta_0)} = \frac{p(x \mid \theta_0)}{p(x \mid \theta_1)} = \mathcal{L}(x)$$

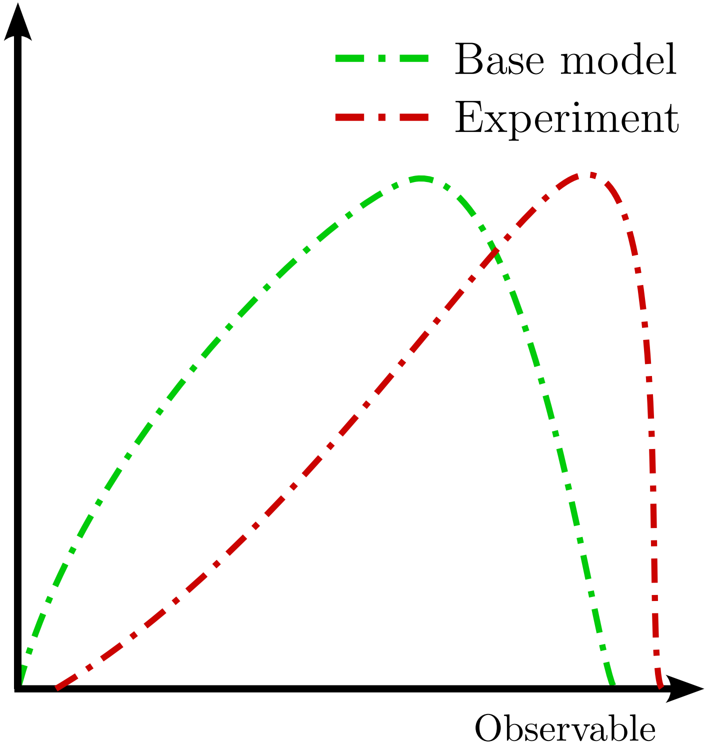
HOMER: $q\bar{q}$



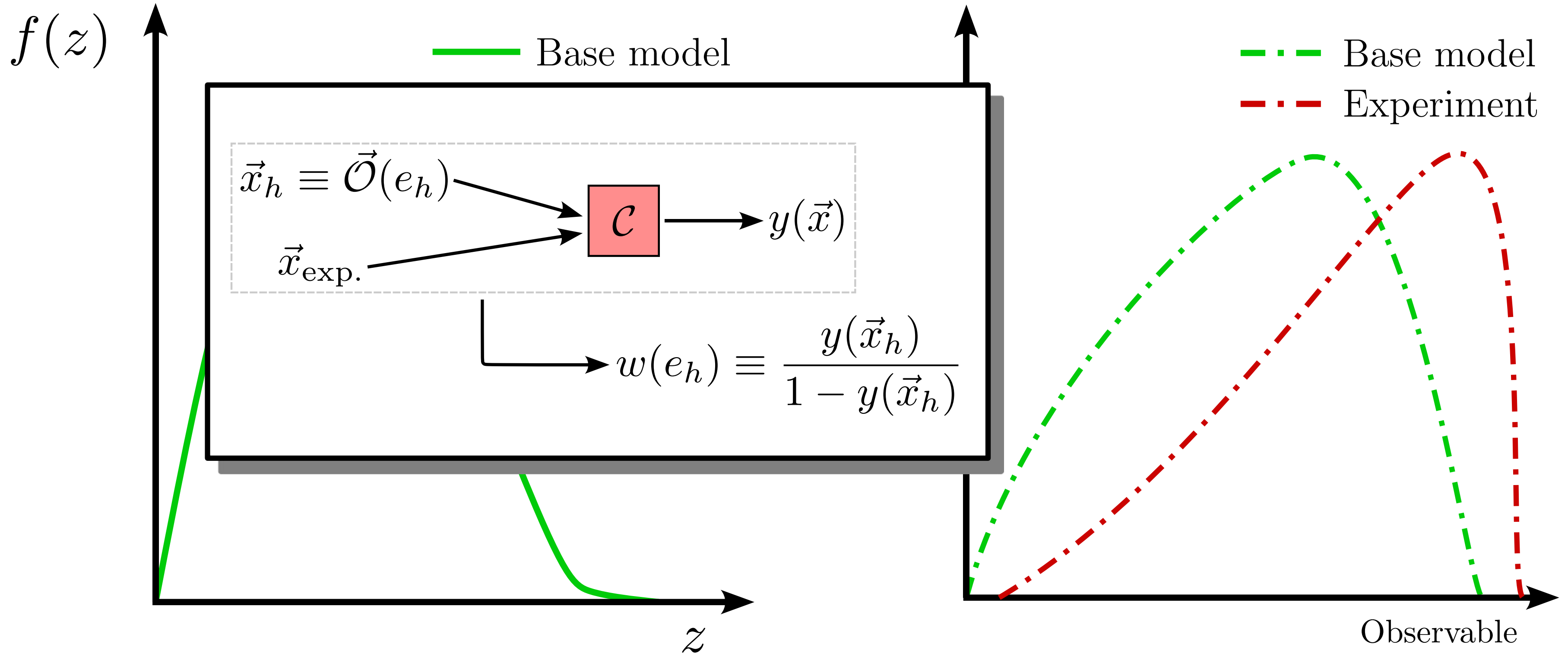
Training
Evaluation
$\vec{S}_1$
$\vec{S}_h$
$\vec{S}_{N_{data}}$
$\vec{S}_{accept}$
$\vec{S}^1_{reject}$
$\vec{S}^{N_h-1}_{reject}$
Chains
$\vec{s}_1$
$\vec{s}_{N_{h,c}}$
Breaks
e_h
Simulated data
Step 1
$\vec{x}_h \equiv \vec{\mathcal{O}}(e_h)$
$\vec{x}_{exp.}$
$\mathcal{C}$
$y(\vec{x})$
$w(e_h) \equiv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

MLHAD efforts: HOMER

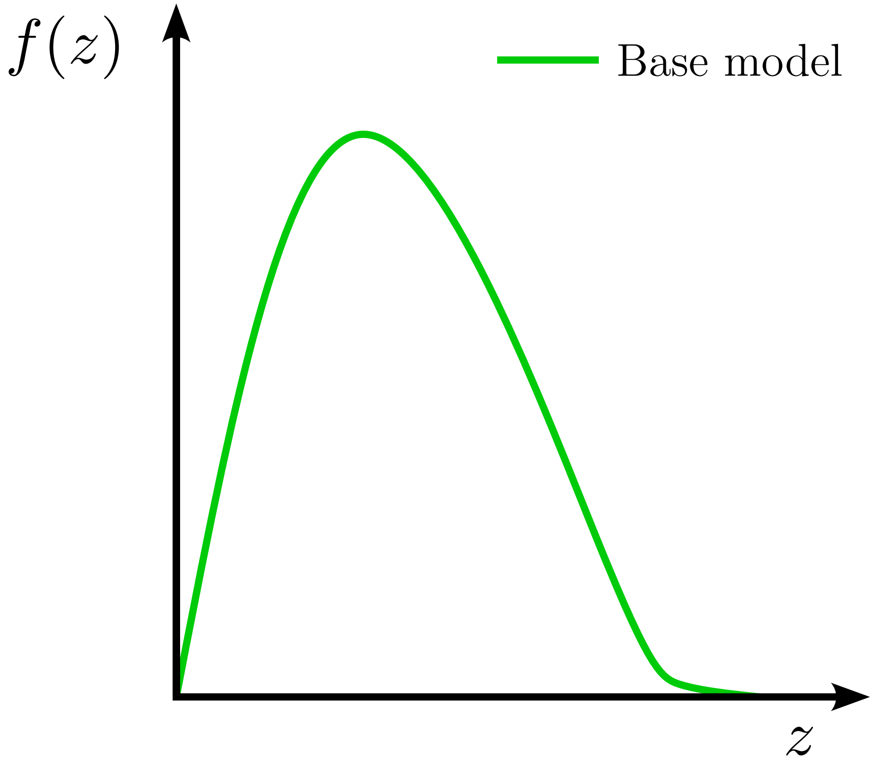


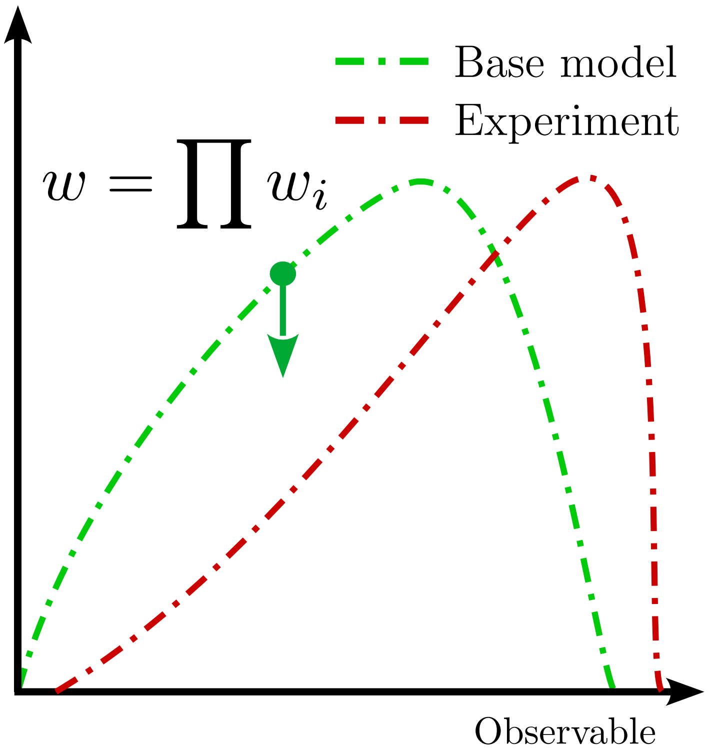


MLHAD efforts: HOMER

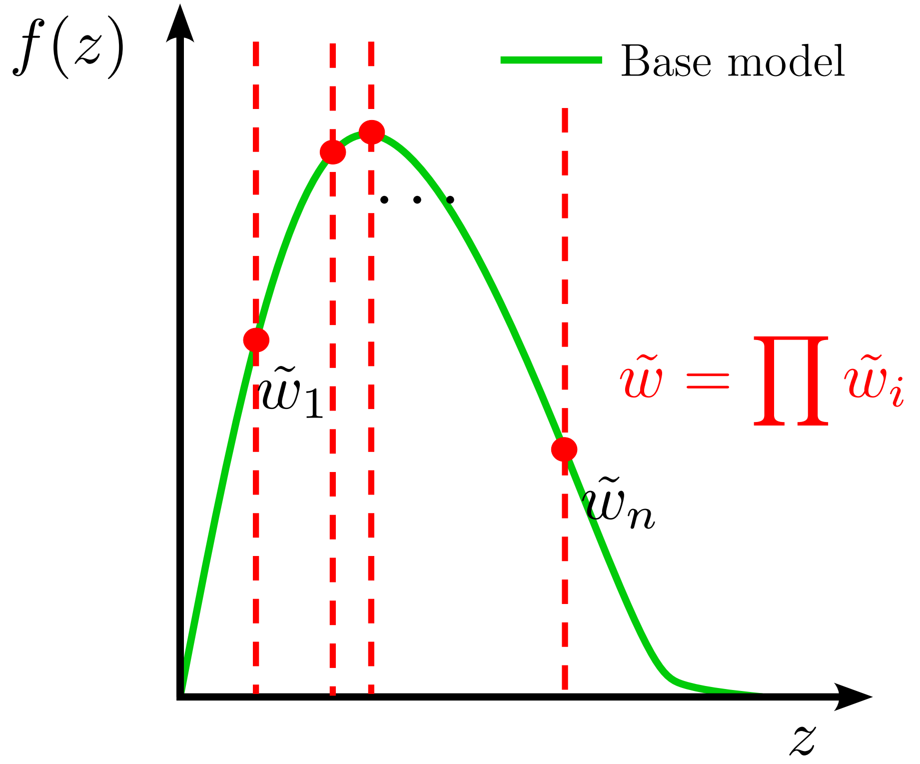


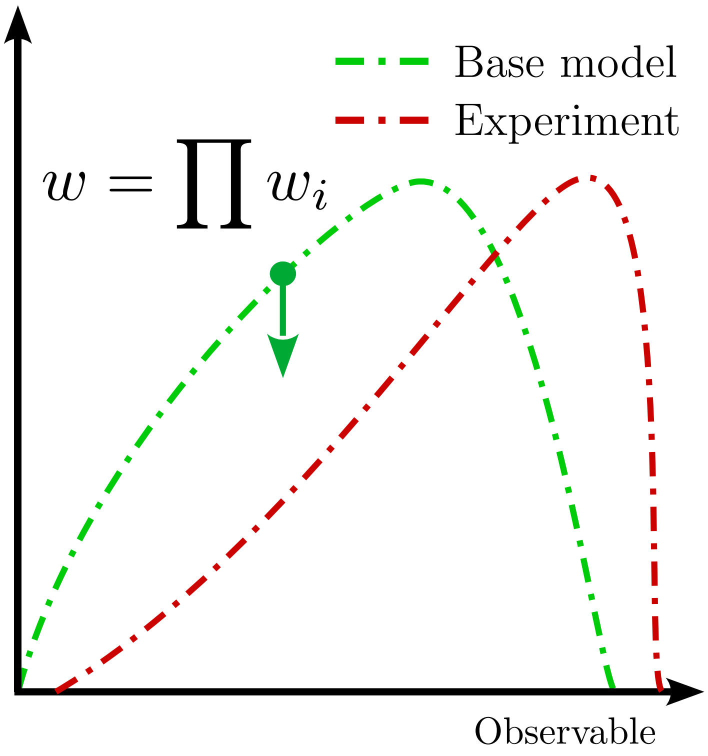
MLHAD efforts: HOMER



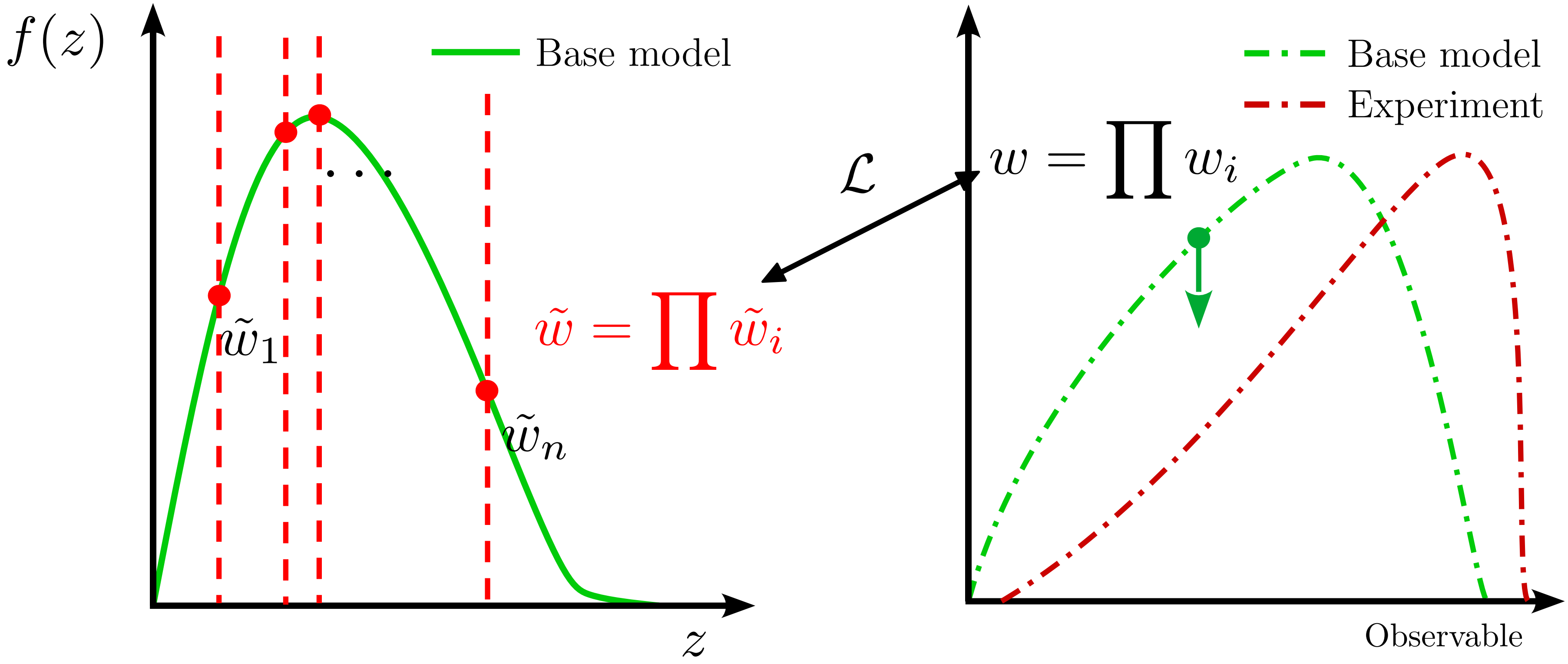


MLHAD efforts: HOMER

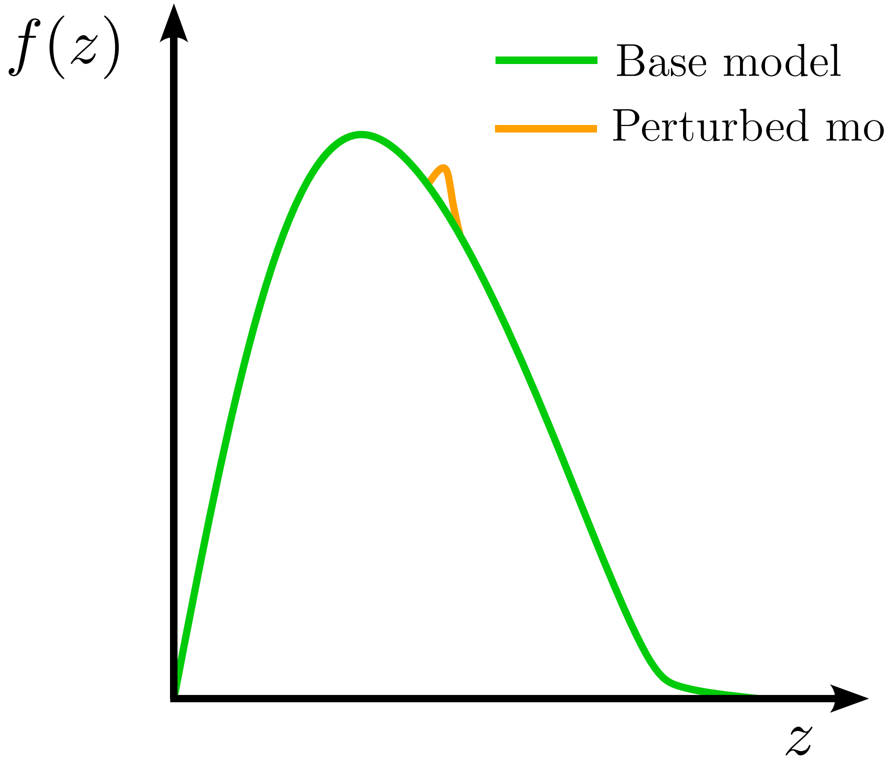


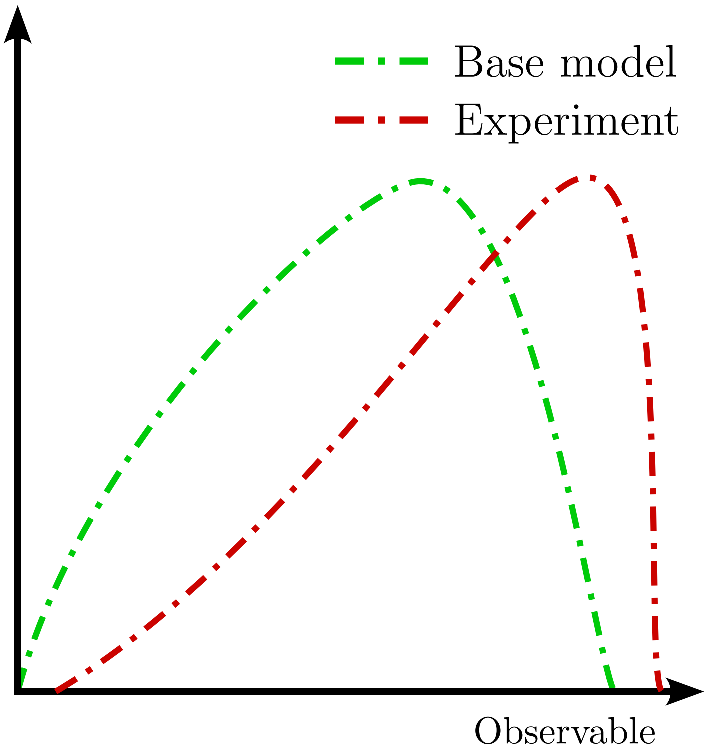


MLHAD efforts: HOMER

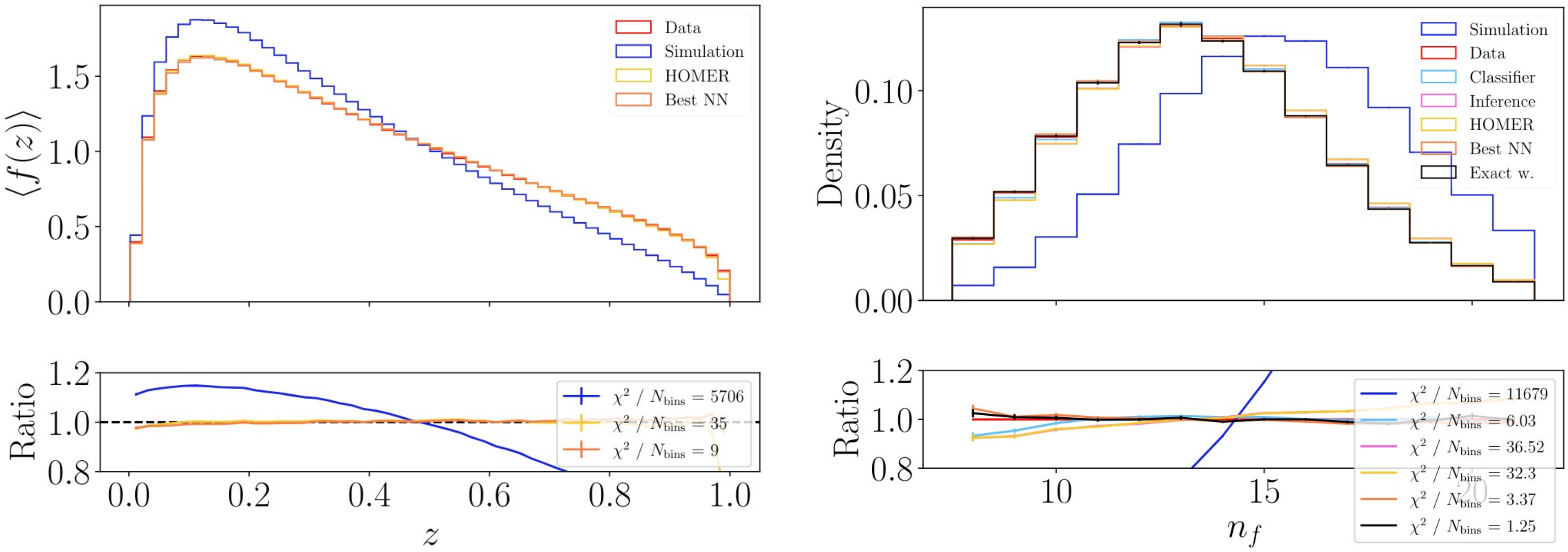


MLHAD efforts: HOMER

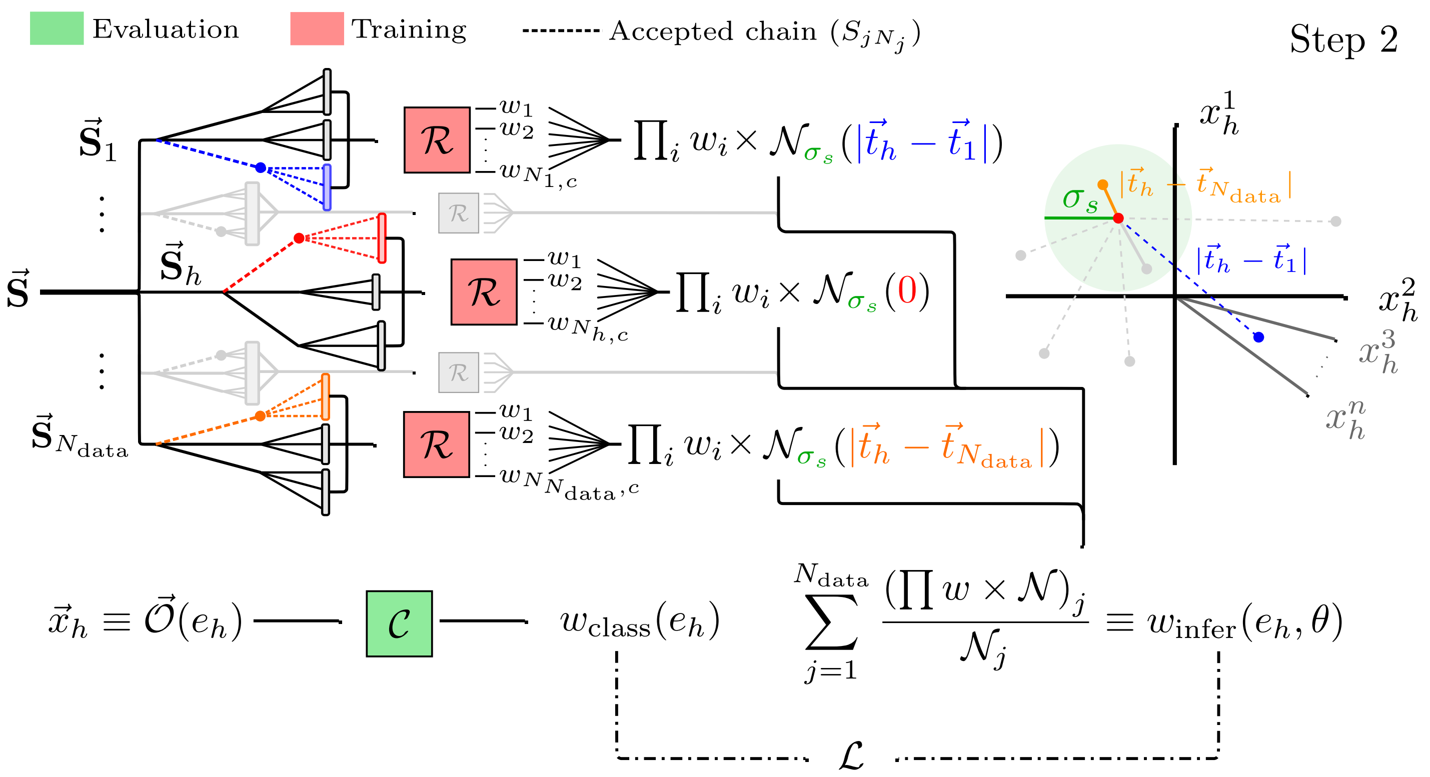




HOMER: $q\bar{q}$


Data
Simulation
HOMER
Best NN
$\langle f(z) \rangle$
Ratio
χ^2 / N_{bins} = 5706
χ^2 / N_{bins} = 35
χ^2 / N_{bins} = 9
z
Simulation
Data
Classifier
Inference
HOMER
Best NN
Exact w.
Density
Ratio
χ^2 / N_{bins} = 11679
χ^2 / N_{bins} = 6.03
χ^2 / N_{bins} = 36.52
χ^2 / N_{bins} = 32.3
χ^2 / N_{bins} = 3.37
χ^2 / N_{bins} = 1.25
n_f

HOMER: w/ gluons



Two “solutions” to the inverse problem of hadronization

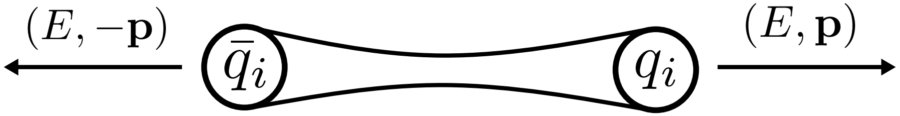
MAGIC

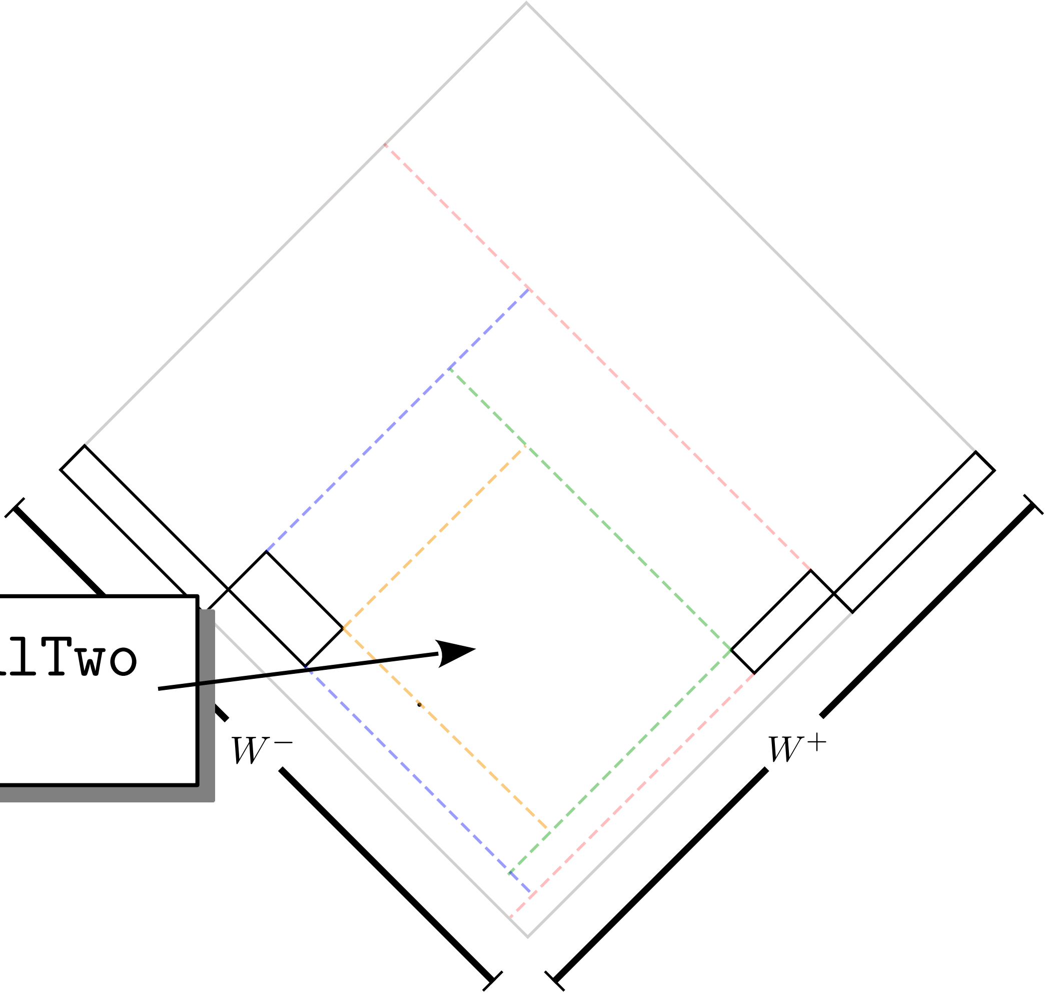
- “Top-down”
- Relies on the construction of an explicit likelihood (normalizing flow)
- Must be embedded into the simulation pipeline

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

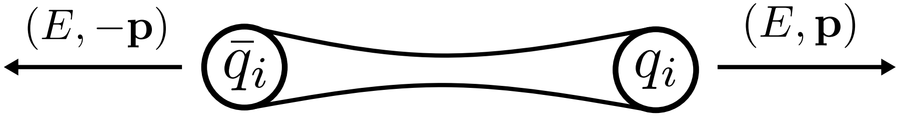
The road ahead





When E_{CM} goes below E_{cut}, `finalTwo` is called.

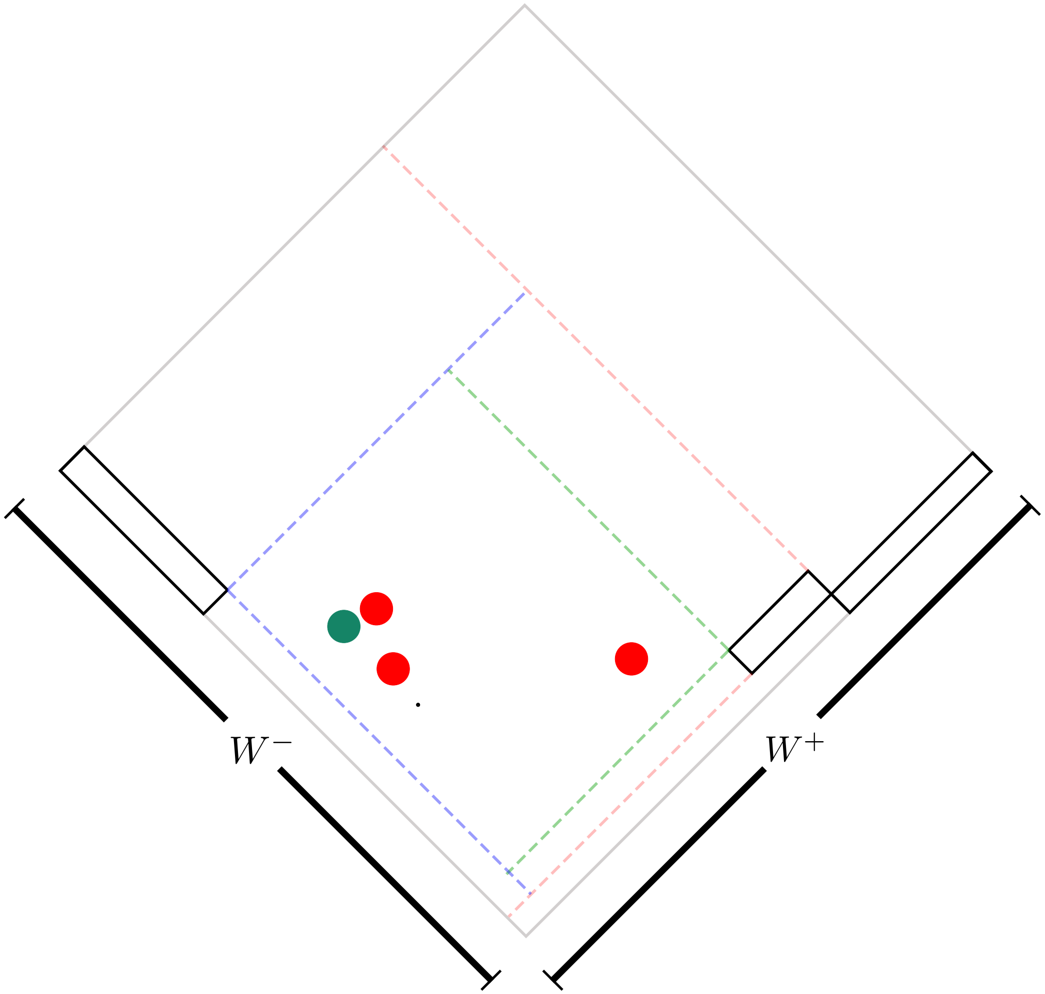
The road ahead



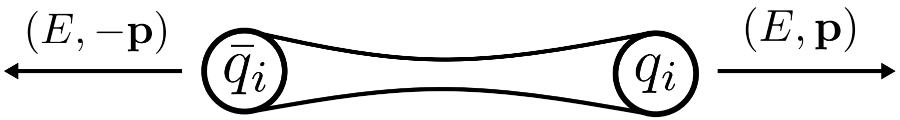
This system has two limits:

1. $E_{CM} >> m_h$
   - Random walk in f(z)

2. $E_{CM} \sim m_h$



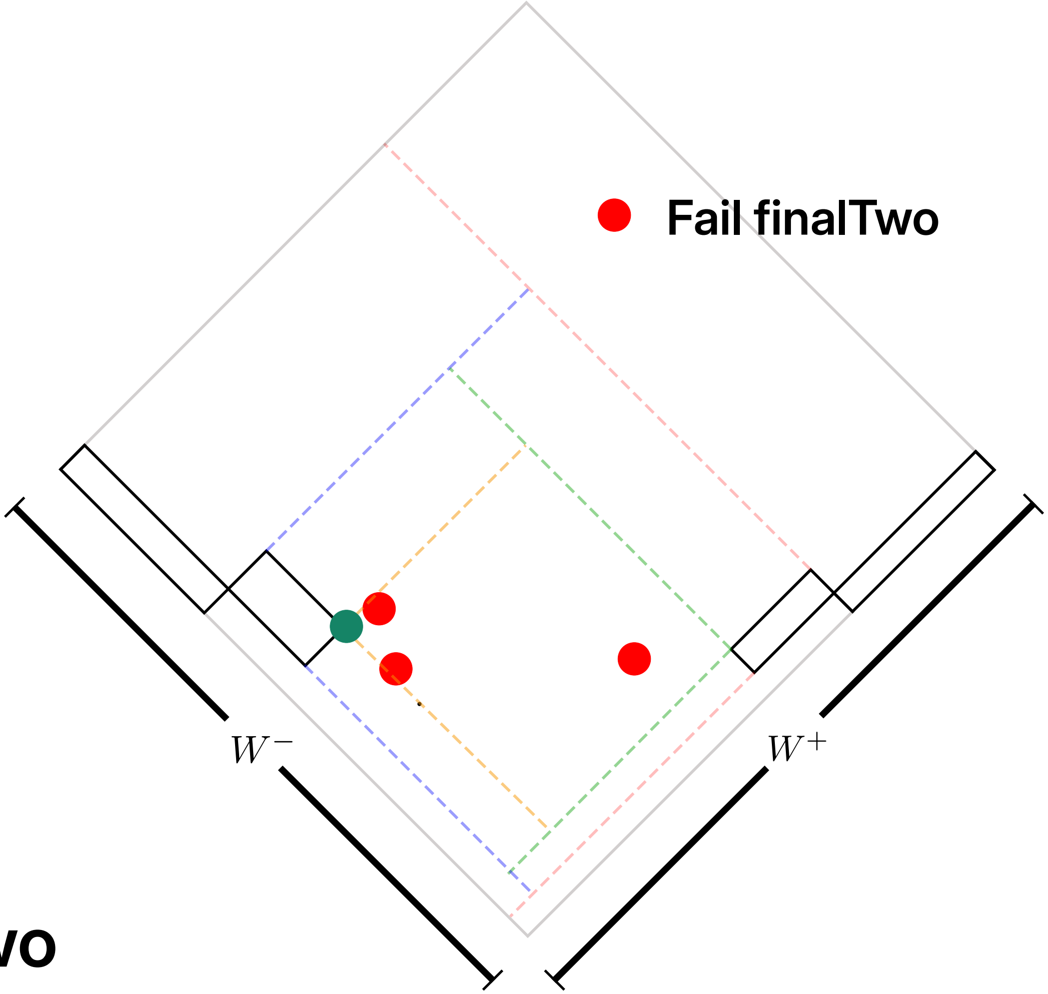
The road ahead



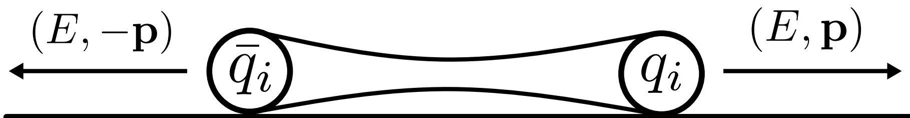
This system has two limits:

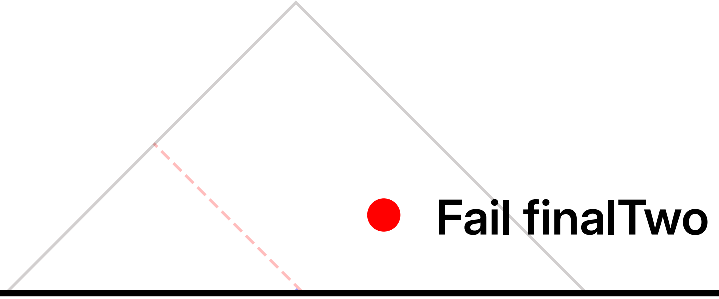
1. $E_{CM} >> m_h$
   - Random walk in f(z)

2. $E_{CM} \sim m_h$
   - f′(z) influenced by finalTwo



The road ahead

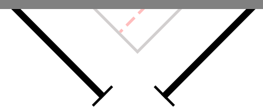




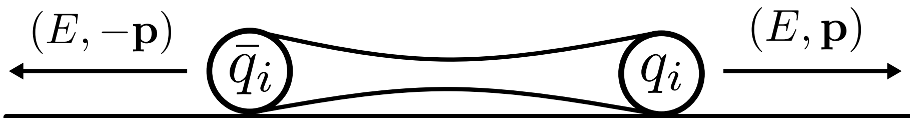
We want to extract f(z), not f′(z) – cleaner extraction would factorize these two pieces.

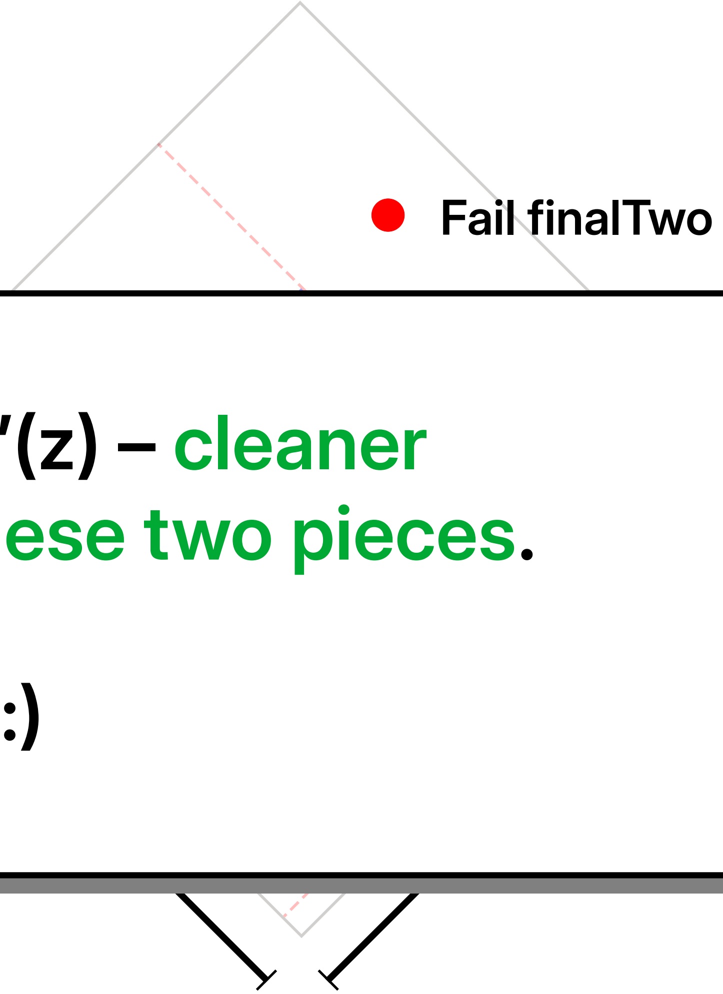
How?

- f′(z) influenced by finalTwo



The road ahead





We want to extract f(z), not f′(z) – cleaner extraction would factorize these two pieces.

How? Maybe another time… :)

- f′(z) influenced by finalTwo

Conclusions

- RSA: flexible, scalable, replicable approach to tuning
  - ML-based observables, optimizers, infrastructure
  - Uncertainty quantification, scaling with parameter dimension
- Data-driven fragmentation function:
  - HOMER and MAGIC offer as complementary methods
    - HOMER + flavor, tackle real data soon?
    - Better observables needed
    - HOMER x MAGIC
    - Better likelihood models? (MAGIC^{+})

MLHAD

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

Conclusions

- RSA: flexible, scalable, replicable approach to tuning
  - ML-based observables, optimizers, infrastructure
  - Uncertainty quantification, scaling with parameter dimension
- Data-driven fragmentation function:
  - HOMER and MAGIC offer as complementary methods
    - HOMER + flavor, tackle real data soon?
    - Better observables needed
    - HOMER x MAGIC
    - Better likelihood models? (MAGIC⁺)

MLHAD

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

Thank you :)

Back-up

Stringy hadronization: overview

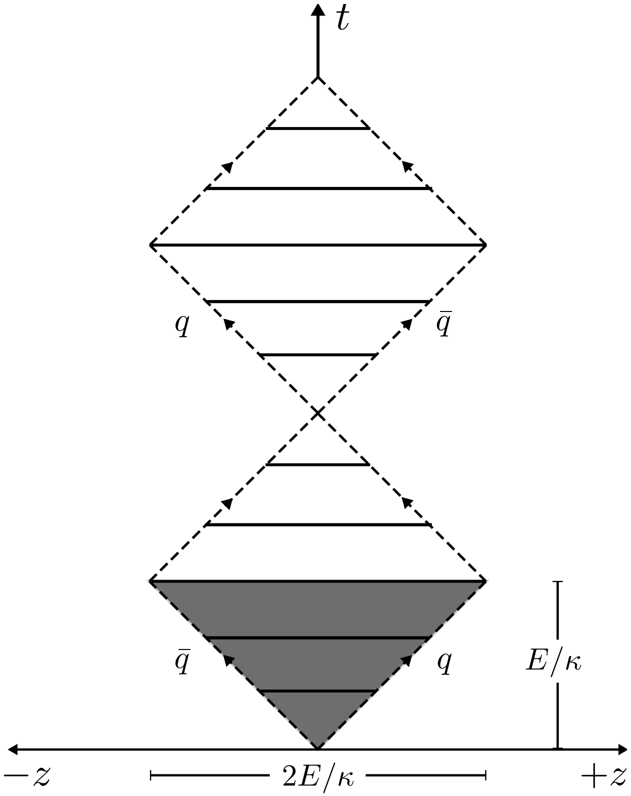
Consider the simplest hadronizing system:

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

Treat this as a semi-classical system with potential

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

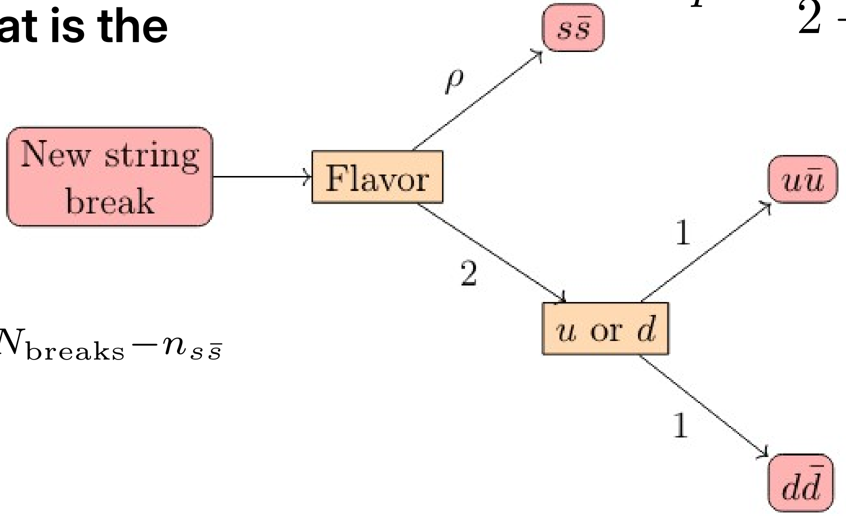
A color flux tube forms between the $q\bar{q}$ pair and in the absence of string breaks the system follows a 'yo-yo' motion



Flavor reweighting (2505.00142)

Consider N_{breaks} string breaks. If strange breaks are suppressed by ρ compared to light quarks, what is the probability to producing n_{ss} breaks?

$$p = \frac{\rho}{2+\rho}$$



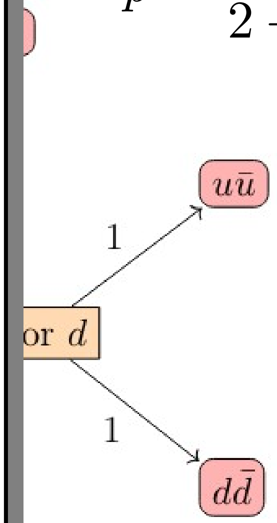
$$P(n_{s\bar{s}}|N_{\text{breaks}}) = \binom{N_{\text{breaks}}}{n_{s\bar{s}}} p^{n_{s\bar{s}}}(1-p)^{N_{\text{breaks}}-n_{s\bar{s}}}$$

$$w = \frac{P(\rho')}{P(\rho)} = \left(\frac{\rho'}{\rho}\right)^{n_{s\bar{s}}} \left(\frac{1-\rho'}{1-\rho}\right)^{N_{\text{breaks}}-n_{s\bar{s}}}$$

Flavor reweighting (2505.00142)

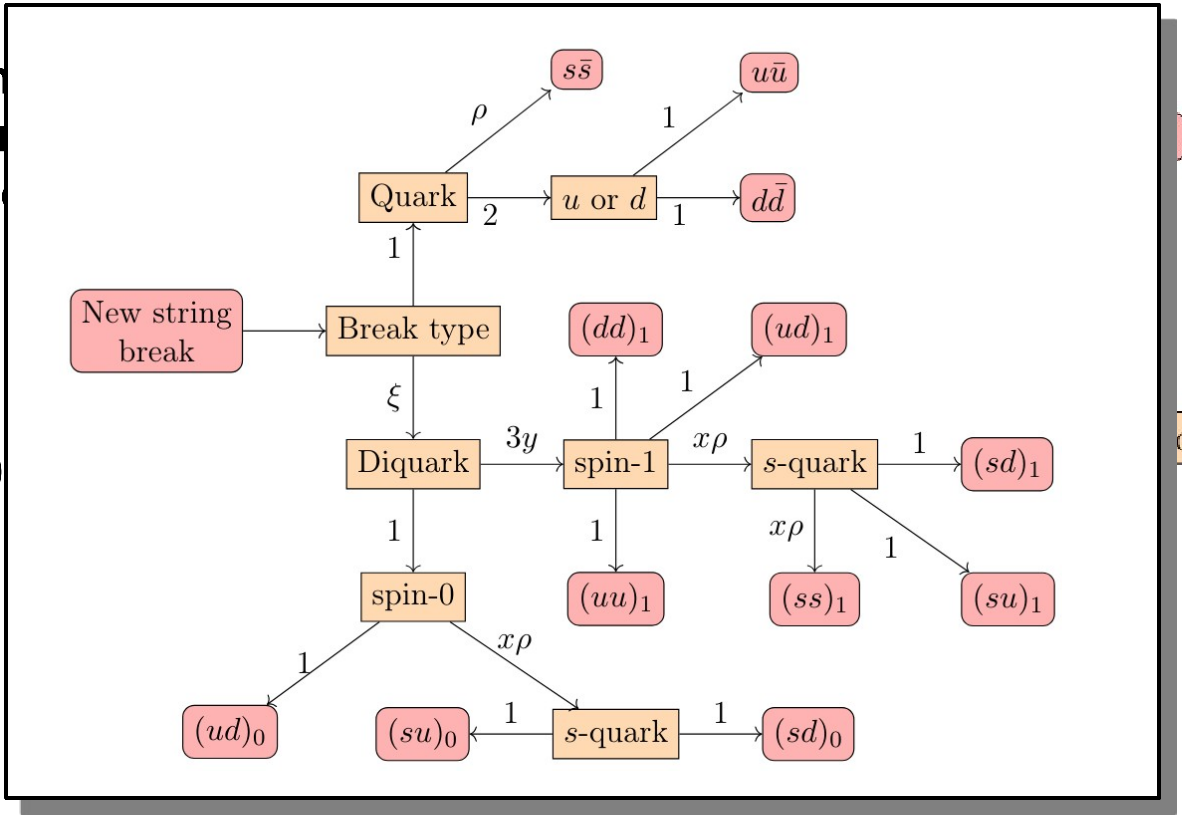
Consider N_{breaks} strin
suppressed by ρ co
probability to produ

$$p = \frac{\rho}{2+\rho}$$



$$P(n_{s\bar{s}}|N_{\text{breaks}})$$

$$w = \frac{P(\rho')}{P(\rho)}$$



MAGIC: Invertible nerual networks (INN)

a.k.a normalizing flow

Invertible Real NVP transformations:

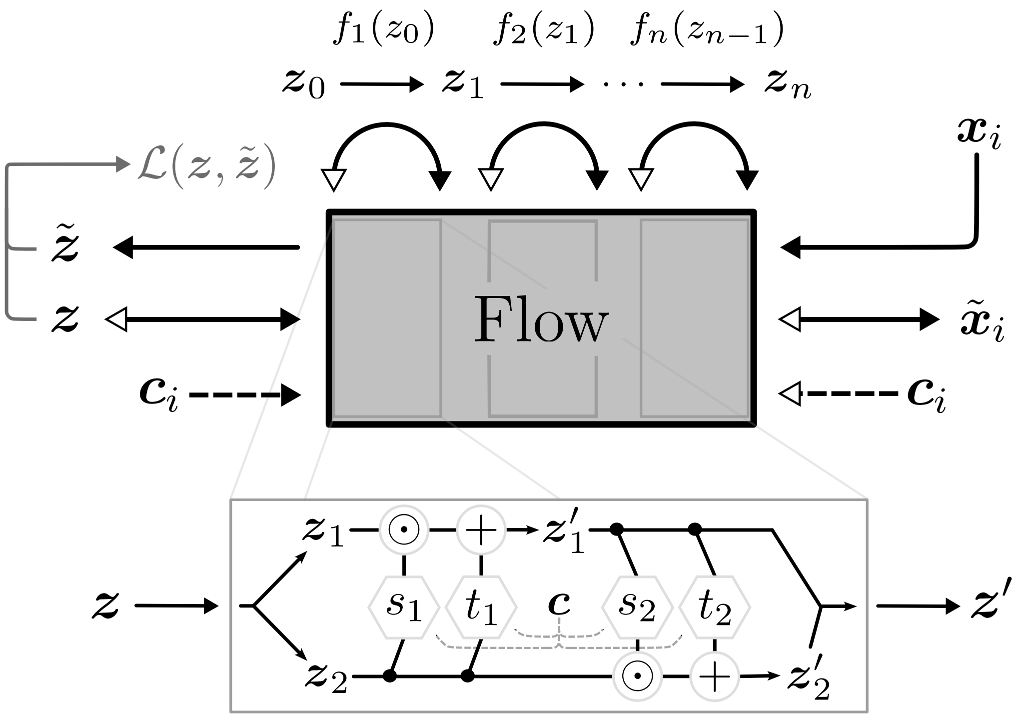
$$z'_1 = z_1 \odot \exp\big(s_1(z_2)\big) + t_1(z_2),$$
$$z'_2 = z_2 \odot \exp\big(s_2(z'_1)\big) + t_2(z'_1),$$

Scale transform

Translation transform

Inverse:

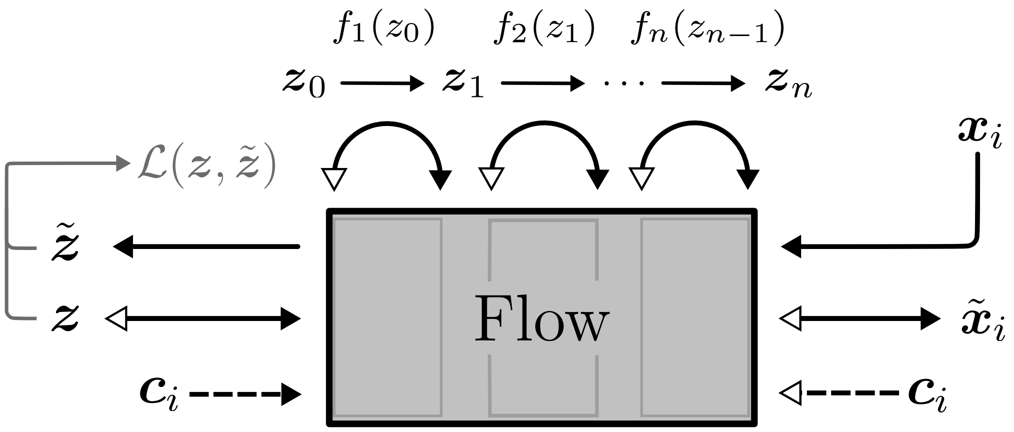
$$z_2 = \big(z'_2 - t(z'_1)\big) \odot \exp\big(-s(z'_1)\big),$$
$$z_1 = \big(z'_1 - t(z_2)\big) \odot \exp\big(-s(z_2)\big),$$



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)

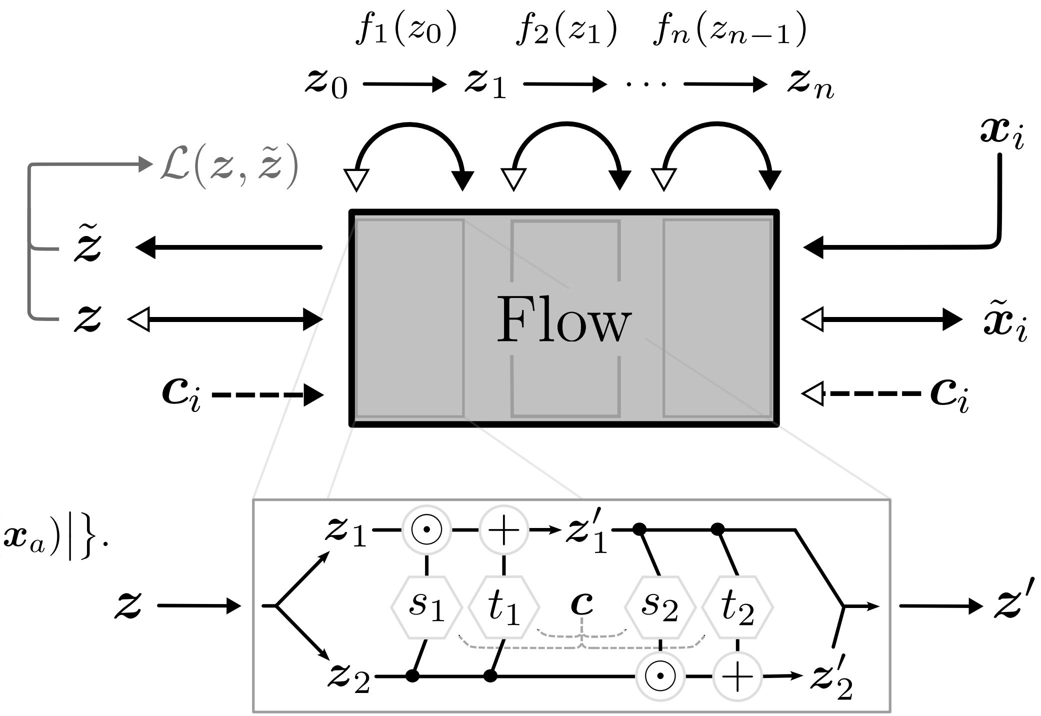
a.k.a normalizing flow

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

Train with the negative log likelihood:

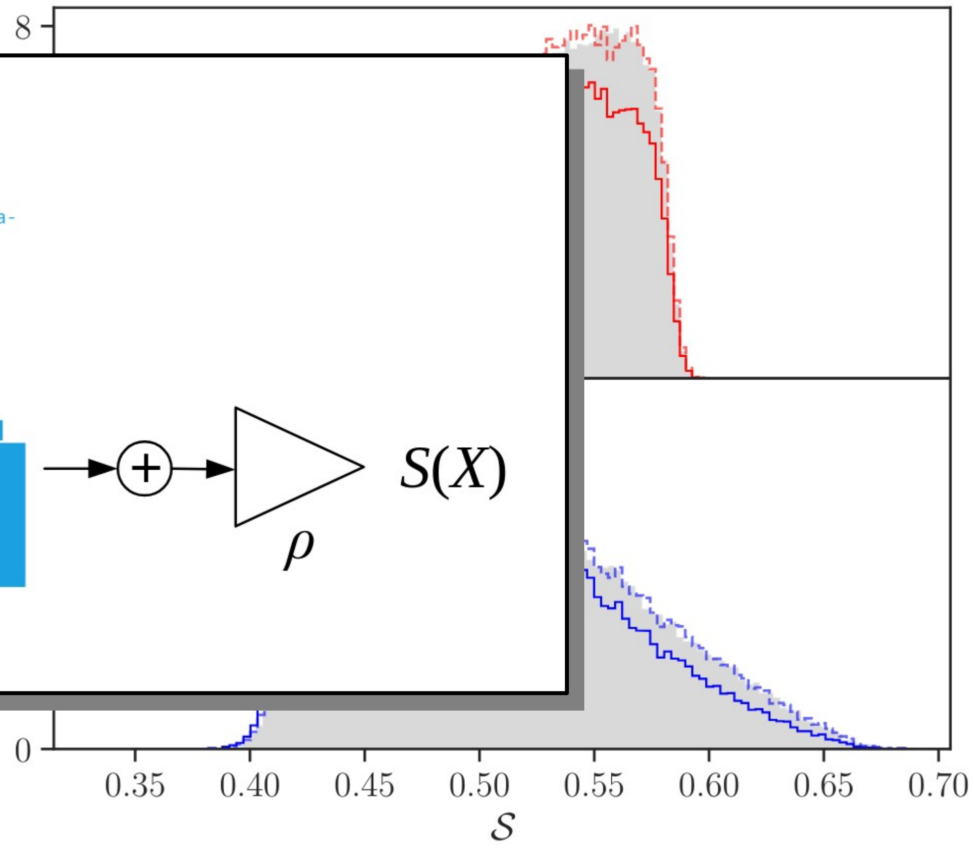
$$\mathcal{L}_{\mathrm{NF}} = -\sum_{a=1}^{N} \log p_X^F(\boldsymbol{x}_a; \boldsymbol{\theta}, \boldsymbol{c}_a)$$

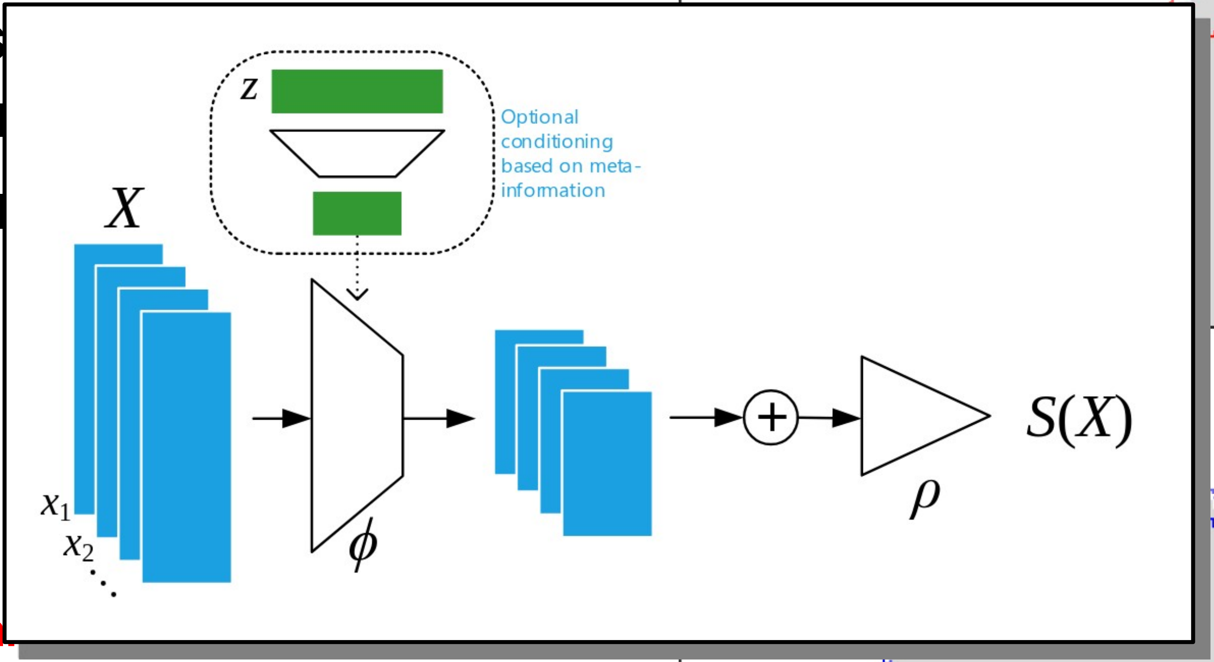
$$= \sum_{a=1}^{N} \left\{ -\log p_Z\left(F^{-1}(\boldsymbol{x}_a; \boldsymbol{\theta}, \boldsymbol{c}_a)\right) + \log \left| \det J_{F^{-1}}(\boldsymbol{x}_a) \right| \right\}.$$



ML-based observables

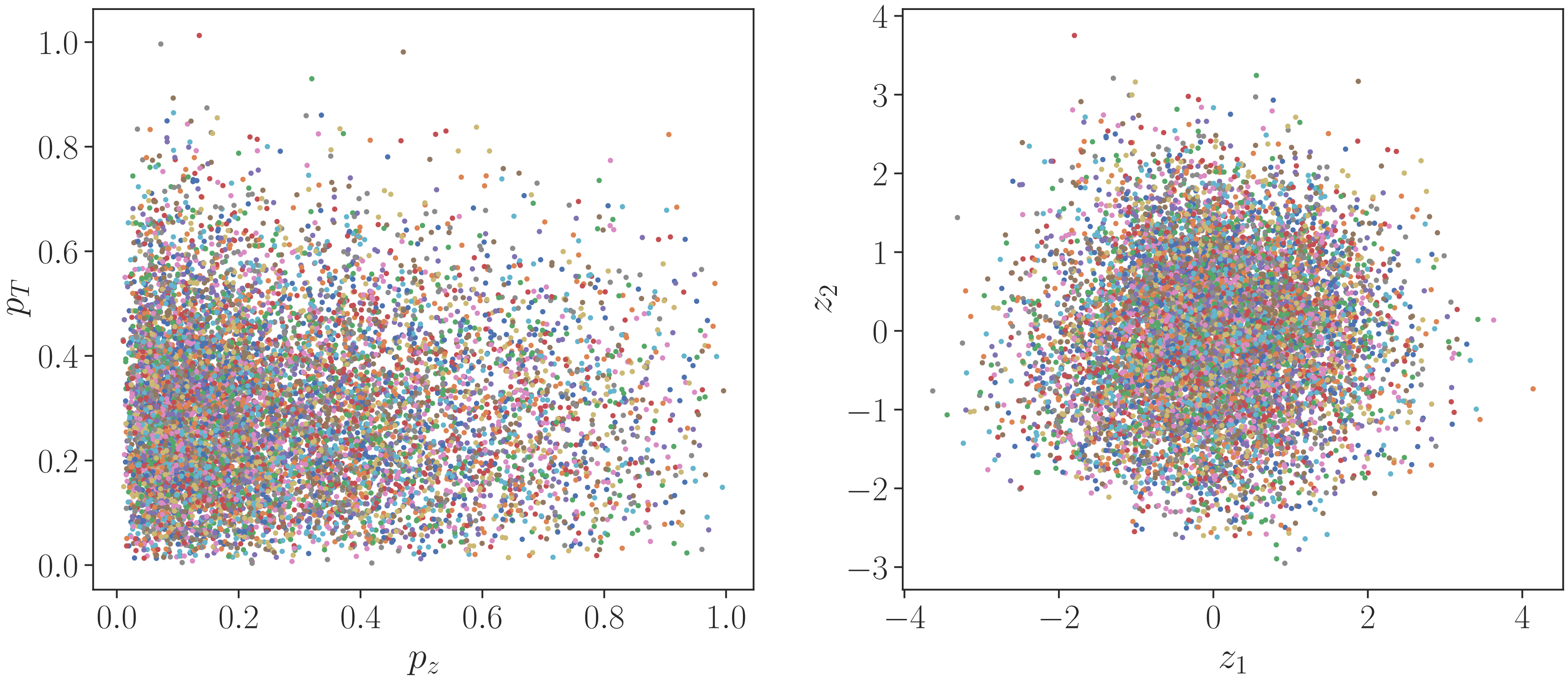
- Train a deepsets classifier to distinguis experime
  - Takes ful input



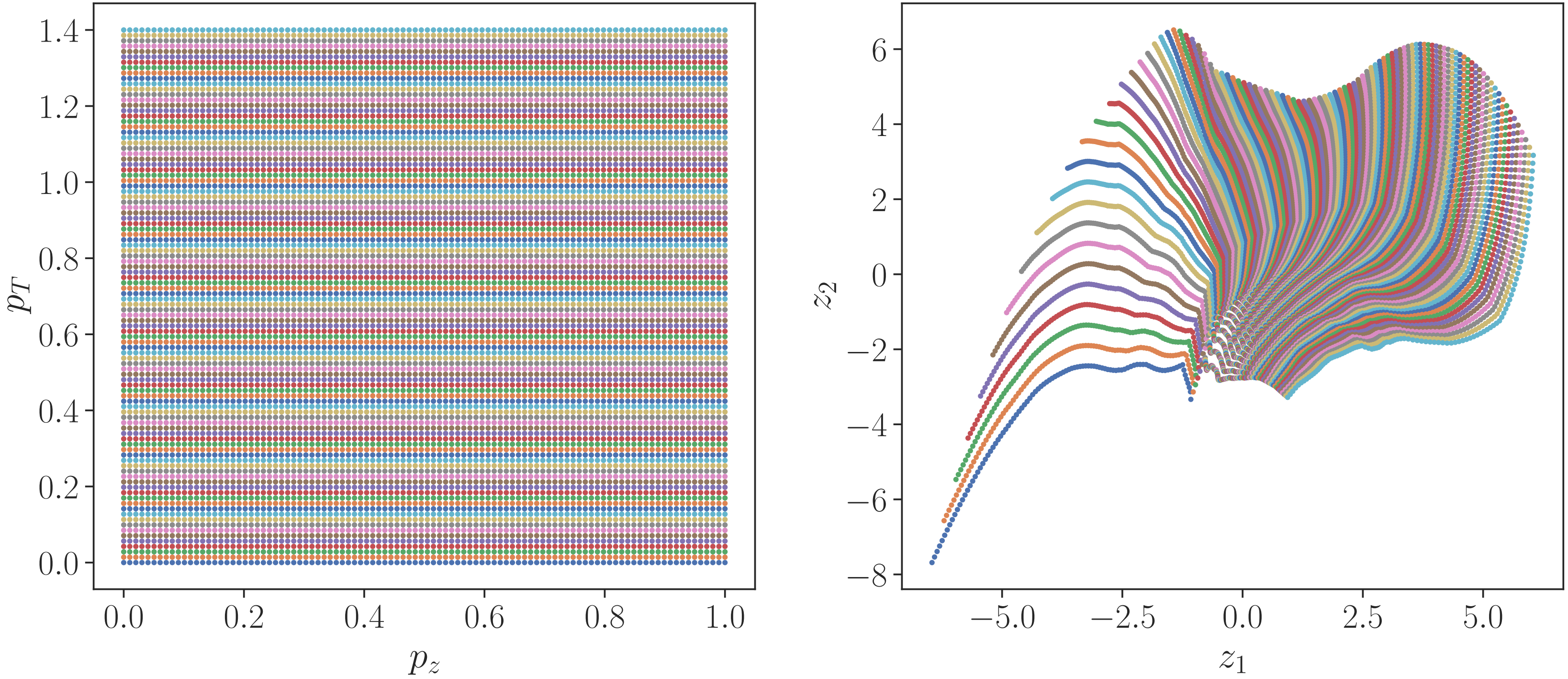


Use tra
(score) as an observable

INN learned mapping

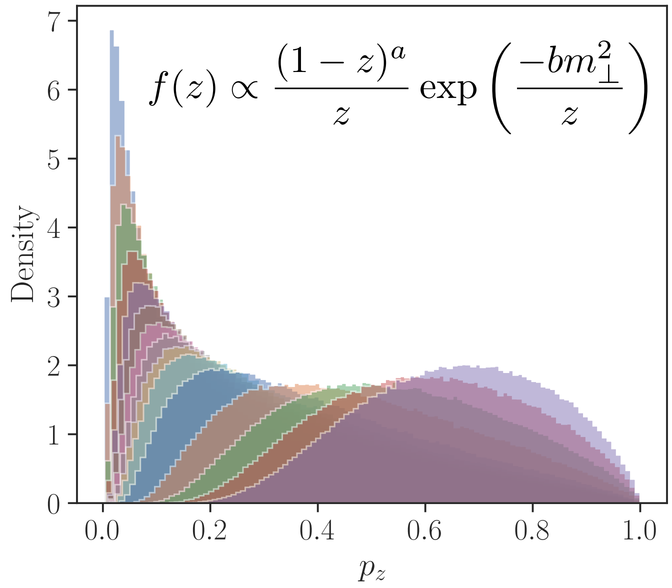


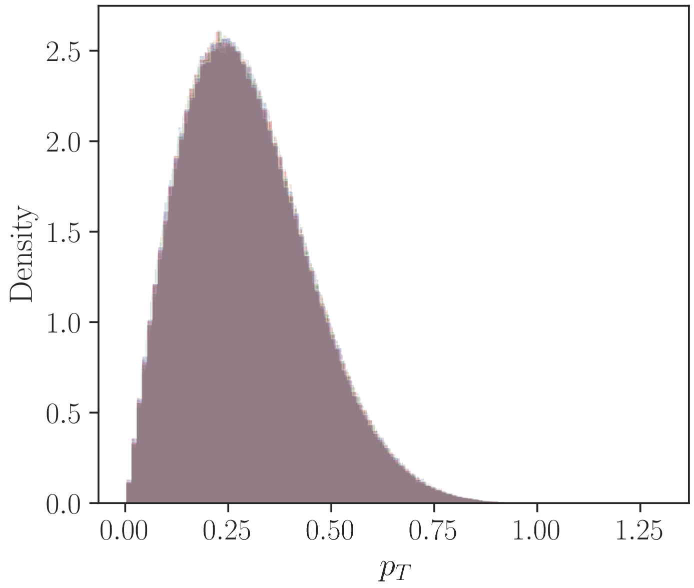
INN learned mapping



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 $q\bar{q} \rightarrow h$ 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.

MAGIC: $q\bar{q}$

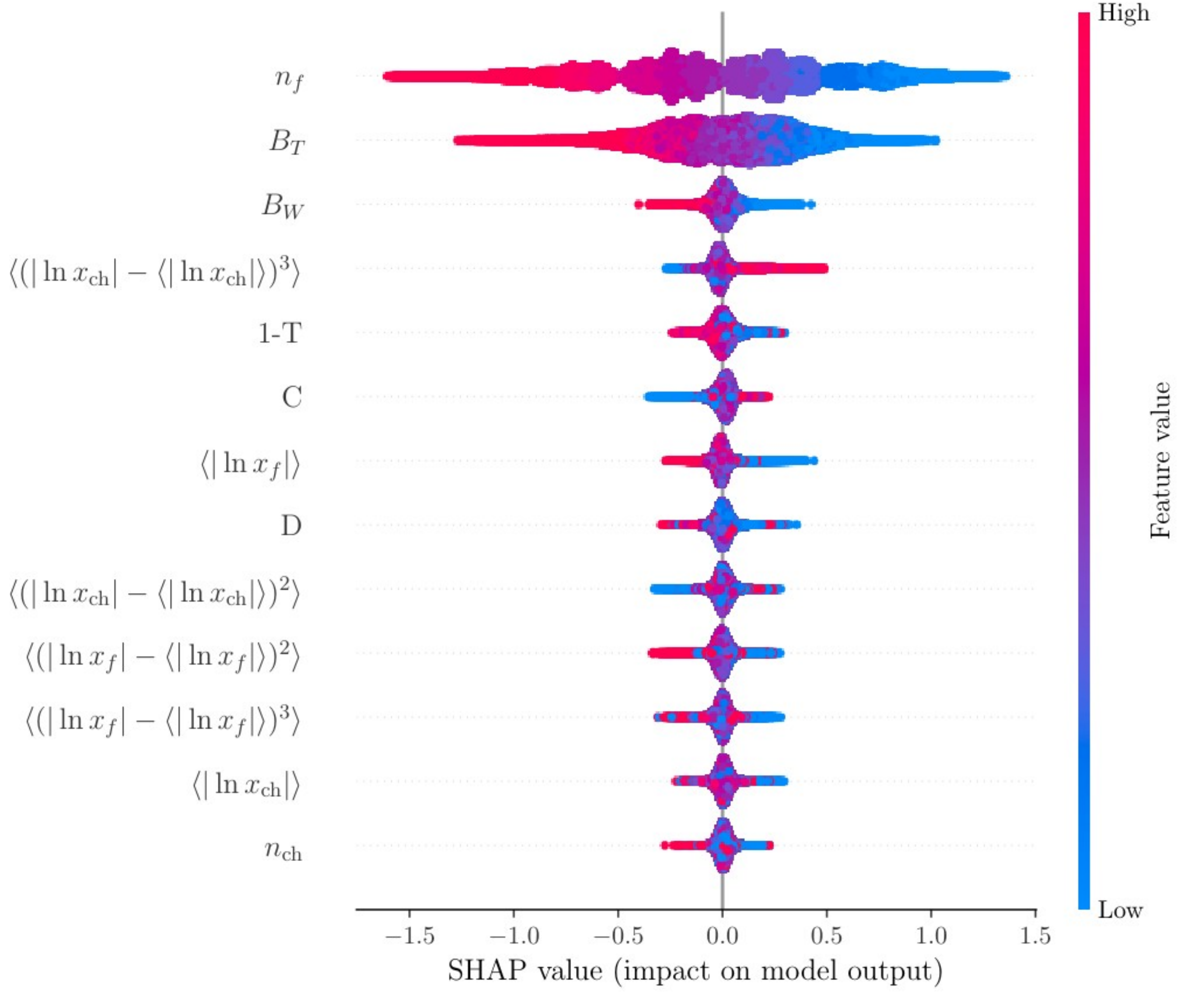
Use hadron multiplicity N as global event-level observable

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

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

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

SHAP values



High
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)
Feature value
Low

Brownian bridge

- Gaussian random walk ($E_{CM}>>m_h$)
  - Markovian

- Gaussian random walk with termination condition, i.e. Gaussian bridge
  - Psuedo-Markovian

Brownian bridge

- Gaussian random walk (E_{CM}>>m_h)

$$dX_t = dW_t$$

- Gaussian bridge
  - Boundary conditions

$$X_0 = x_0 \qquad X_T = x_T$$

$$dX_t = \frac{x_T - X_t}{T - t} dt + dW_t$$

Brownian bridge

- Gaussian random walk ($E_{CM} >> m_h$)

$$dX_t = dW_t$$

- Gaussian bridge
  - Boundary conditions

$$X_0 = x_0 \qquad X_T = x_T$$

$$dX_t = \frac{x_T - X_t}{T - t} dt + dW_t$$

Doob h-transform

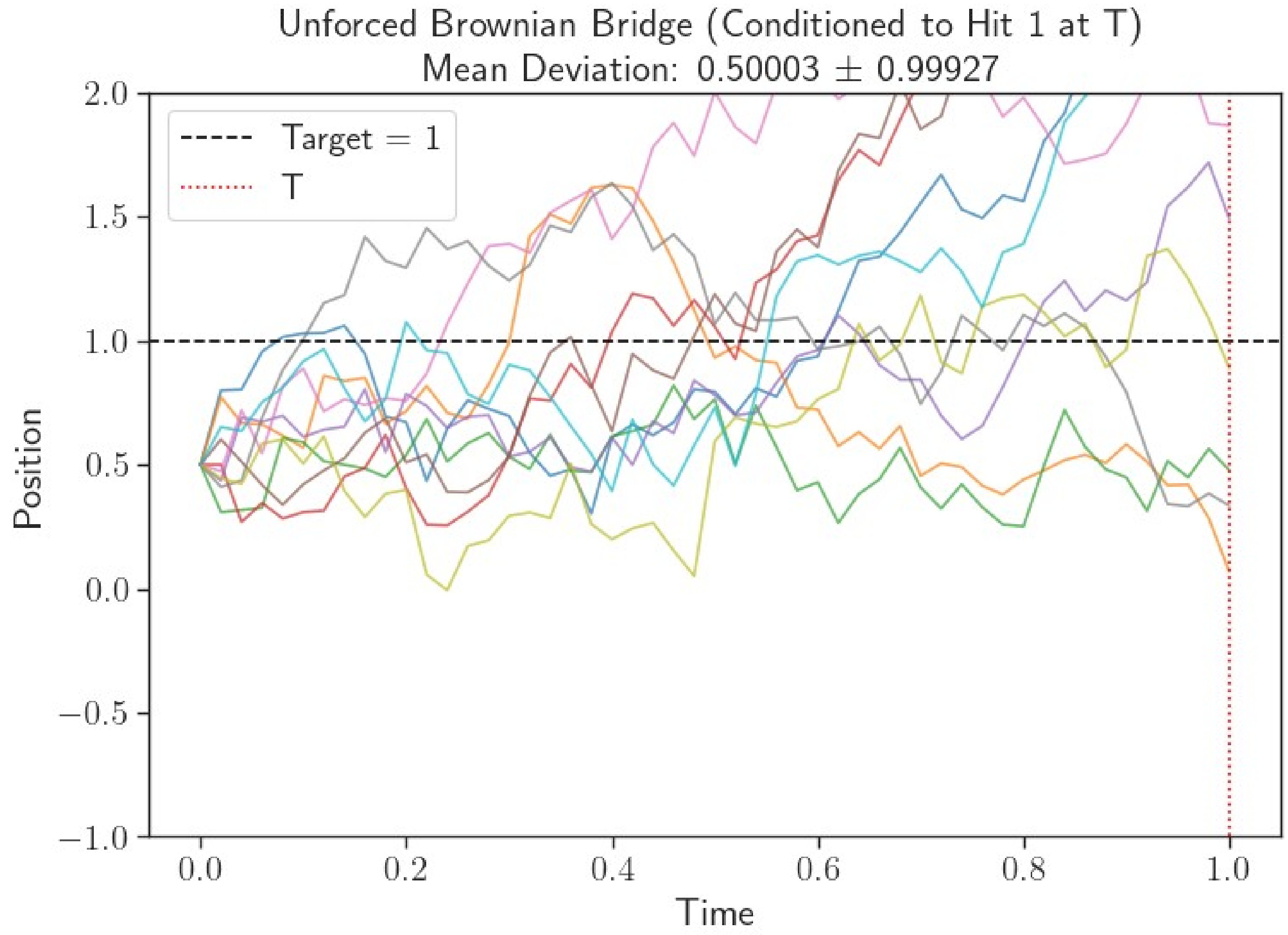
• Gau ge

wall nditions

c $_T = x_T$

$+ dW_t$

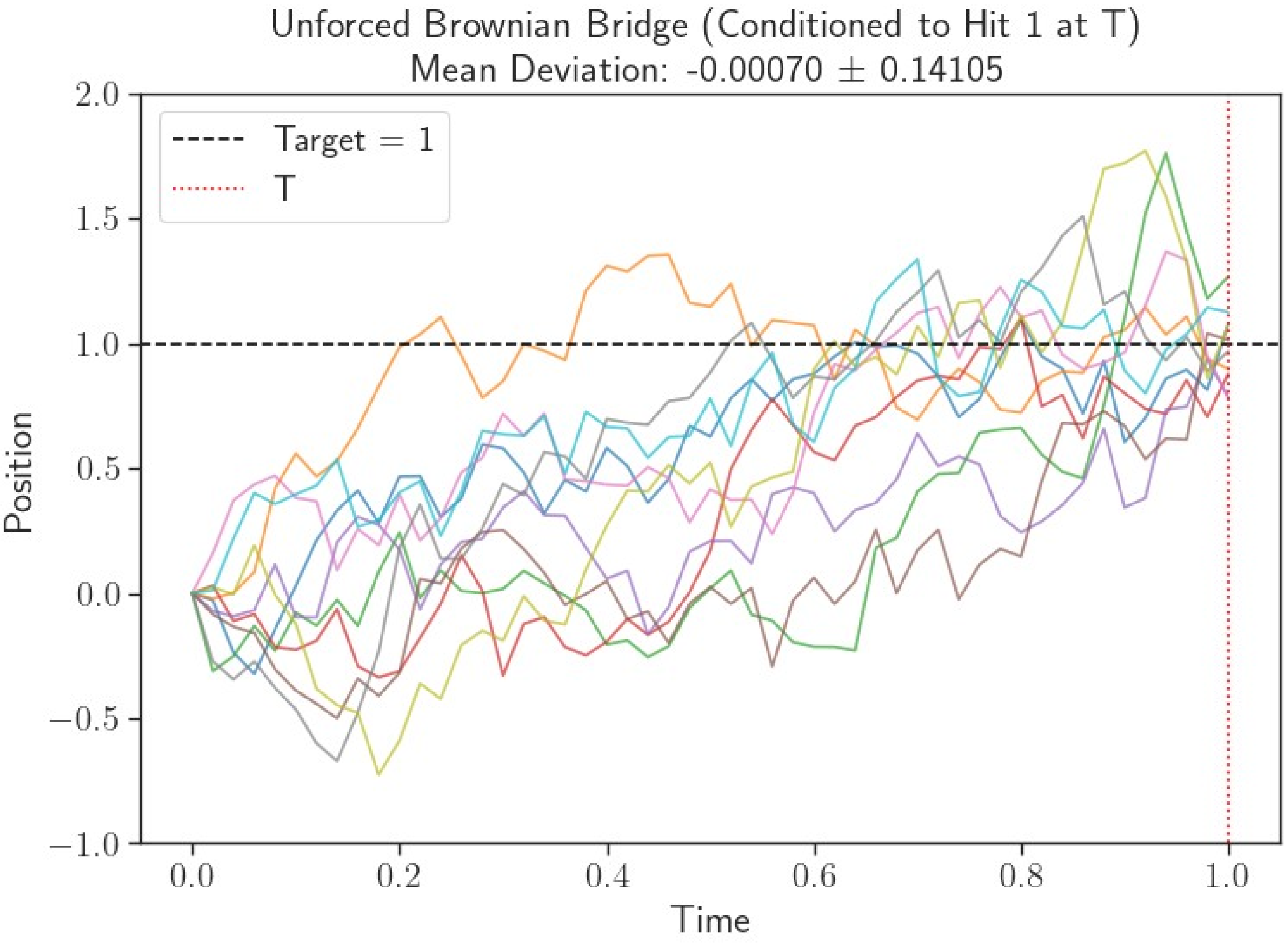
rm



• **Gaus**
walk **tions**

dX $= x_T$

dW_t



- **Gau… …e walk… …ditions**

d… … $= x_T$

… $+ dW_t$

…m