Modern machine learning methods for new physics searches

Sapienta ex machina?

UCLouvain

14th NExT PhD workshop — Cosener's House 2023

Ramon Winterhalder — UCLouvain

Why care about ML in physics?

Why ML in HEP?

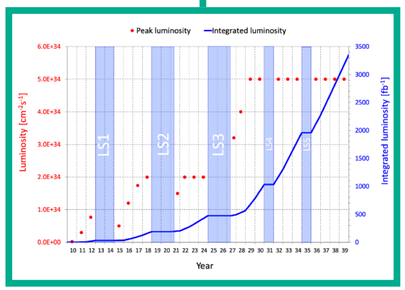
Data volume

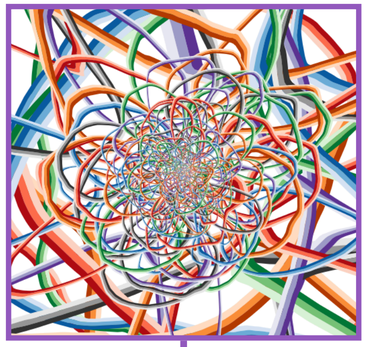
Large amounts of data

1. labeled (Simulation)
2. unlabeled (Detector)

ML wants lots of data

1





2

Complexity

High-dimensional & highly correlated data structure

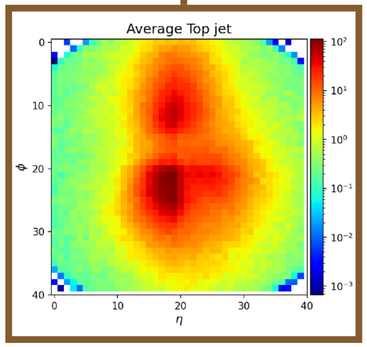
ML is expressive and interpretable

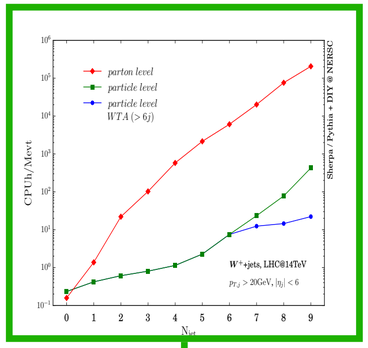
Signal detection

Rare and elusive signals among large backgrounds

ML has high accuracy and sensitivity

3





4

Computing Budget

Simulation & analysis is computationally expensive

ML is fast

Increasing interest

> 150 paper/year

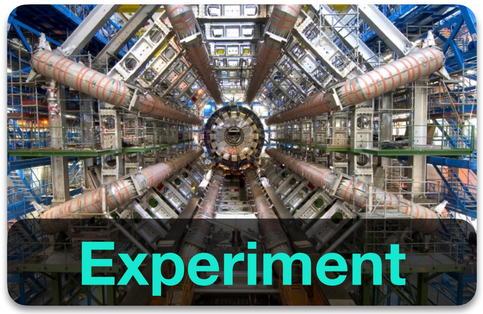
Future of HEP?

ML is fun!

5

LHC analysis (oversimplified)

Nature

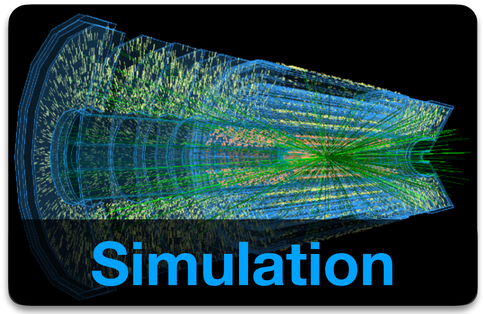


Detector-level observables

New Physics?

Pattern recognition

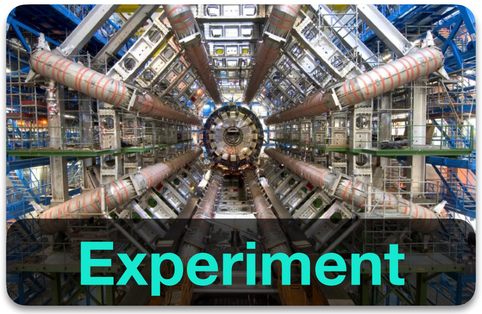
Theory



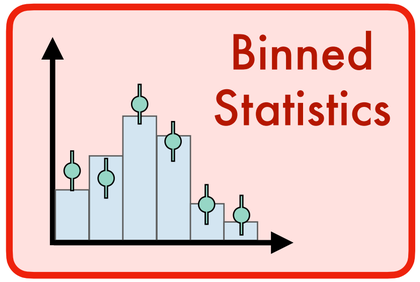
Detector-level observables

Problem: Information bottleneck

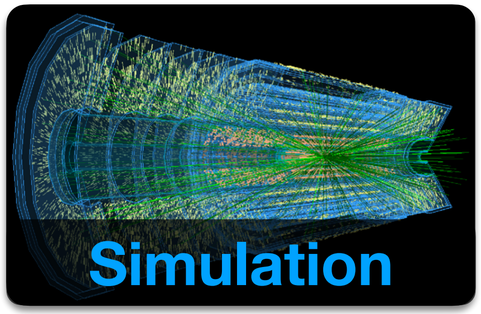
Nature



Detector-level observables



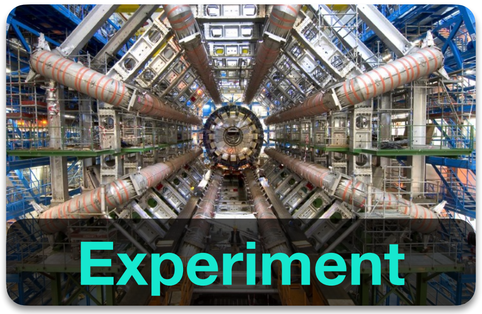
Theory



Detector-level observables

Problem: Information bottleneck

Nature

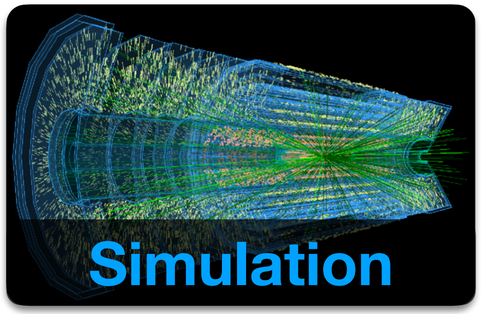


Detector-level observables

HL-LHC challenge

- ⊖ x20 speed up
- ⊖ higher precision in theory predictions

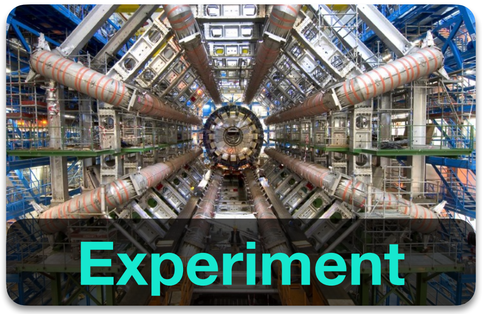
Theory



Detector-level observables

Solution: LHC analysis + ML

Nature



Detector-level observables

HL-LHC chance

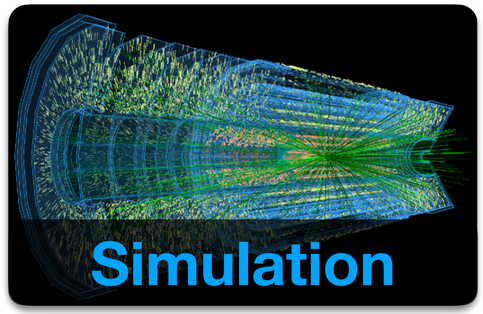
- ⊕ x20 speed up
- ⊕ higher precision in theory predictions

Optimal Inference

$$r = \frac{p(x\,|\,\theta_0)}{p(x\,|\,\theta_1)}$$

$$s = \nabla_\theta \log p(x\,|\,\theta)$$

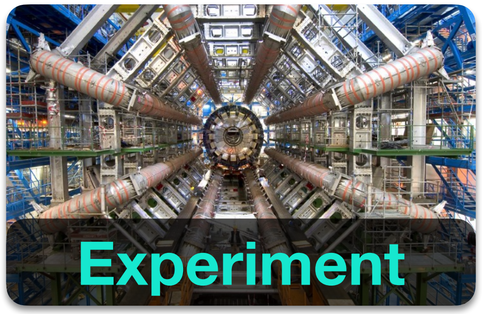
Theory



Detector-level observables

Solution: LHC analysis + ML

Nature



Detector-level observables

HL-LHC chance

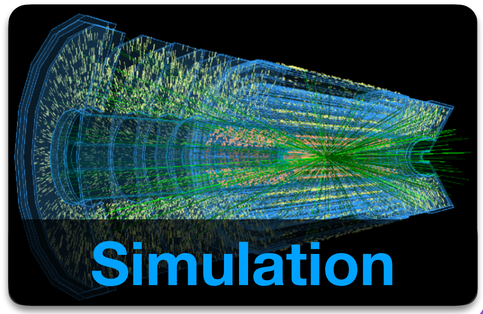
- ⊕ x20 speed up
- ⊕ higher precision in theory predictions

$$r = \frac{p(x\,|\,\theta_0)}{p(x\,|\,\theta_1)}$$

$$s = \nabla_\theta \log p(x\,|\,\theta)$$

Optimal Inference

Theory



Detector-level observables

Fast simulation

Solution: LHC analysis + ML

Nature

Anomaly detection

Detector-level observables

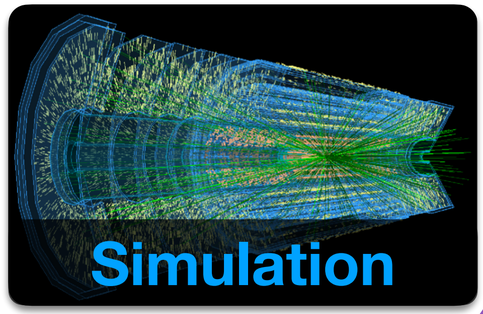
HL-LHC chance

⊕ x20 speed up
⊕ higher precision in theory predictions

$r = \frac{p(x|\theta_0)}{p(x|\theta_1)}$ Optimal Inference

$s = \nabla_\theta \log p(x|\theta)$

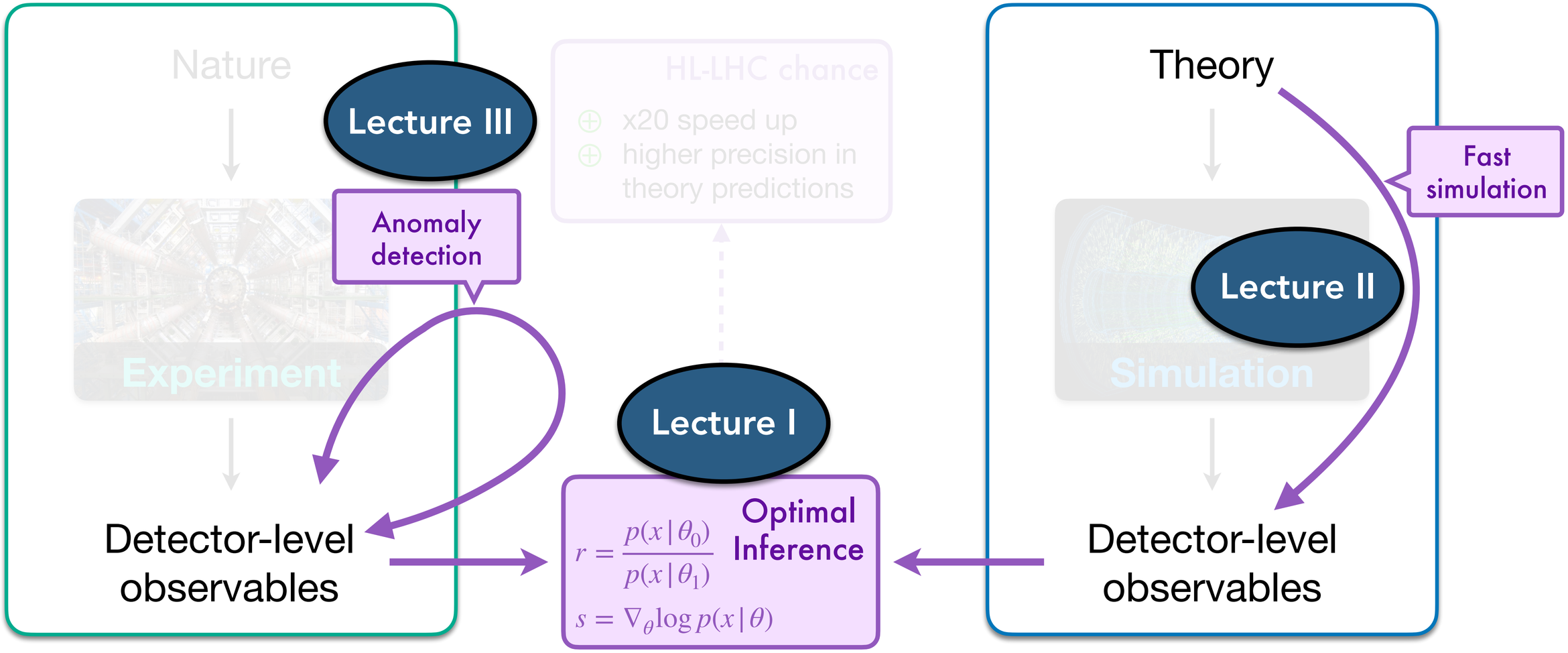
Theory



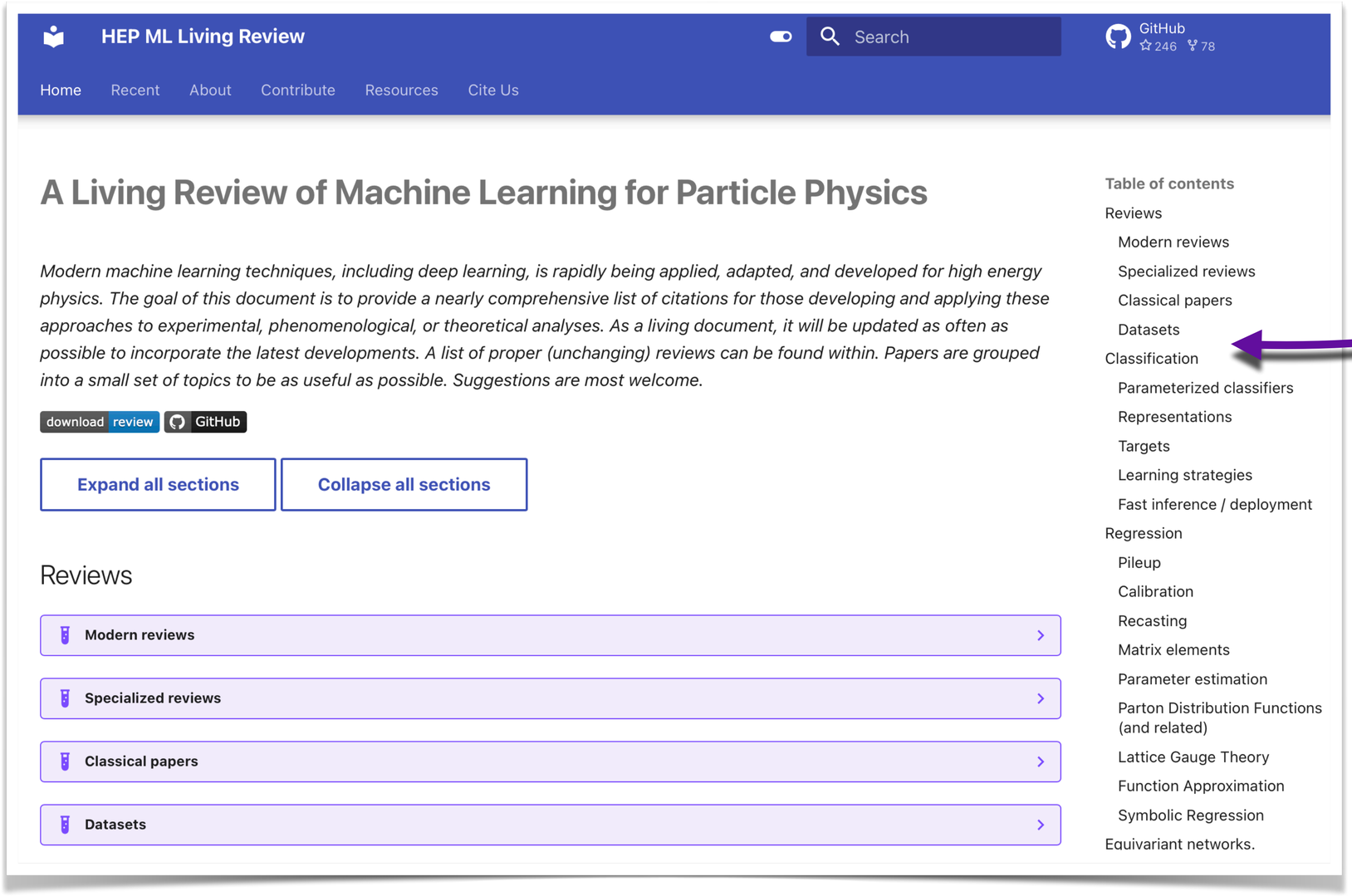
Fast simulation

Detector-level observables

Solution: LHC analysis + ML



HEPML Living Review



- Check LivingReview for many **ML4HEP applications**

HEPML

Plan of attack

Today

1. Introduction to Machine Learning

- Classification and Regression — Lecture I (60min)
- Example: **MadMiner, Top Tagging**

Coffee Break

2. Generative Models for the LHC

- Diffusion Models, Normalizing flows — Lecture II (60min)
- Example: **MadNIS, MEM-ML**

Tomorrow

3. Anomaly detection

- Autoencoders and CWoLA — Lecture III (60min)
- Examples: **CWoLA-Hunting, ANODE, CATHODE,…**

Plan of attack

Today

1. Introduction to Machine Learning
   - Classification and Regression — Lecture I (60min)
   - Example: **MadMiner, Top Tagging**

COFFEE BREAK

2. Generative Models for the LHC
   - Diffusion Models, Normalizing flows — Lecture II (60min)
   - Example: **MadNIS, MEM-ML**

Tomorrow

3. Anomaly detection
   - Autoencoders and CWoLA — Lecture III (60min)
   - Examples: **CWoLA-Hunting, ANODE, CATHODE,…**

Part I

Introduction to Machine Learning

What is machine learning?

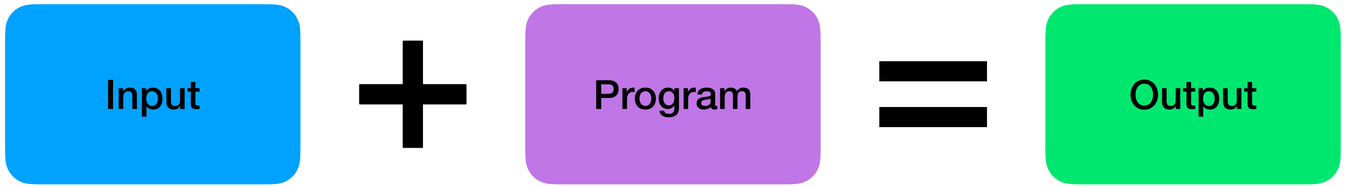
What is machine learning?

Tom Mitchell

ML pioneer

"Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data"

Traditional approach



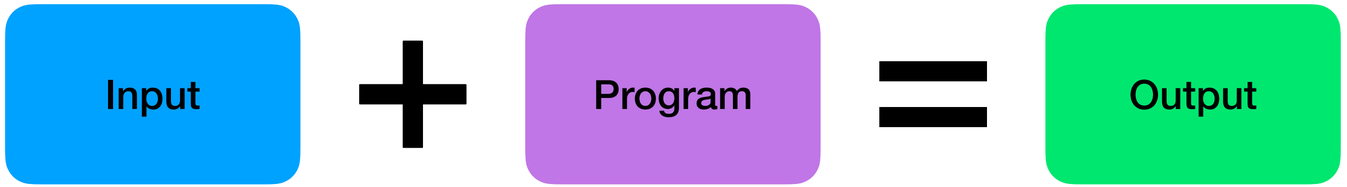
What is machine learning?

Tom Mitchell

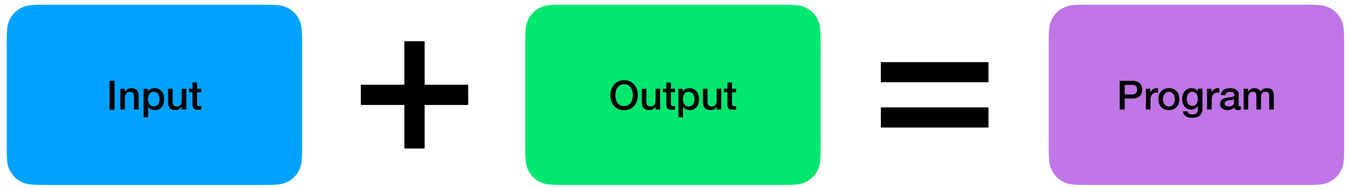
ML pioneer

"Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data"

Traditional approach



Machine learning



What is machine learning?

Tom Mitchell

ML pioneer

"Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data"

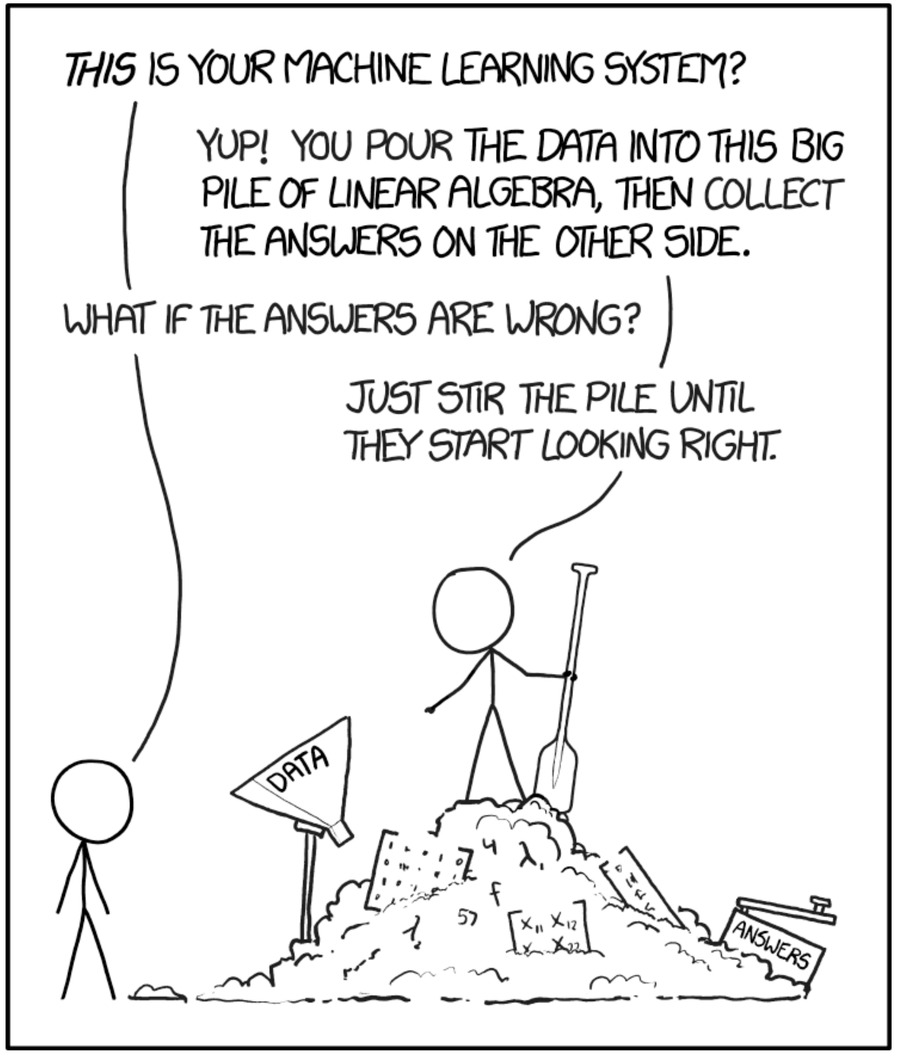
1. **algorithm**: a method to perform a task of interest
2. **experience**: training data, which the algorithm can use to learn how to perform a task
3. **improve**: a way to measure the performance on the training data
4. **automatically**: a strategy to exploit the training data, without external input

What is machine learning?

What if feels like… (sometimes)

Aim of the lectures:

Giving you the tools and ideas to use it right!



What is machine learning?

What if feels like… (sometimes)

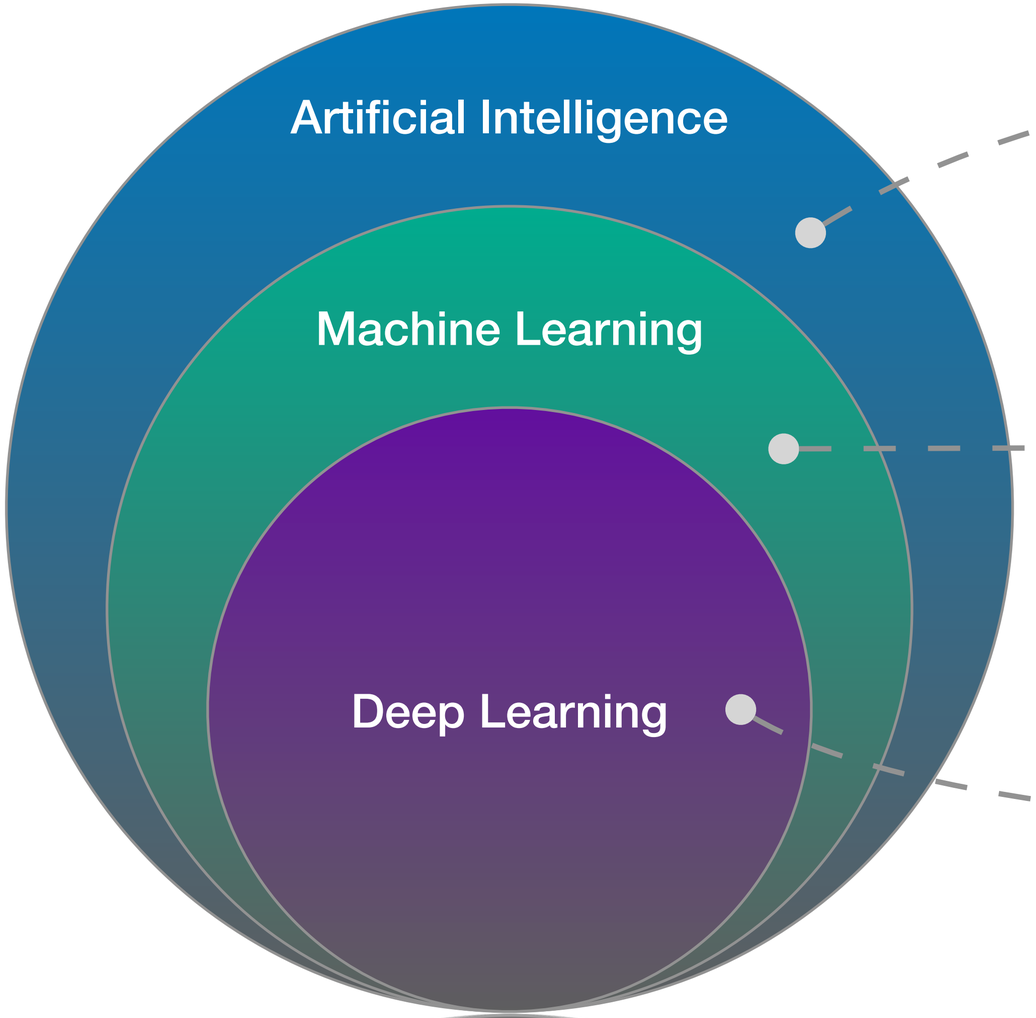
Be aware!

The core of machine learning is to find structure in data - **no more no less!**

Thanks to machine-learning algorithms,
the robot apocalypse was short-lived.

What is deep learning?

What is deep learning?



ARTIFICIAL INTELLIGENCE

A technique which enables machines to mimic human behaviour

MACHINE LEARNING

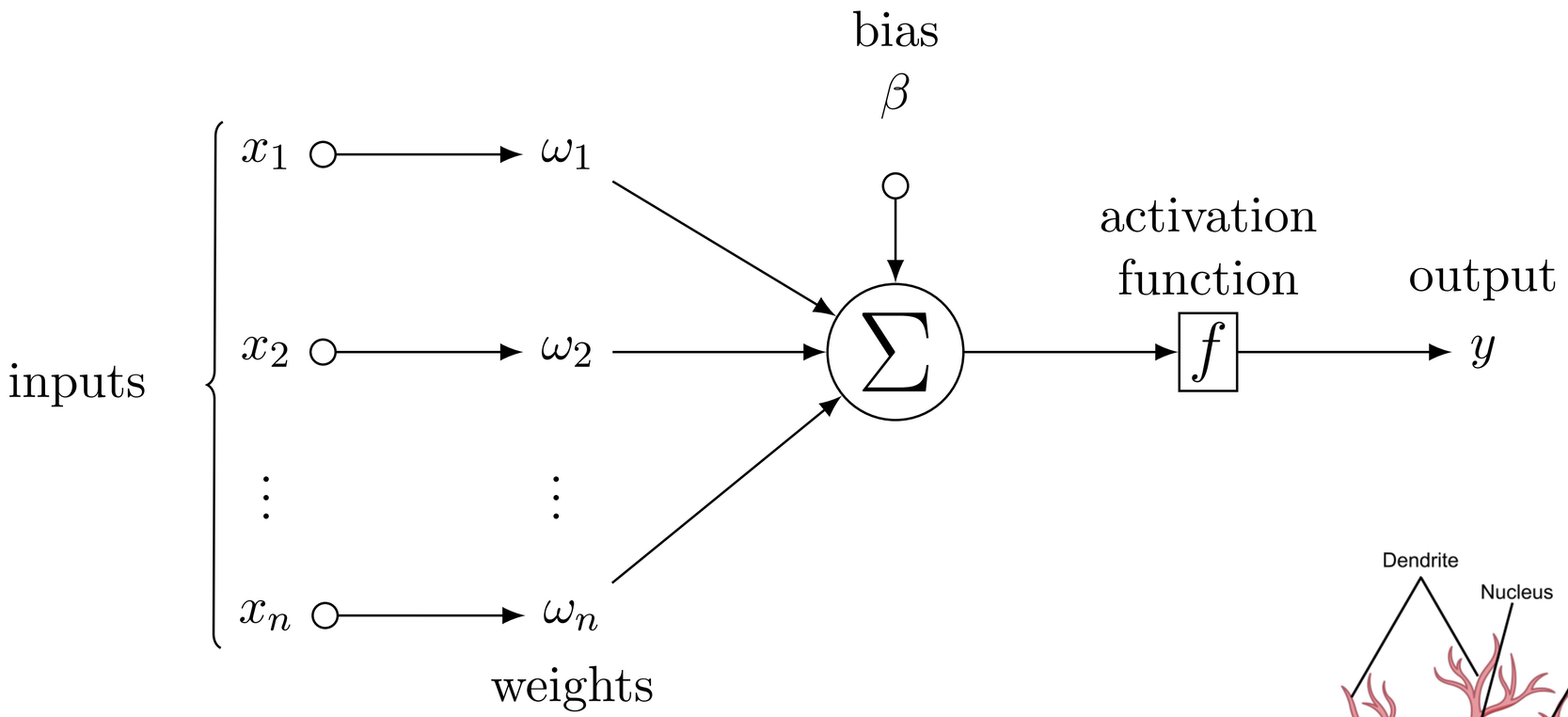
Subset of AI technique which use statistical methods to enable machines to improve with experience

DEEP LEARNING

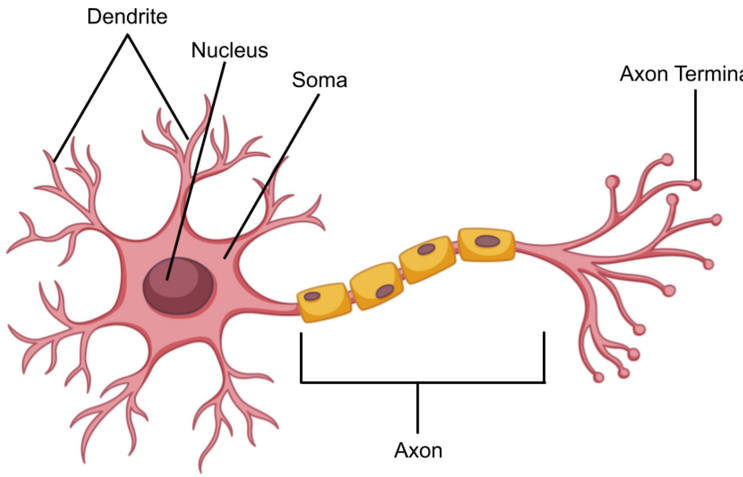
Subset of ML which make the computation of multi-layer neural networks feasible

What are neural networks?

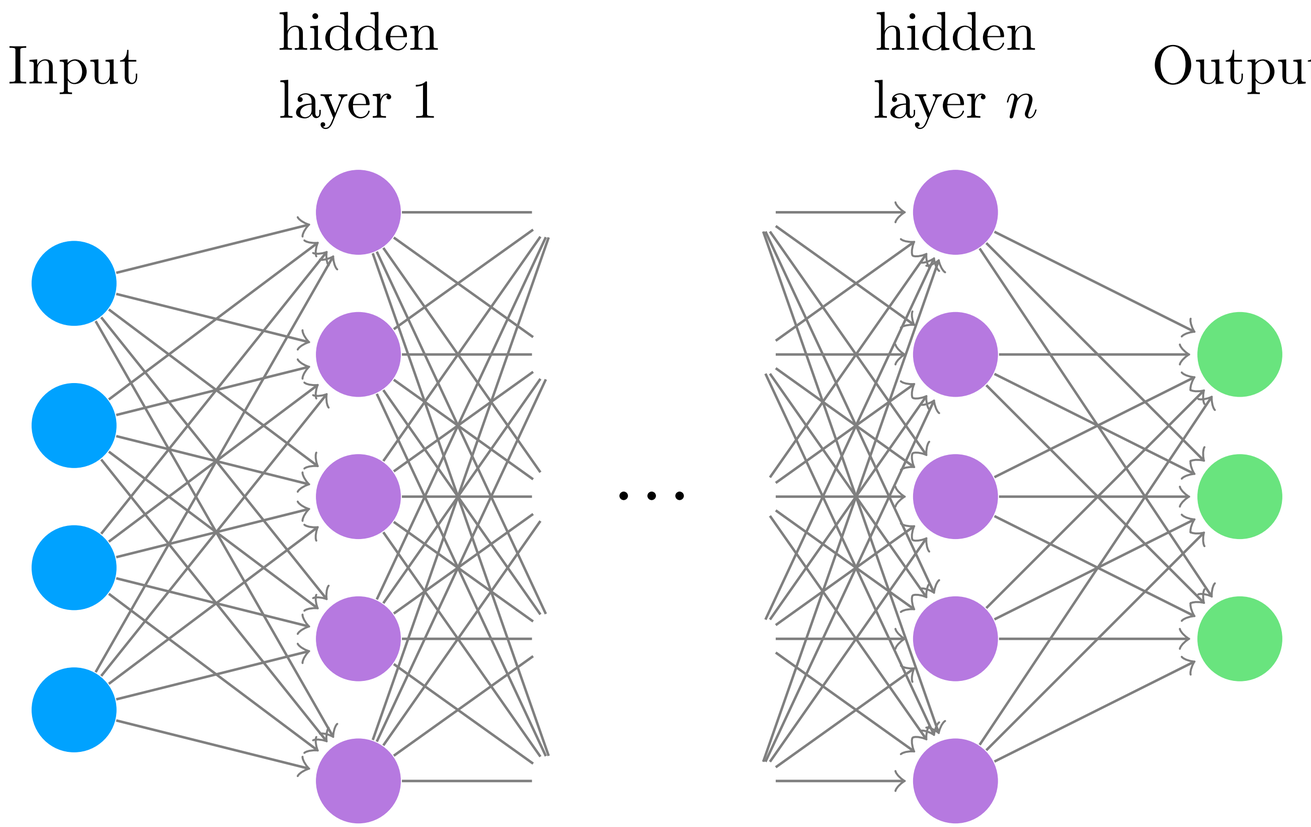
Components of artificial neurons



$$y = f\left(\sum_{i=1}^{n} \omega_i x_i + \beta_i\right)$$



Deep neural network



$$y_j = f_j\left(\sum_m \omega_{jm}^{(n+1)}\, g_m^{(n)}\left(\cdots\left(\sum_i \omega_{kl}^{(2)} g_l^{(1)}\left(\sum_i \omega_{li}^{(1)} x_i + \beta_l^{(1)}\right) + \beta_k^{(2)}\right)\cdots\right) + \beta_j^{(n+1)}\right)$$

Training objective and optimization

Performance measure

In order to train the neural network, we need a **training objective** or **loss function**:

$$\mathscr{L}_{\text{tot}} = \sum_{i}^{N} \mathscr{L}(f_\omega(x_i), y_i)$$

Backpropagation

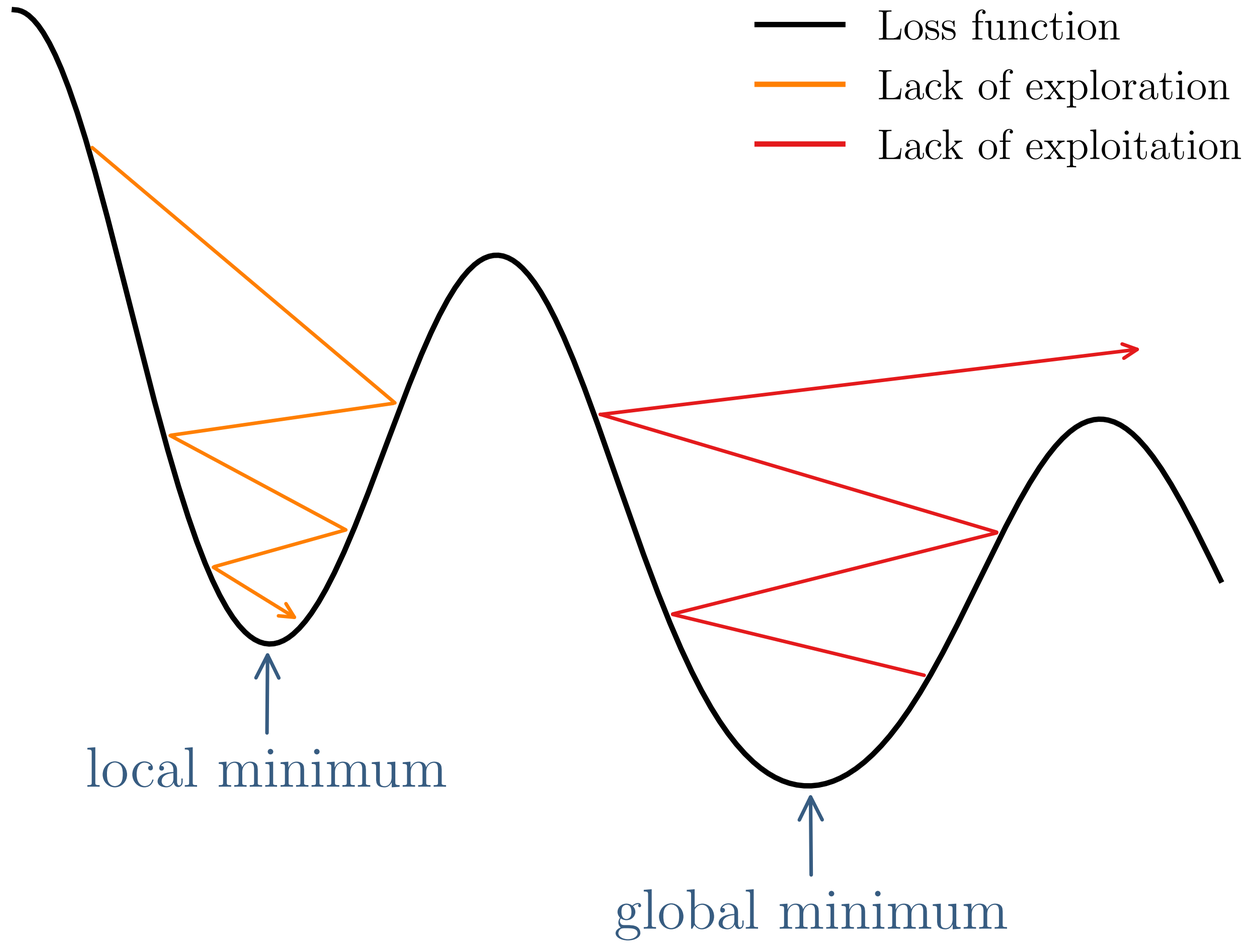
$$\nabla_\omega \mathscr{L}(f_\omega(x), y) = \frac{\mathrm{d}\mathscr{L}(f_\omega(x), y)}{\mathrm{d}\omega} = \sum_{ij} \frac{\partial \mathscr{L}}{\partial f_{j,\omega}} \cdot \frac{\partial f_{j,\omega}}{\partial \omega_i} \stackrel{!}{=} 0$$

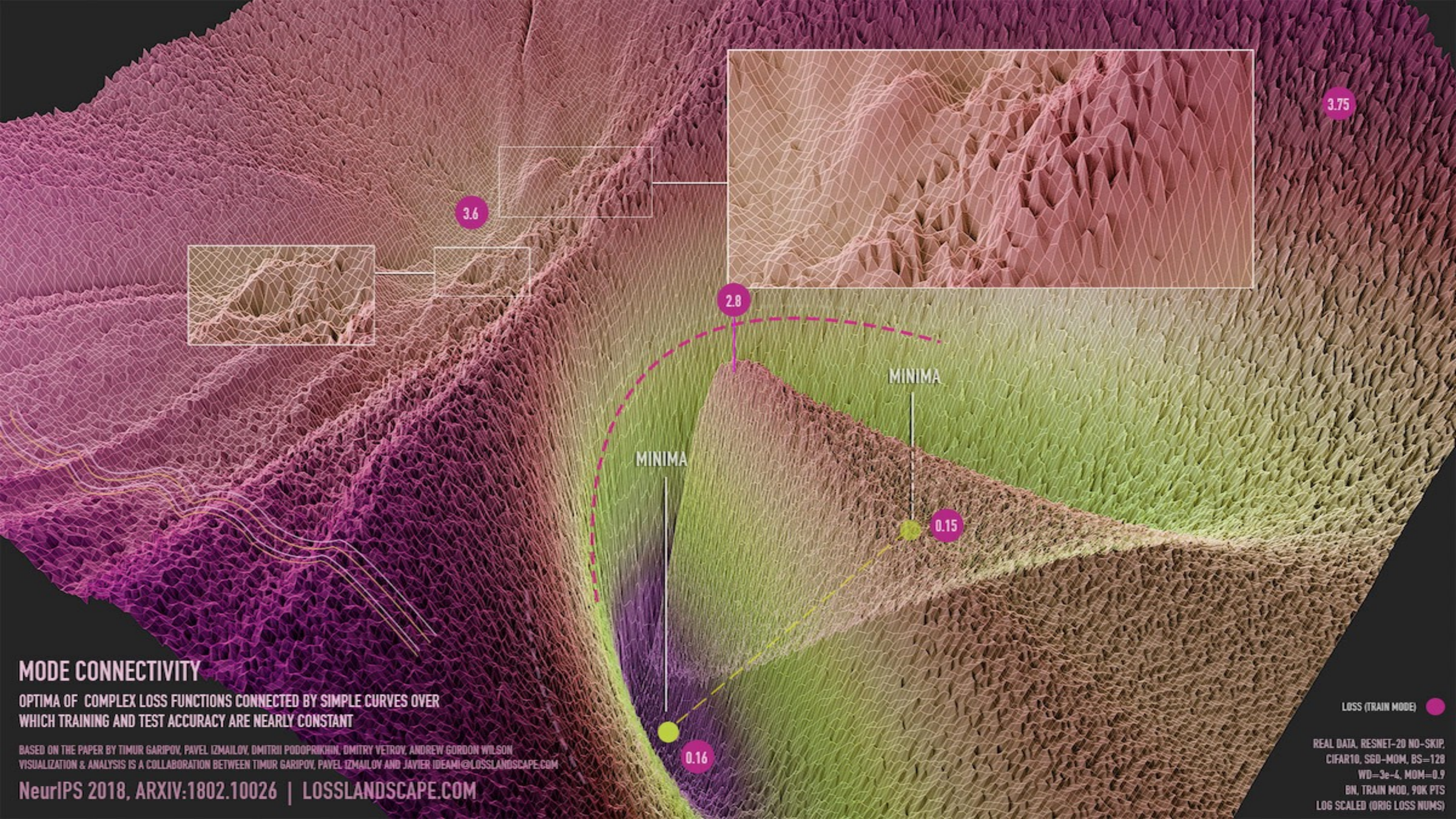
Stochastic gradient descent

$$\omega_{t+1} = \omega_t - \eta \nabla_{\omega_t} \mathscr{L}_b(\omega), \qquad \text{with} \quad \mathscr{L}_b(\omega) = \sum_{i}^{b} \mathscr{L}(f_\omega(x_i), y_i)$$

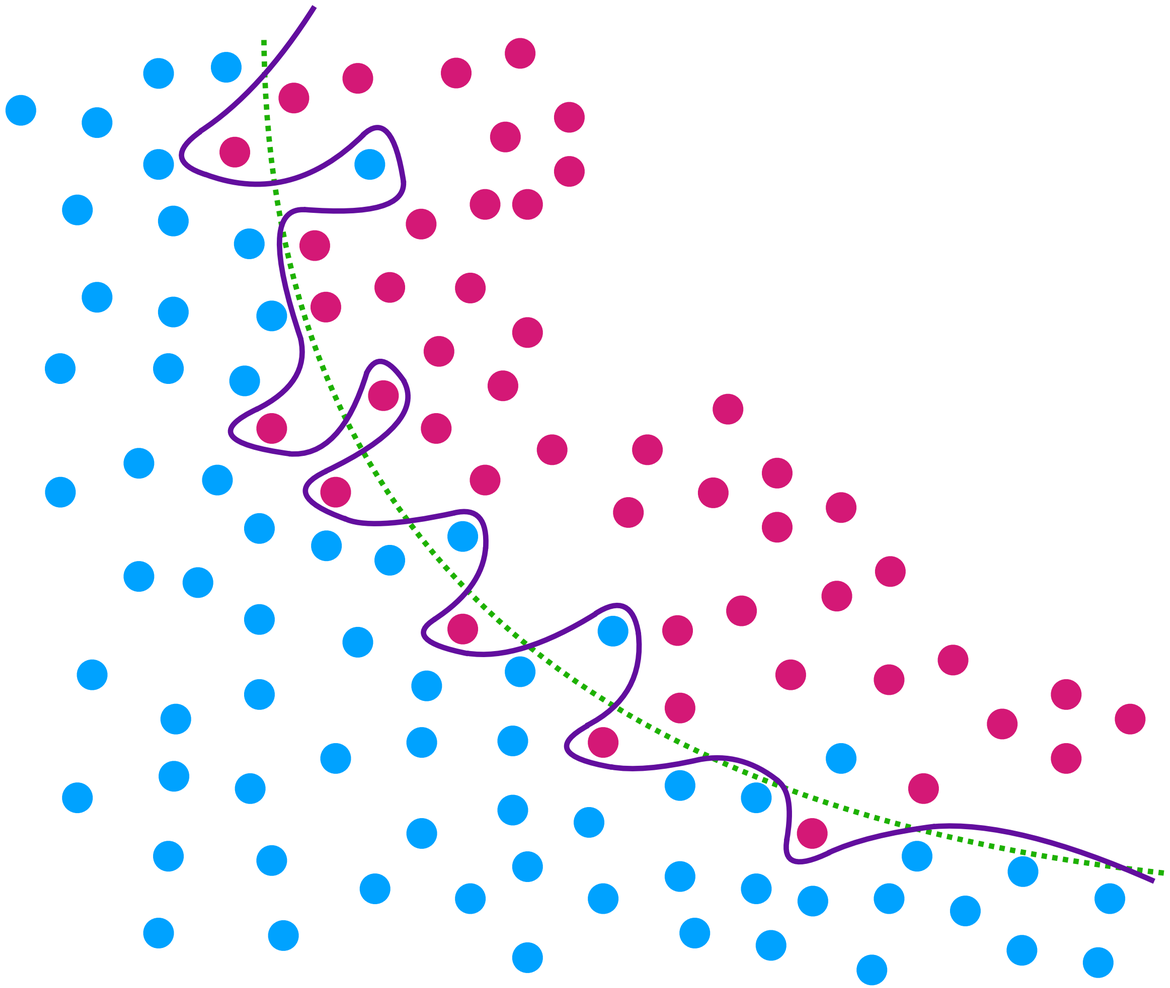
Optimizer and loss landscape

Finding the global minimum in a sea of local minima




3.75
3.6
2.8
MINIMA
MINIMA
0.15
0.16
MODE CONNECTIVITY
OPTIMA OF COMPLEX LOSS FUNCTIONS CONNECTED BY SIMPLE CURVES OVER
WHICH TRAINING AND TEST ACCURACY ARE NEARLY CONSTANT
BASED ON THE PAPER BY TIMUR GARIPOV, PAVEL IZMAILOV, DMITRII PODOPRIKHIN, DMITRY VETROV, ANDREW GORDON WILSON
VISUALIZATION & ANALYSIS IS A COLLABORATION BETWEEN TIMUR GARIPOV, PAVEL IZMAILOV AND JAVIER IDEAMI@LOSSLANDSCAPE.COM
NeurIPS 2018, ARXIV:1802.10026 | LOSSLANDSCAPE.COM
LOSS (TRAIN MODE)
REAL DATA, RESNET-20 NO-SKIP,
CIFAR10, SGD-MOM, BS=128
WD=3e-4, MOM=0.9
BN, TRAIN MOD, 90K PTS
LOG SCALED (ORIG LOSS NUMS)

Overfitting und regularization



Identify

- Test/validation set:
  independent samples for testing/ validation

Regularization

- L1 & L2 Regularization: penalty term in the loss function
- Dropout: randomly sets the inputs to some nodes to zero

How to choose the loss function?

Fits and interpolations

Approximate function

$$f_\omega(x) \approx f(x)$$

Maximize probability for fit output

$$p(x\,|\,\omega) = \prod_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{|f_j - f_\omega(x_j)|^2}{2\sigma_j}\right)$$

$$\Rightarrow \quad \log p(x\,|\,\omega) = -\sum_j \frac{|f_j - f_\omega(x_j)|^2}{2\sigma_j} + \text{const}$$

Fits and interpolations

Approximate function

$$f_\omega(x) \approx f(x)$$

Assumes Gaussian probablity distribution

Maximize probability for fit output

$$p(x\,|\,\omega) = \prod_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{|f_j - f_\omega(x_j)|^2}{2\sigma_j} \right)$$

$$\Rightarrow \quad \log p(x\,|\,\omega) = -\sum_j \frac{|f_j - f_\omega(x_j)|^2}{2\sigma_j} + \text{const}$$

Fits and interpolations

Approximate function

$$f_\omega(x) \approx f(x)$$

Assumes Gaussian probablity distribution

Maximize probability for fit output

$$p(x\,|\,\omega) = \prod_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{|f_j - f_\omega(x_j)|^2}{2\sigma_j}\right)$$

$$\Rightarrow \quad \log p(x\,|\,\omega) = -\sum_j \frac{|f_j - f_\omega(x_j)|^2}{2\sigma_j} + \text{const}$$

Minimize negative log-likelihood

Loss function of the fit

$$\mathscr{L}_{\text{fit}} = \sum_j \frac{\mathscr{L}_j}{N} = \sum_j \frac{|f_j - f_\omega(x_j)|^2}{2\sigma_j N}$$

Regression in Machine Learning

Loss function in fits

$$\mathscr{L}_{\text{fit}} = \frac{1}{N}\sum_j \mathscr{L}_j = \frac{1}{N}\sum_j \frac{|f_j - f_\omega(x_j)|^2}{2\sigma_j}$$

Regression in Machine Learning

Loss function in fits

$$\mathscr{L}_{\text{fit}} = \frac{1}{N}\sum_j \mathscr{L}_j = \frac{1}{N}\sum_j \frac{|f_j - f_\omega(x_j)|^2}{2\sigma_j}$$

if error σ_j unknown
or same for all

Typical ML Regression loss

$$\mathscr{L} = \frac{1}{2\sigma N}\sum_j |f_j - f_\omega(x_j)|^2 \equiv \frac{1}{2\sigma}\text{MSE}$$

Regression in Machine Learning

Loss function in fits

$$\mathscr{L}_{\text{fit}} = \frac{1}{N}\sum_j \mathscr{L}_j = \frac{1}{N}\sum_j \frac{|f_j - f_\omega(x_j)|^2}{2\sigma_j}$$

if error σ_j unknown or same for all

Typical ML Regression loss

$$\mathscr{L} = \frac{1}{2\sigma N}\sum_j |f_j - f_\omega(x_j)|^2 \equiv \frac{1}{2\sigma}\text{MSE}$$

Puts more weight solely on deviations of the function values!

Regression in Machine Learning

Loss function in fits

$$\mathscr{L}_{\text{fit}} = \frac{1}{N}\sum_j \mathscr{L}_j = \frac{1}{N}\sum_j \frac{|f_j - f_\omega(x_j)|^2}{2\sigma_j}$$

if error σ_j unknown or same for all

Typical ML Regression loss

$$\mathscr{L} = \frac{1}{2\sigma N}\sum_j |f_j - f_\omega(x_j)|^2 \equiv \frac{1}{2\sigma}\text{MSE}$$

Puts more weight solely on deviations of the function values!

Preprocessing of the training data

$$f_j \to \log f_j \qquad f_j \to f_j - \langle f_j \rangle \qquad f_j \to \frac{f_j}{\langle f_j \rangle} \cdots$$

Regression in Machine Learning

Loss function in fits

$$\mathscr{L}_{\text{fit}} = \frac{1}{N}\sum_j \mathscr{L}_j = \frac{1}{N}\sum_j \frac{|f_j - f_\omega(x_j)|^2}{2\sigma_j}$$

if error σ_j unknown or same for all

Typical ML Regression loss

$$\mathscr{L} = \frac{1}{2\sigma N}\sum_j |f_j - f_\omega(x_j)|^2 \equiv \frac{1}{2\sigma}\text{MSE}$$

Puts more weight solely on deviations of the function values!

Preprocessing of the training data

$$f_j \to \log f_j \qquad f_j \to f_j - \langle f_j \rangle \qquad f_j \to \frac{f_j}{\langle f_j \rangle} \cdots$$

Mean-absolute error

$$\mathscr{L}_{\text{MAE}} = \frac{1}{N}\sum_j |f_j - f_\omega(x_j)|$$

What about Classification?

Approximate data probability

$$p_\omega(x) \approx p_{\text{data}}(x)$$

What about Classification?

Approximate data probability

$$p_\omega(x) \approx p_{\text{data}}(x)$$

IX. *On the Problem of the most Efficient Tests of Statistical Hypotheses.*

By J. NEYMAN, *Nencki Institute, Soc. Sci. Lit. Varsoviensis, and Lecturer at the Central College of Agriculture, Warsaw, and* E. S. PEARSON, *Department of Applied Statistics, University College, London.*

(*Communicated by* K. PEARSON, *F.R.S.*)

(Received August 31, 1932.—Read November 10, 1932.)

ratio of the two likelihoods is the **most powerful** test statistic

What about Classification?

Approximate data probability

$$p_\omega(x) \approx p_{\text{data}}(x)$$

Optimal test statistic

Kullback-Leibler divergence

$$D_{\text{KL}}(p_{\text{data}}\,|\,p_\omega) = \left\langle \log \frac{p_{\text{data}}(x)}{p_\omega(x)} \right\rangle_{p_{\text{data}}} = \int \mathrm{d}x\, p_{\text{data}}(x)\, \log \frac{p_{\text{data}}(x)}{p_\omega(x)}$$

IX. *On the Problem of the most Efficient Tests of Statistical Hypotheses.*

By J. NEYMAN, *Nencki Institute, Soc. Sci. Lit. Varsoviensis, and Lecturer at the Central College of Agriculture, Warsaw, and* E. S. PEARSON, *Department of Applied Statistics, University College, London.*

(*Communicated by* K. PEARSON, *F.R.S.*)

(Received August 31, 1932.—Read November 10, 1932.)

ratio of the two likelihoods is the **most powerful** test statistic

Classification loss function

Classification loss

$$\mathscr{L}_{\text{class}} = \sum_{j=S,B} D_{\text{KL}}(p_{\text{data},j} \,|\, p_{\omega,,j})$$

$$= -\sum_{\{x\}} \left[p_{\text{data},S} \, \log p_{\omega,S} + p_{\text{data},B} \, \log p_{\omega,B} \right] + \text{const}$$

Classification loss function

Classification loss

$$\mathscr{L}_{\text{class}} = \sum_{j=S,B} D_{\text{KL}}(p_{\text{data},j} \,|\, p_{\omega,,j})$$
$$= -\sum_{\{x\}} \left[p_{\text{data},S} \log p_{\omega,S} + p_{\text{data},B} \log p_{\omega,B} \right] + \text{const}$$

Using $p_B = 1 - p_S$

Binary cross-entropy loss

$$\mathscr{L}_{\text{BCE}} = -\sum_{\{x\}} \left[p_{\text{data},S} \log p_{\omega,S} + (1 - p_{\text{data},S}) \log(1 - p_{\omega,S}) \right]$$

Classification loss function

Classification loss

$$\mathscr{L}_{\text{class}} = \sum_{j=S,B} D_{\text{KL}}(p_{\text{data},j} \,|\, p_{\omega,,j})$$
$$= -\sum_{\{x\}} \left[p_{\text{data},S} \log p_{\omega,S} + p_{\text{data},B} \log p_{\omega,B} \right] + \text{const}$$

Using $p_B = 1 - p_S$

Binary cross-entropy loss

$$\mathscr{L}_{\text{BCE}} = -\sum_{\{x\}} \left[p_{\text{data},S} \log p_{\omega,S} + (1 - p_{\text{data},S}) \log(1 - p_{\omega,S}) \right]$$

Generalization (multi-class)

$$\mathscr{L}_{\text{CE}} = -\sum_{j \in C_i} p_{\text{data},j} \log p_{\omega,j}$$

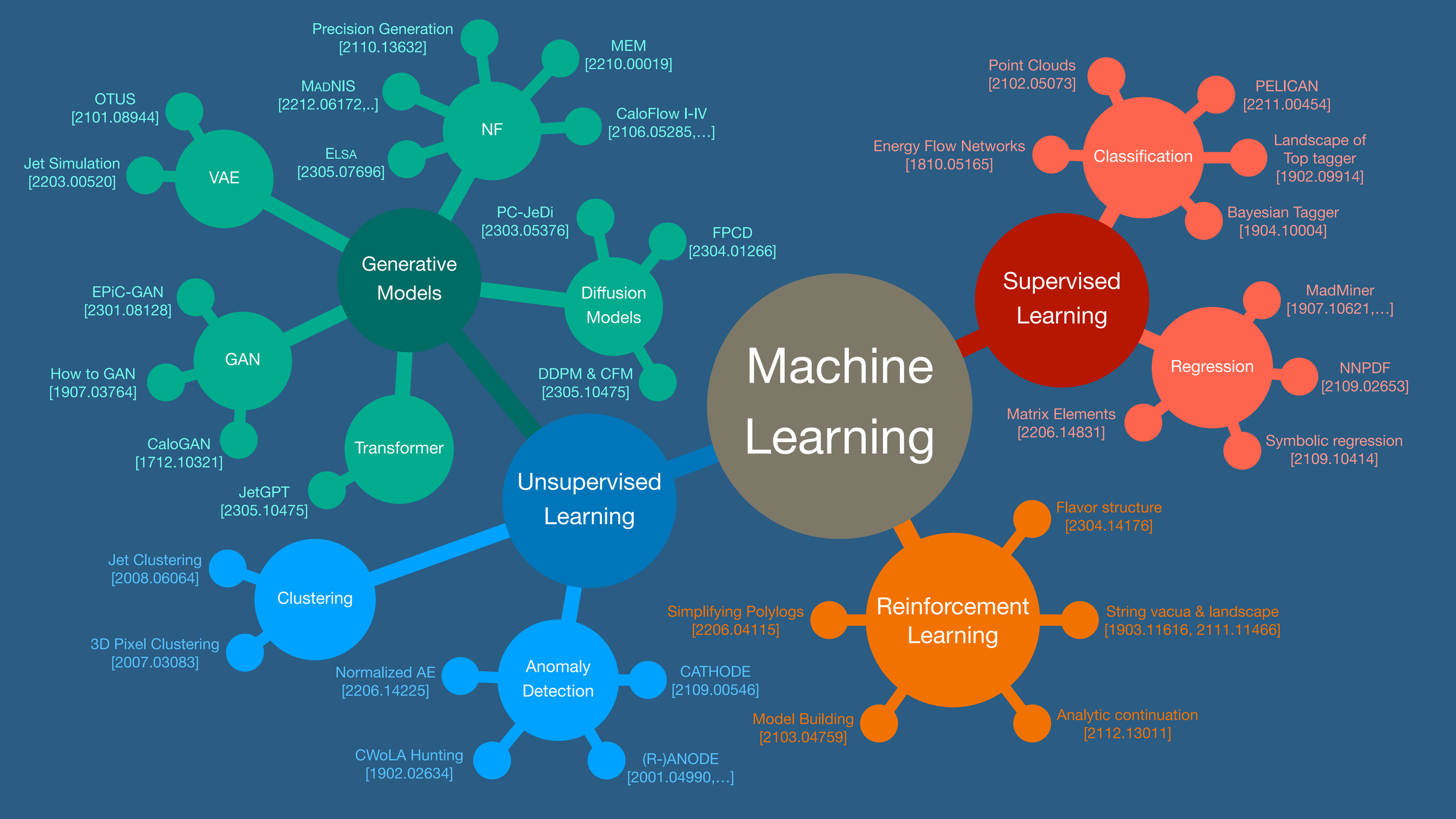
Useful libraries + algorithms

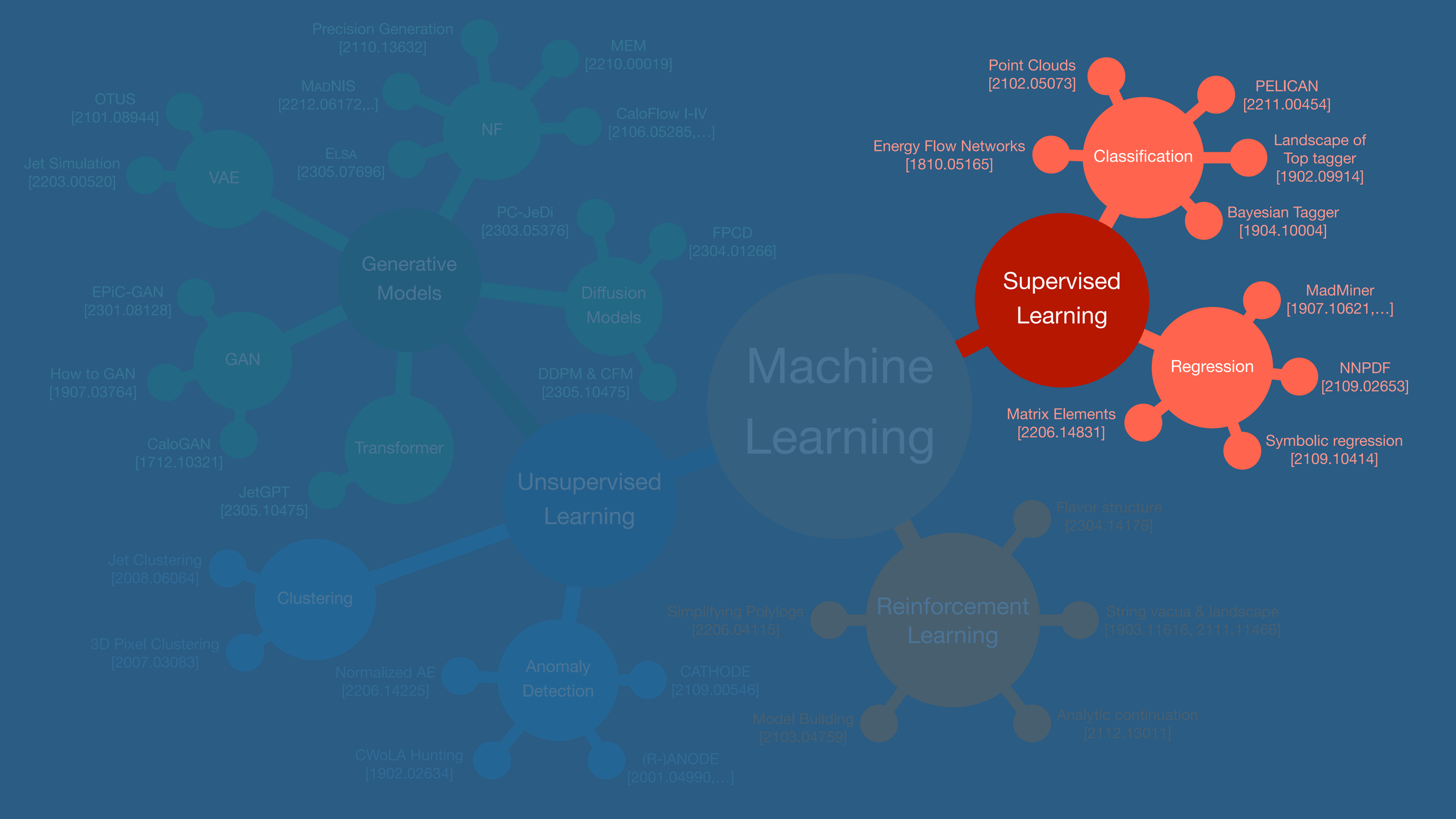
- **scikit-learn:** For most of the “basic” algorithms like Linear Regression and Boosted Decision Trees. Also useful for preprocessing, model combination, etc.

- **XGBoost, LightGBM, CatBoost:** For optimized tree-based models

- **TensorFlow, PyTorch, Jax:** For deep learning models

The Landscape of Machine Learning


Machine Learning
Generative Models
VAE
OTUS [2101.08944]
Jet Simulation [2203.00520]
NF
Precision Generation [2110.13632]
MADNIS [2212.06172,..]
ELSA [2305.07696]
MEM [2210.00019]
CaloFlow I-IV [2106.05285,…]
GAN
EPiC-GAN [2301.08128]
How to GAN [1907.03764]
CaloGAN [1712.10321]
Transformer
JetGPT [2305.10475]
Diffusion Models
PC-JeDi [2303.05376]
FPCD [2304.01266]
DDPM & CFM [2305.10475]
Unsupervised Learning
Clustering
Jet Clustering [2008.06064]
3D Pixel Clustering [2007.03083]
Anomaly Detection
Normalized AE [2206.14225]
CWoLA Hunting [1902.02634]
CATHODE [2109.00546]
(R-)ANODE [2001.04990,…]
Supervised Learning
Classification
Point Clouds [2102.05073]
Energy Flow Networks [1810.05165]
PELICAN [2211.00454]
Landscape of Top tagger [1902.09914]
Bayesian Tagger [1904.10004]
Regression
MadMiner [1907.10621,…]
NNPDF [2109.02653]
Symbolic regression [2109.10414]
Matrix Elements [2206.14831]
Reinforcement Learning
Flavor structure [2304.14176]
Simplifying Polylogs [2206.04115]
String vacua & landscape [1903.11616, 2111.11466]
Model Building [2103.04759]
Analytic continuation [2112.13011]


Machine Learning
Generative Models
NF
Precision Generation [2110.13632]
MEM [2210.00019]
MADNIS [2212.06172,..]
CaloFlow I-IV [2106.05285,…]
ELSA [2305.07696]
VAE
OTUS [2101.08944]
Jet Simulation [2203.00520]
Diffusion Models
PC-JeDi [2303.05376]
FPCD [2304.01266]
DDPM & CFM [2305.10475]
GAN
EPiC-GAN [2301.08128]
How to GAN [1907.03764]
CaloGAN [1712.10321]
Transformer
JetGPT [2305.10475]
Unsupervised Learning
Clustering
Jet Clustering [2008.06064]
3D Pixel Clustering [2007.03083]
Anomaly Detection
Normalized AE [2206.14225]
CATHODE [2109.00546]
CWoLA Hunting [1902.02634]
(R-)ANODE [2001.04990,…]
Reinforcement Learning
Simplifying Polylogs [2206.04115]
Flavor structure [2304.14176]
String vacua & landscape [1903.11616, 2111.11466]
Model Building [2103.04759]
Analytic continuation [2112.13011]
Supervised Learning
Classification
Point Clouds [2102.05073]
PELICAN [2211.00454]
Energy Flow Networks [1810.05165]
Landscape of Top tagger [1902.09914]
Bayesian Tagger [1904.10004]
Regression
MadMiner [1907.10621,…]
NNPDF [2109.02653]
Matrix Elements [2206.14831]
Symbolic regression [2109.10414]

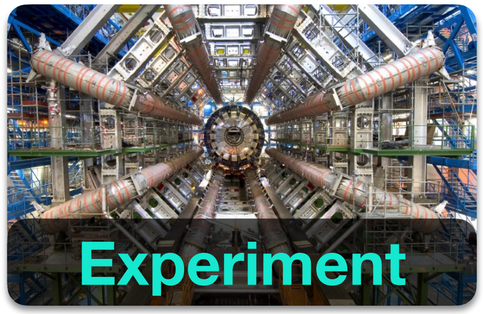
Example I

Regression with MadMiner

Brehmer, Cranmer, Louppe, Pavez [1805.00013, 1805.00020, 1805.12244]

Recap — HEP analysis

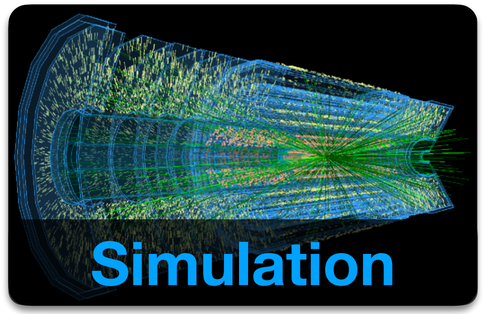
Nature



Detector-level observables

Pattern recognition

Theory



Detector-level observables

Recap — HEP analysis

Theory parameters

θ

Evolution

Recap — HEP analysis

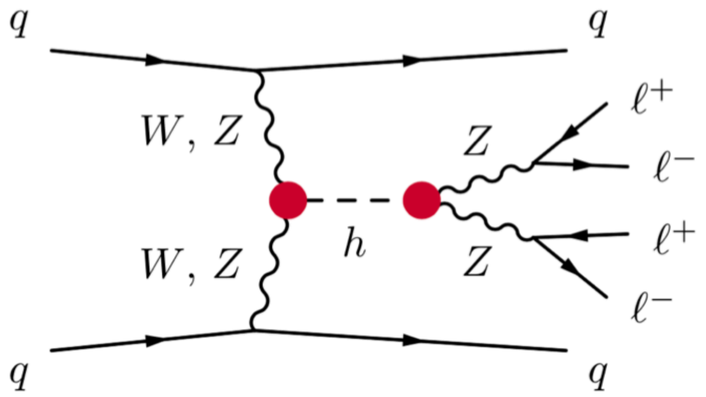
Latent variables

Parton-level momenta

z_p

Theory parameters

θ



Evolution

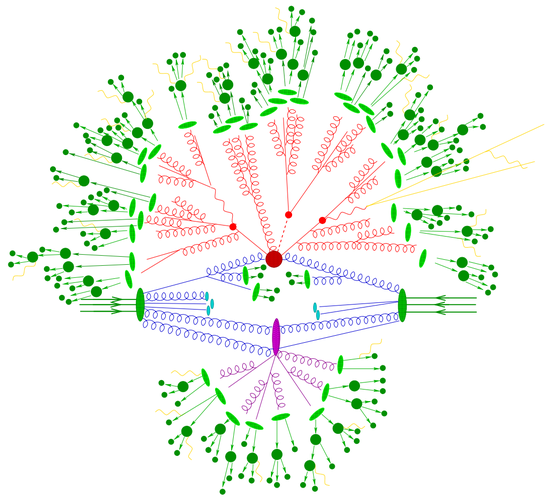
Recap — HEP analysis

Latent variables

Shower splittings z_s

Parton-level momenta z_p

Theory parameters θ



Evolution

Recap — HEP analysis

Latent variables

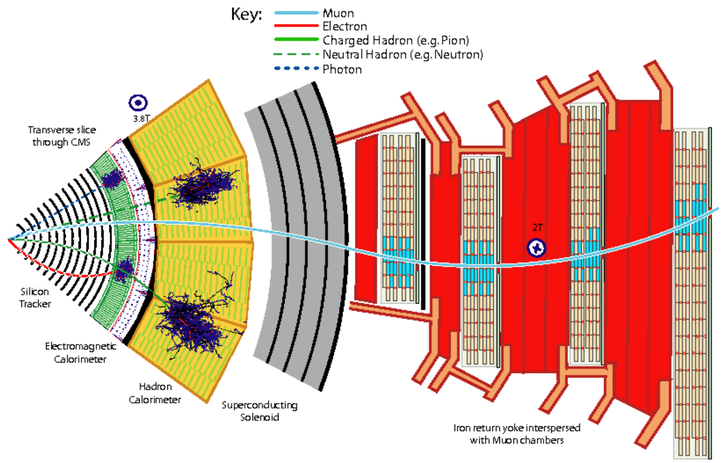
Detector Interactions

Shower splittings

Parton-level momenta

Theory parameters

z_d ← z_s ← z_p ← θ



Evolution

Recap — HEP analysis

Latent variables

Observables

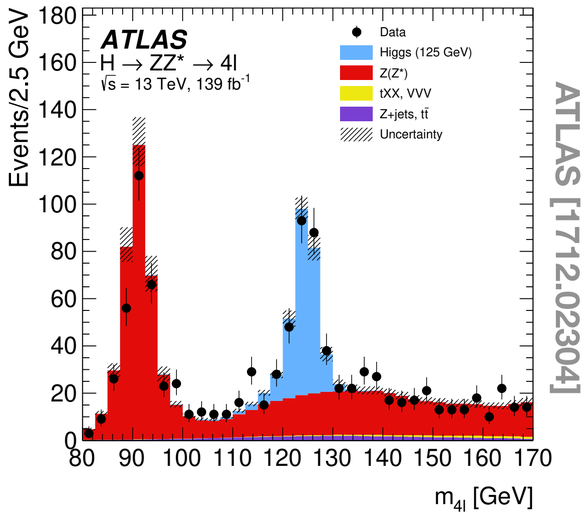
Detector Interactions

Shower splittings

Parton-level momenta

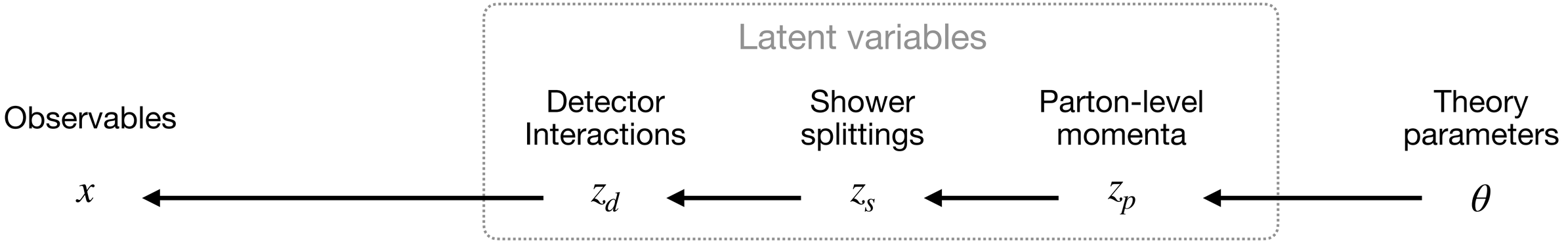
Theory parameters

x ← z_d ← z_s ← z_p ← θ



Evolution

Recap — HEP analysis



Sample from $p(x \mid z_d)$ $p(z_d \mid z_s)$ $p(z_s \mid z_p)$ $p(z_p \mid \theta)$

GEANT4 A SIMULATION TOOLKIT

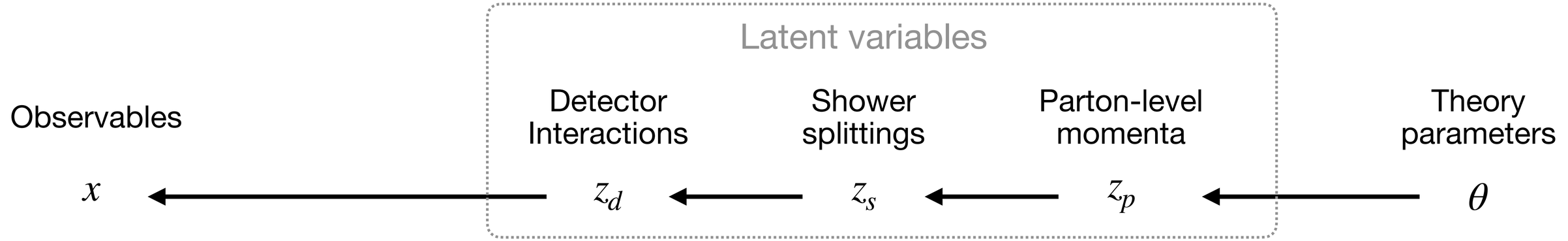
DELPHES fast simulation

MADGRAPH5 aMC@NLO

SHERPA

Prediction (Simulation)

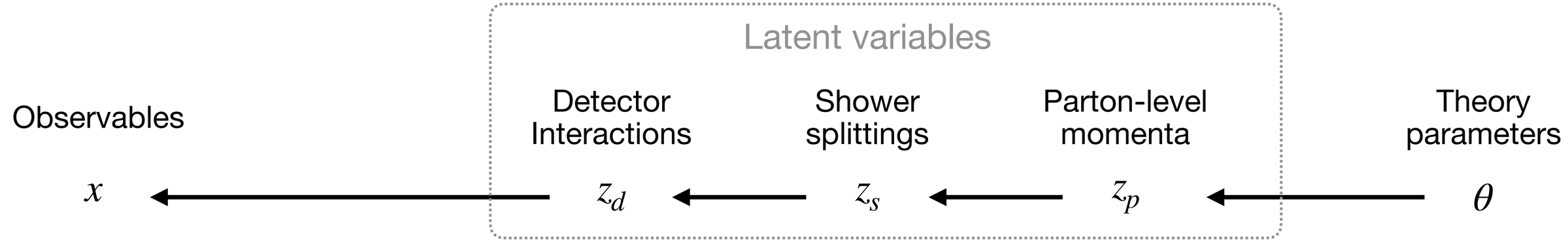
Recap — HEP analysis



$$p(x\,|\,\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \quad p(x\,|\,z_d) \qquad p(z_d\,|\,z_s) \qquad p(z_s\,|\,z_p) \qquad p(z_p\,|\,\theta)$$

Inference

Recap — HEP analysis

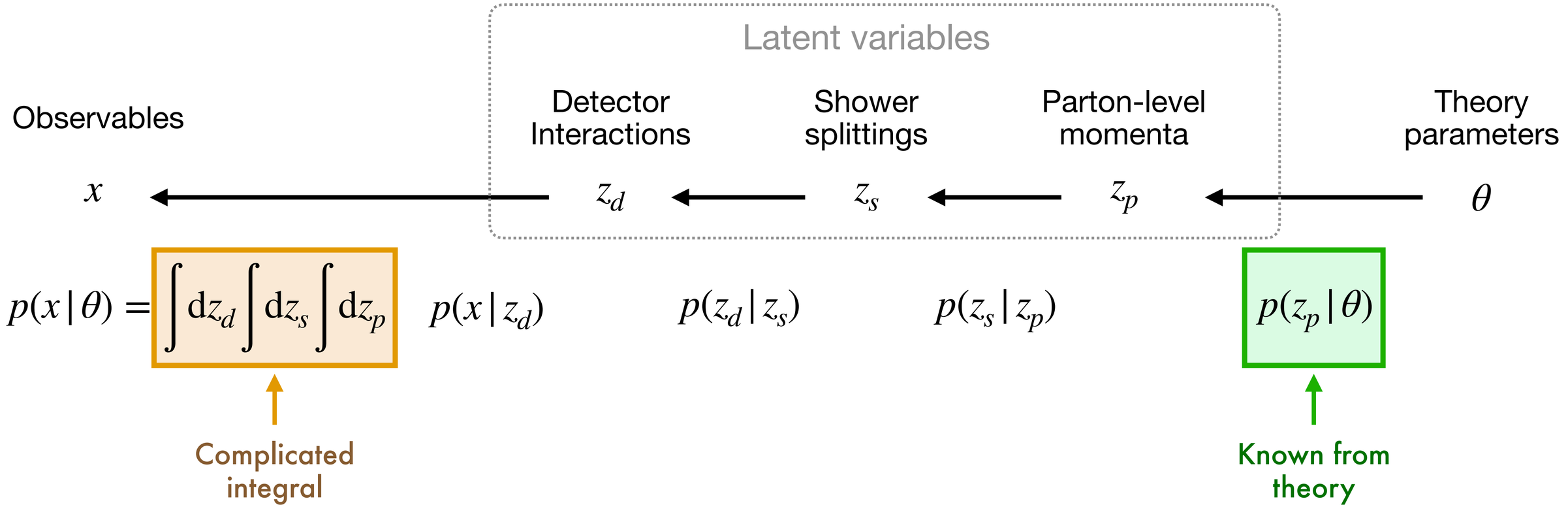


$$p(x\,|\,\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \quad p(x\,|\,z_d) \qquad p(z_d\,|\,z_s) \qquad p(z_s\,|\,z_p) \qquad p(z_p\,|\,\theta)$$

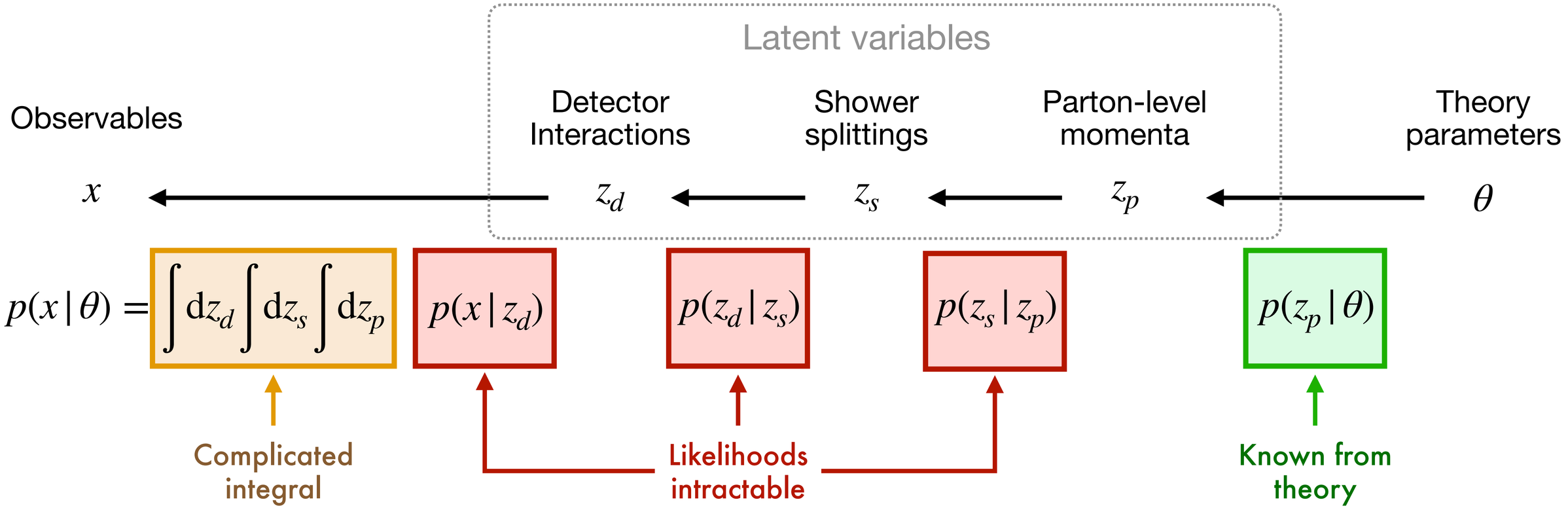
Known from theory

Inference

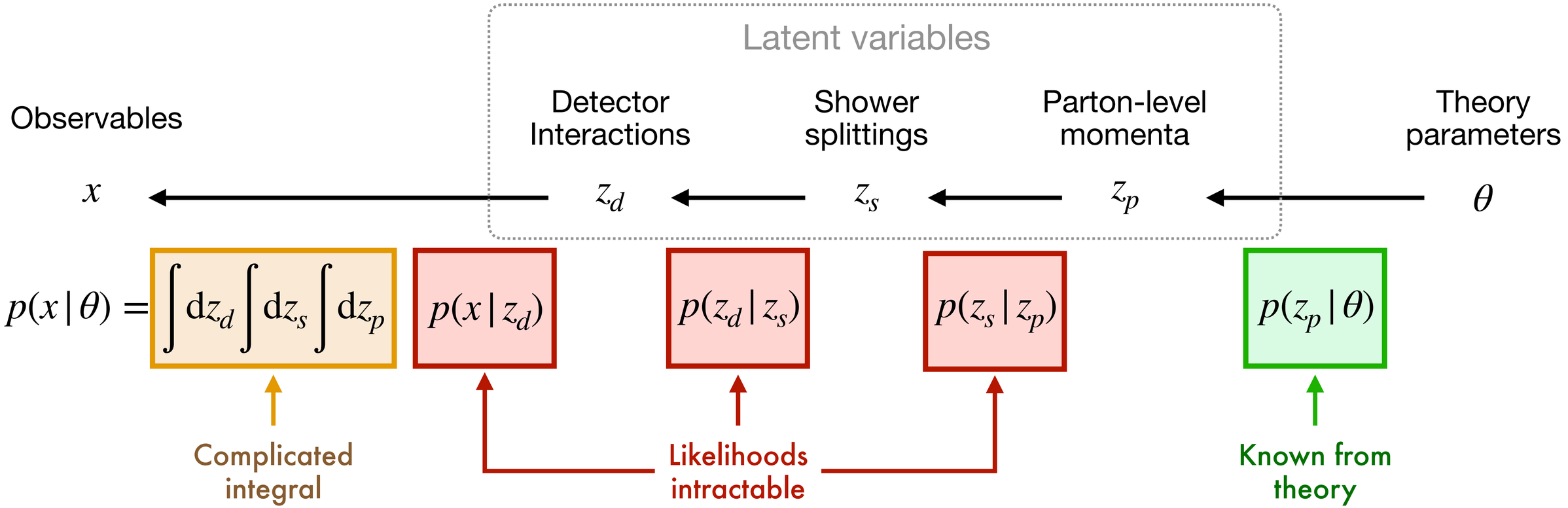
Recap — HEP analysis



Recap — HEP analysis



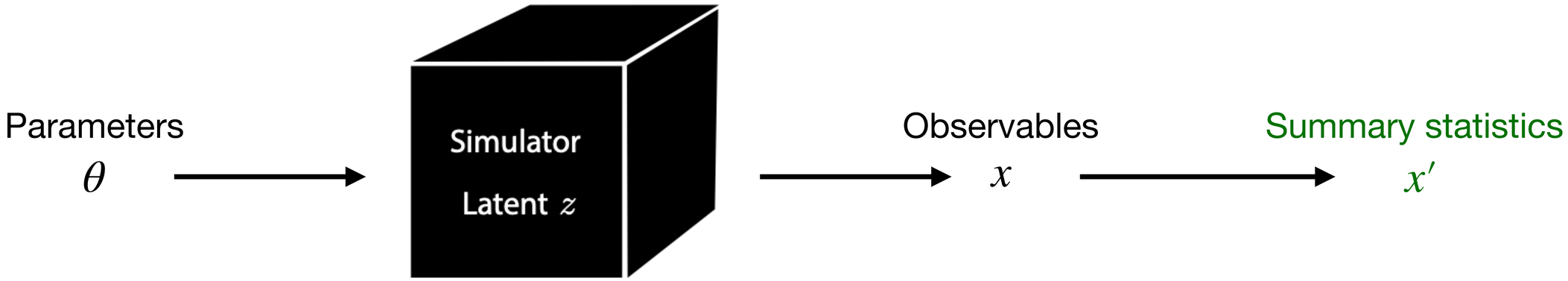
Recap — HEP analysis



Why has that not stopped us so far?

Inference

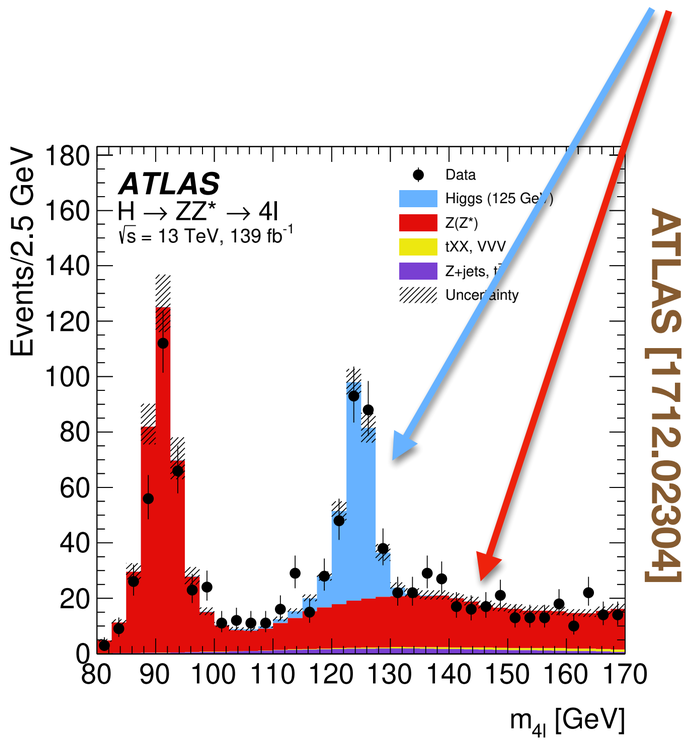
Solve it by histogramming summary statistics



Solve it by histogramming summary statistics

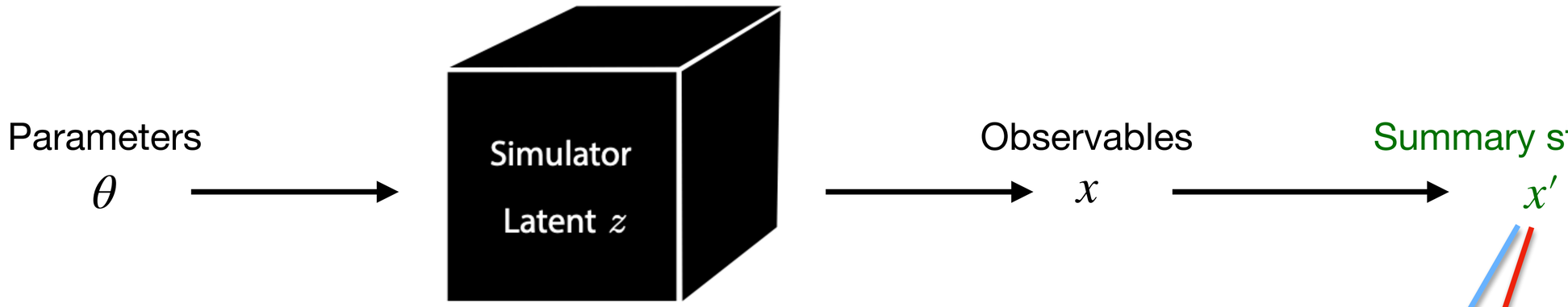
Parameters θ → Simulator, Latent z → Observables x → Summary statistics x'

$$\hat{p}(x \mid \theta) = p(x' \mid \theta) =$$

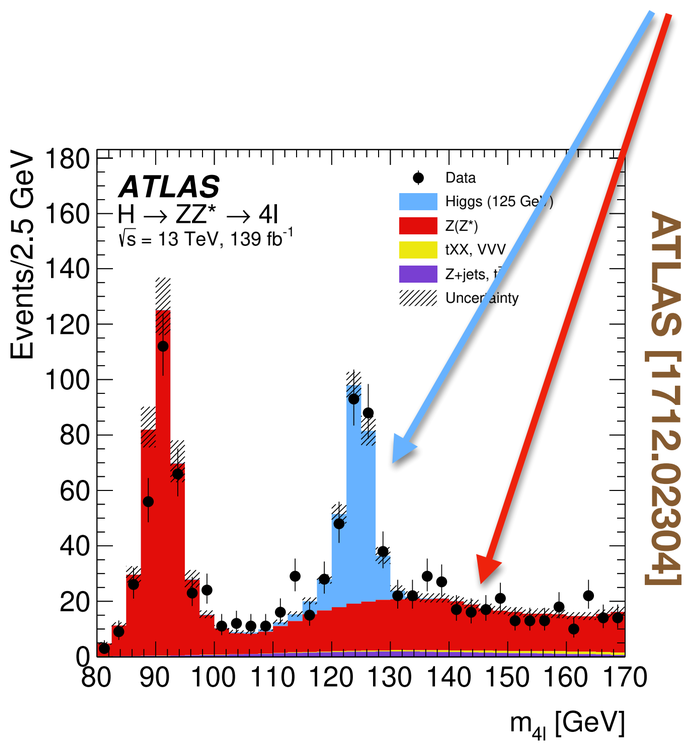


Run simulator for different θ, fill histograms

Solve it by histogramming summary statistics



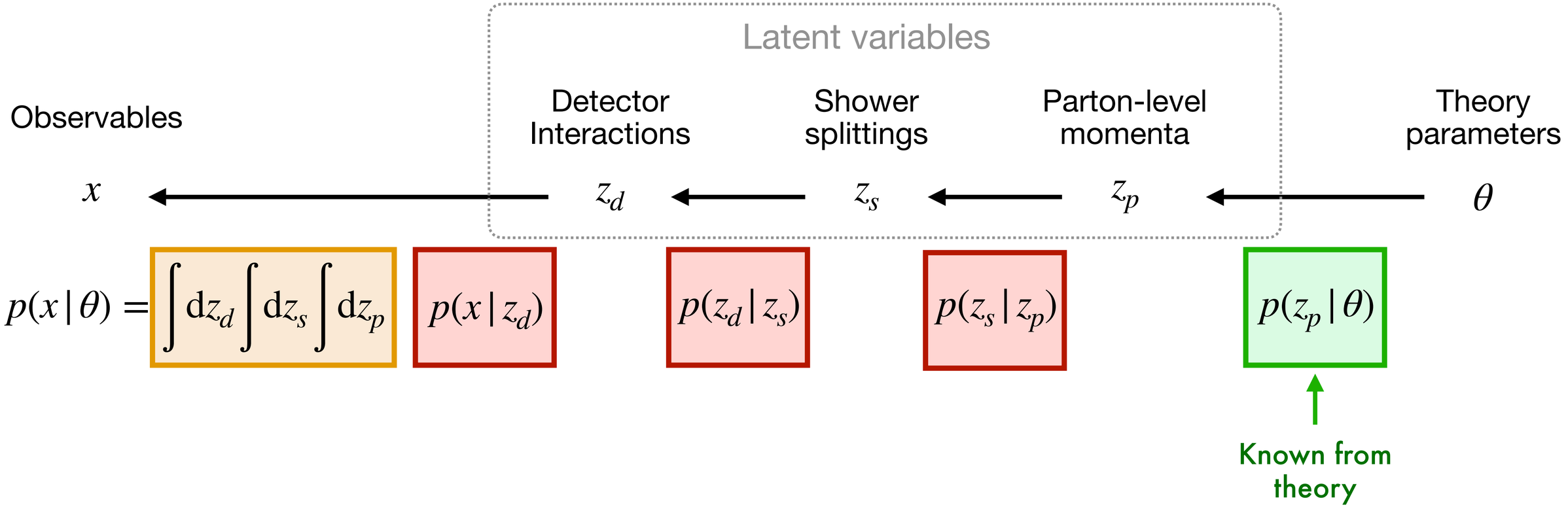
$$\hat{p}(x \mid \theta) = p(x' \mid \theta) =$$



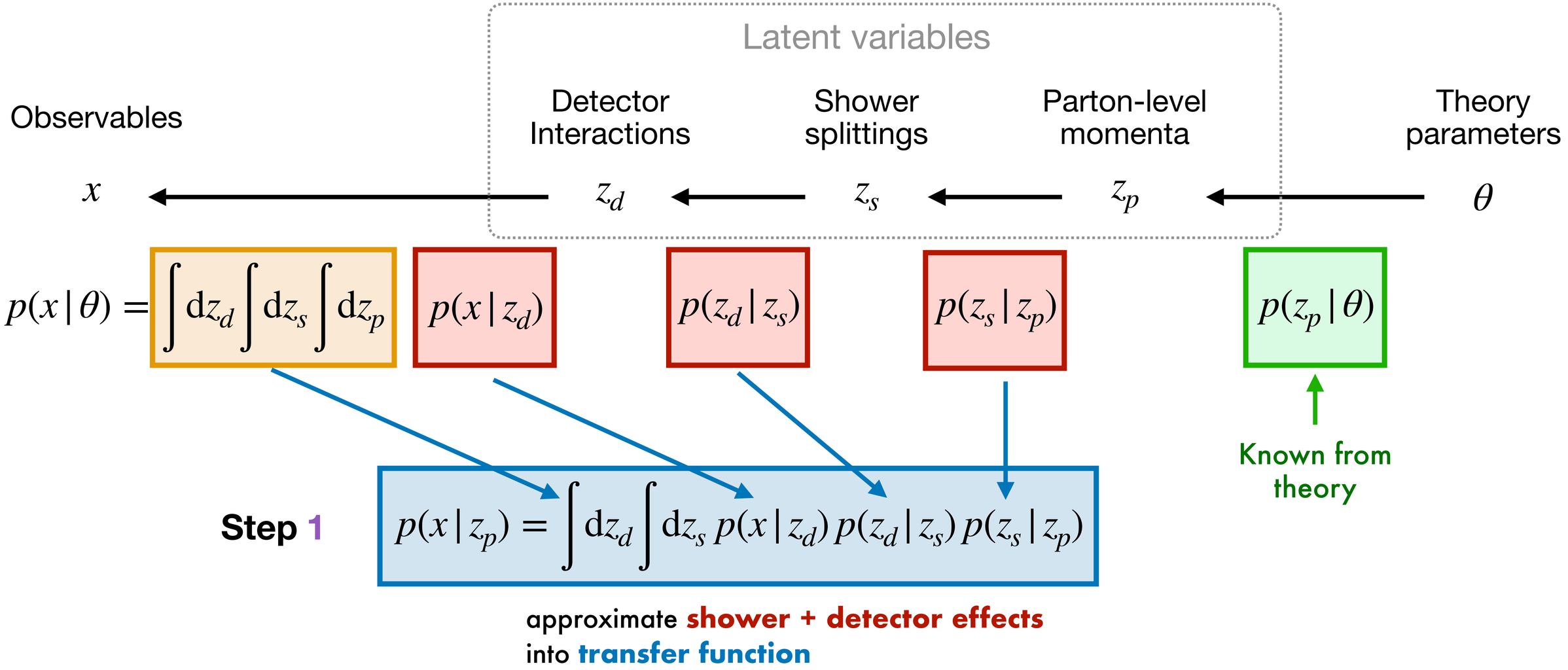
Run simulator for different θ, fill histograms

- How to choose x' ? Standard variables often lose information **[1612.05261,1712.02350]**
- "Curse of dimensionality": Histograms don't scale to high-dimensional x

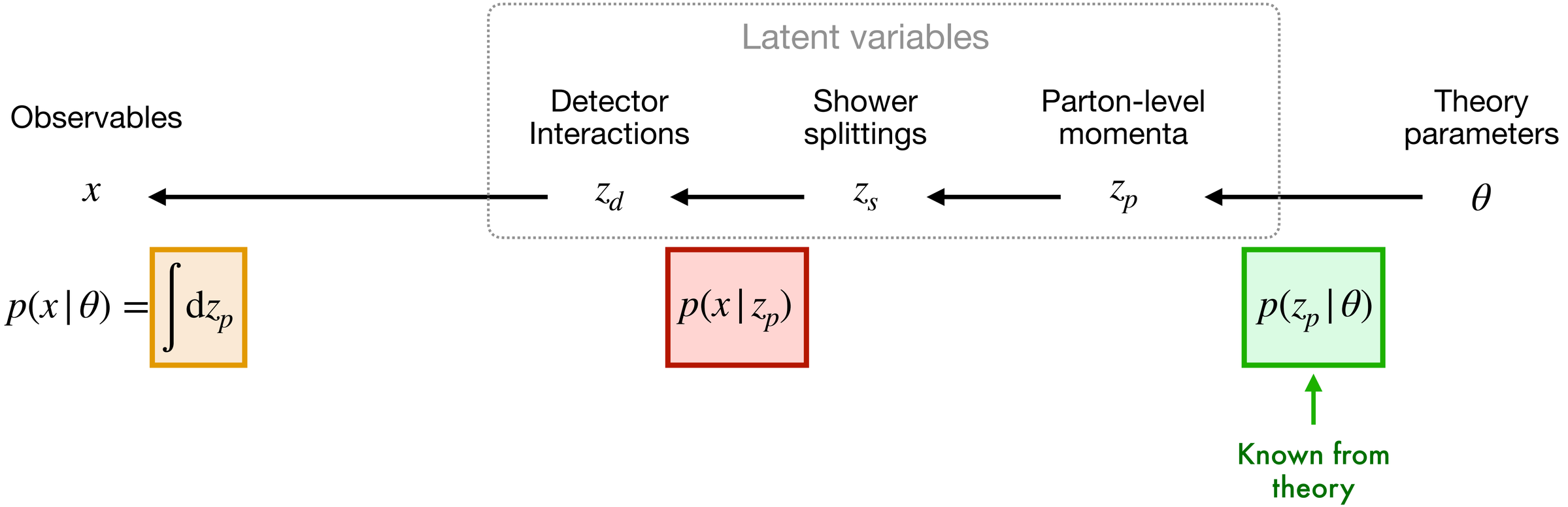
Solve it by calculating the integral with ML



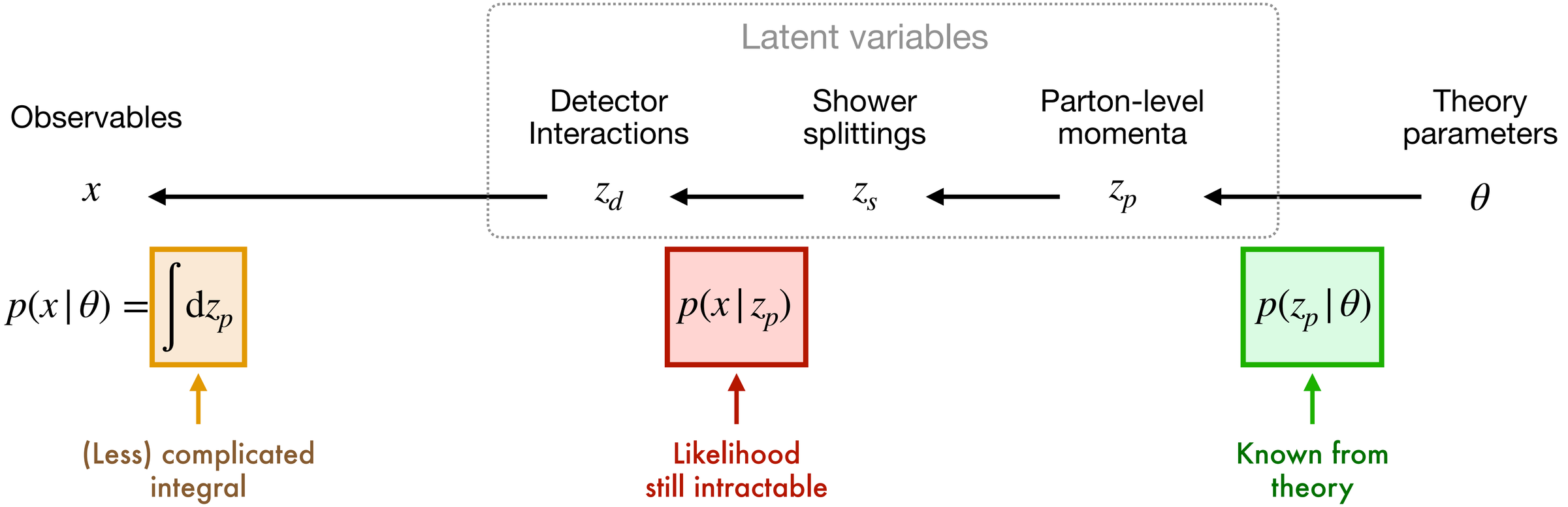
Solve it by calculating the integral with ML



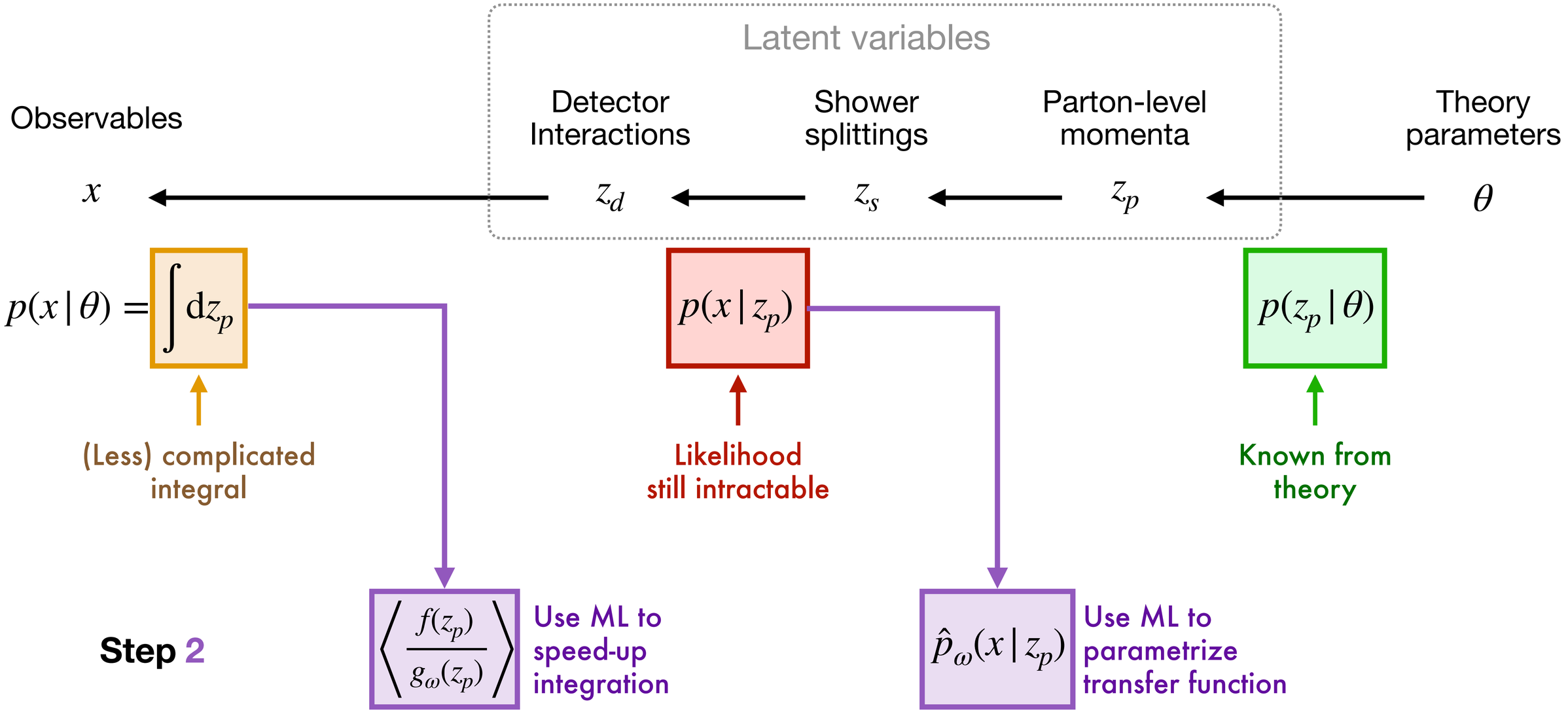
Solve it by calculating the integral with ML



Solve it by calculating the integral with ML



Solve it by calculating the integral with ML



Solve it by calculating the integral with ML

Latent variables

Observables | Detector Interactions | Shower splittings | Parton-level momenta | Theory parameters

θ

Matrix Element Method [K. Kondo 1988]

$$p(x\,|\,\theta) = \int \mathrm{d}z_p \; p(x\,|\,z_p)\; p(z_p\,|\,\theta)$$

(Less) complicated integral

Likelihood still intractable

Known from theory

Step 2

$\left\langle \frac{f(z_p)}{g_\omega(z_p)} \right\rangle$ Use ML to speed-up integration

$\hat{p}_\omega(x\,|\,z_p)$ Use ML to parametrize transfer function

Solve it by calculating the integral with ML

Latent variables

Observables

Detector Interactions

Shower splittings

Parton-level momenta

Theory

Lecture II

Matrix Element Method [K. Kondo 1988]

$$p(x|\theta) = \int \mathrm{d}z_p \; p(x|z_p) \; p(z_p|\theta)$$

(Less) complicated integral

Likelihood still intractable

Known from theory

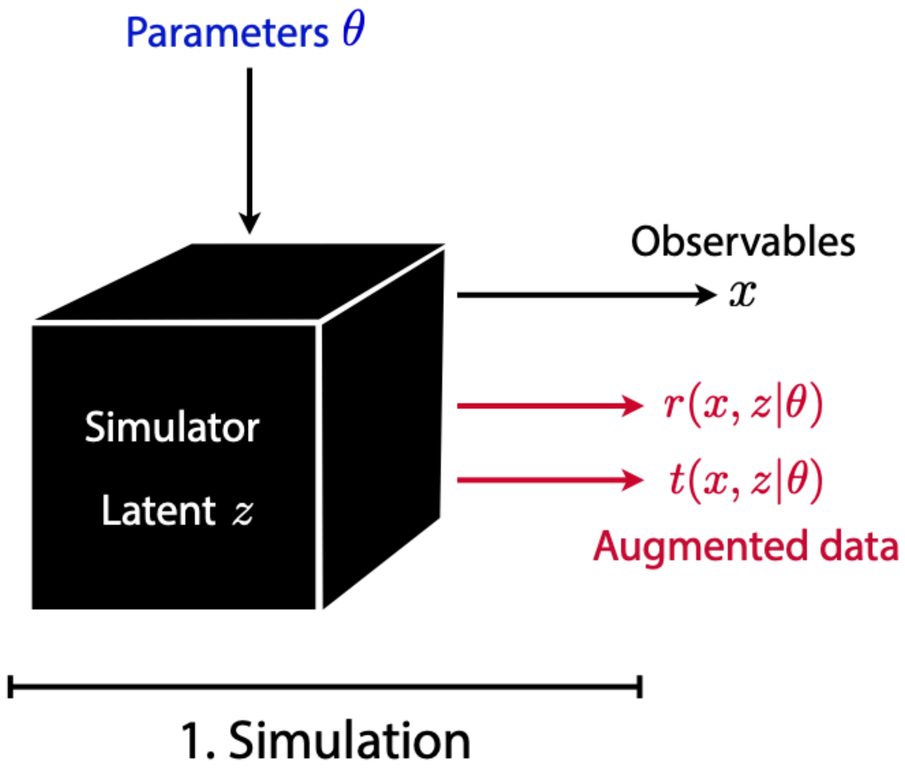
Step 2

$\left\langle \frac{f(z_p)}{g_\omega(z_p)} \right\rangle$ Use ML to speed-up integration

$\hat{p}_\omega(x|z_p)$ Use ML to parametrize transfer function

Solving it with likelihood-free inference

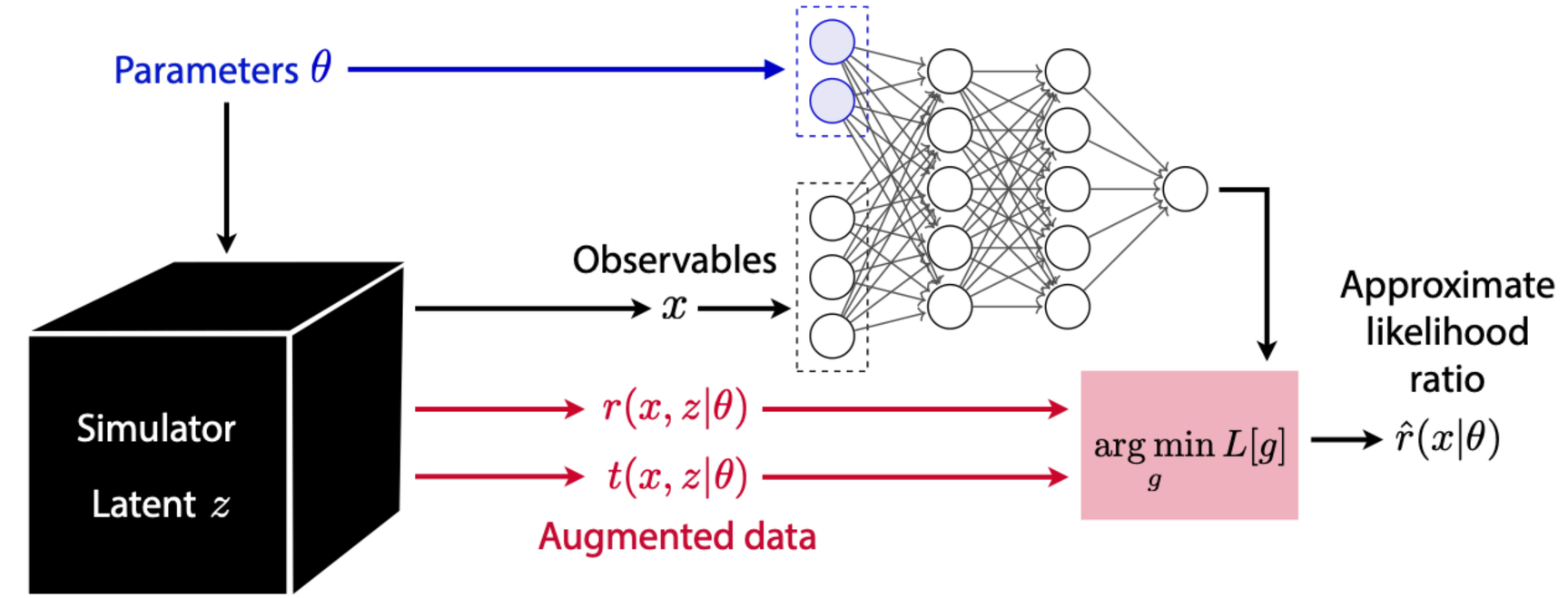
The MadMiner approach



"Mining gold": Extract additional information from simulator

Brehmer, Cranmer, Louppe, Pavez [1805.00013, 1805.00020, 1805.12244]

The MadMiner approach

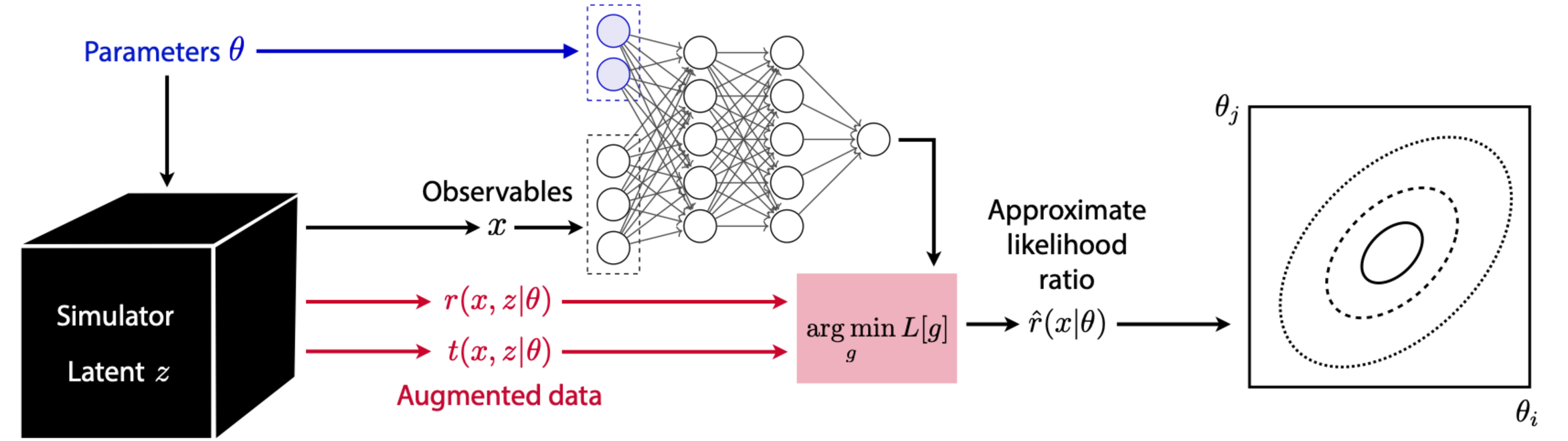


"Mining gold": Extract additional information from simulator

Use this information to train estimator for likelihood ratio

Brehmer, Cranmer, Louppe, Pavez [1805.00013, 1805.00020, 1805.12244]

The MadMiner approach



1. Simulation

"Mining gold": Extract additional information from simulator

2. Machine Learning

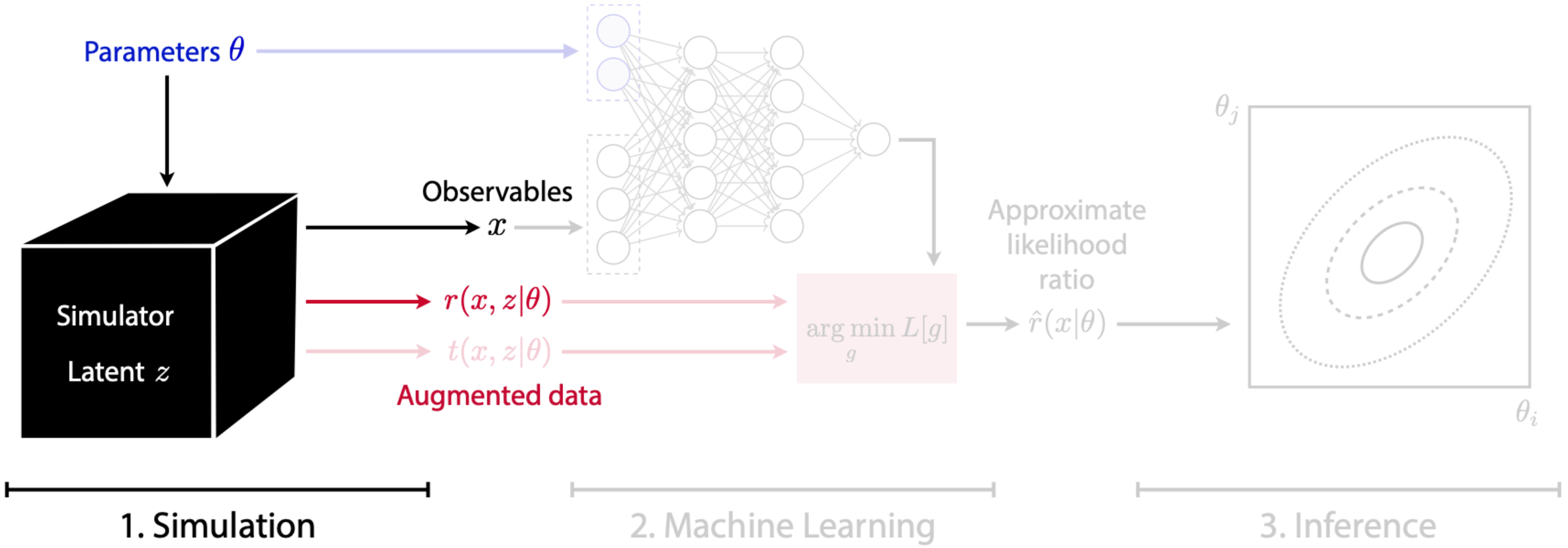
Use this information to train estimator for likelihood ratio

3. Inference

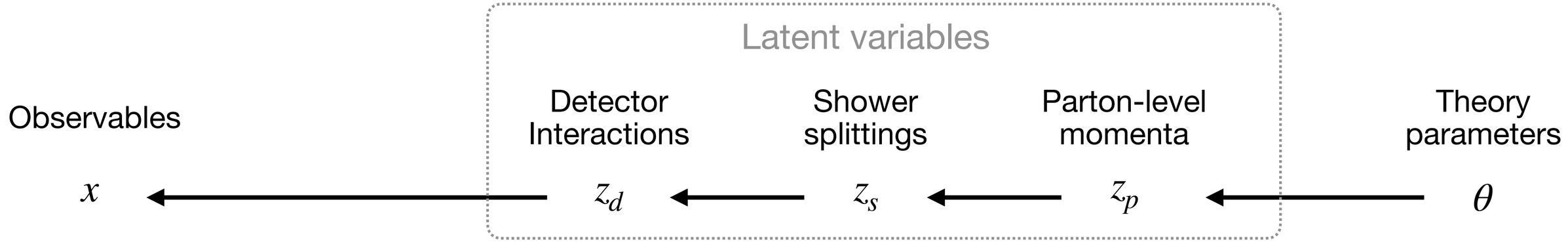
Set new constrains with standard hypothesis tests

Brehmer, Cranmer, Louppe, Pavez [1805.00013, 1805.00020, 1805.12244]

The MadMiner approach



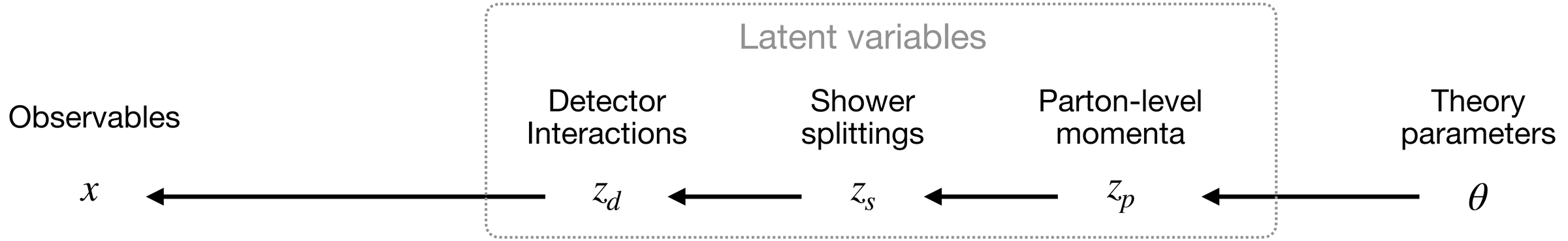
Mining gold from the simulator



$$p(x\,|\,\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \quad p(x\,|\,z_d) \quad p(z_d\,|\,z_s) \quad p(z_s\,|\,z_p) \quad p(z_p\,|\,\theta)$$

Intractable integrals

Mining gold from the simulator



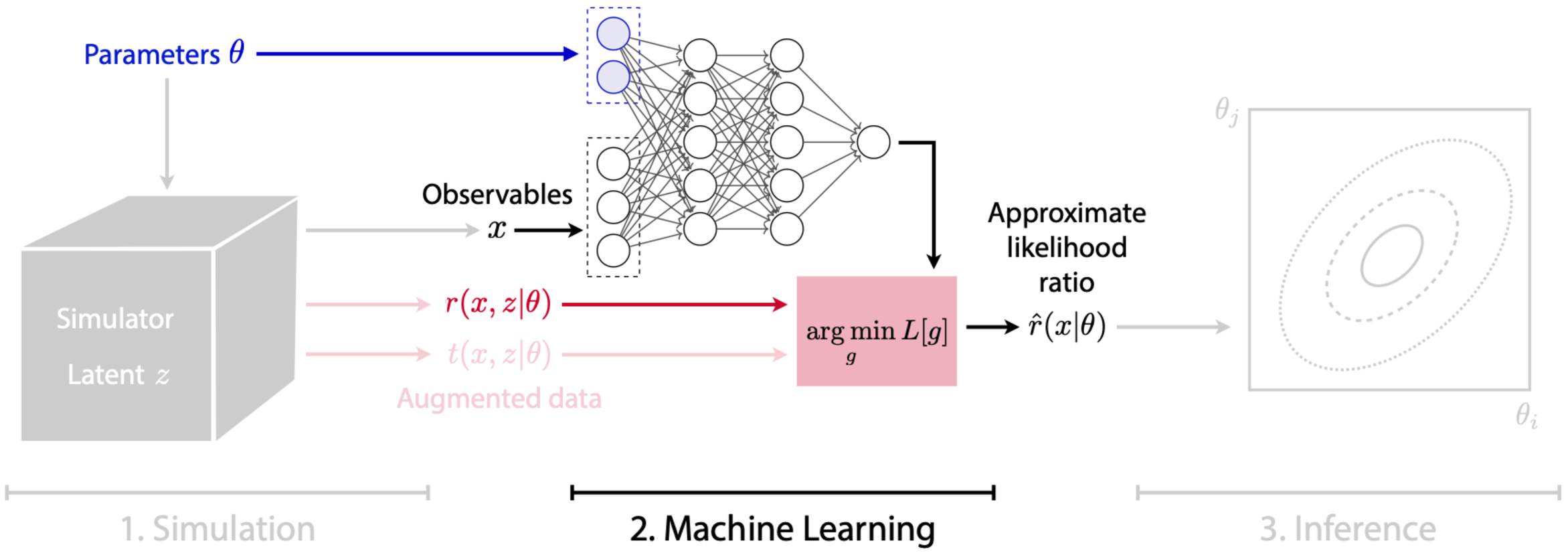
$$p(x\,|\,\theta) = \int \mathrm{d}z_d \int \mathrm{d}z_s \int \mathrm{d}z_p \quad p(x\,|\,z_d) \qquad p(z_d\,|\,z_s) \qquad p(z_s\,|\,z_p) \qquad \boxed{p(z_p\,|\,\theta)}$$

Parton-level likelihood can be evaluated!

⇒ For each generated event, we can calculate the **joint likelihood ratio** conditional on its evolution:

$$r(x, z\,|\,\theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p\,|\,\theta_0)}{p(x, z_d, z_s, z_p\,|\,\theta_1)} = \frac{p(x\,|\,z_d)}{p(x\,|\,z_d)} \frac{p(z_d\,|\,z_s)}{p(z_d\,|\,z_s)} \frac{p(z_s\,|\,z_p)}{p(z_s\,|\,z_p)} \boxed{\frac{p(z_p\,|\,\theta_0)}{p(z_p\,|\,\theta_1)}}$$

The MadMiner approach



The value of gold

We can calculate the **joint likelihood ratio**

$$r(x, z \,|\, \theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p \,|\, \theta_0)}{p(x, z_d, z_s, z_p \,|\, \theta_1)}$$

We want the **likelihood ratio function**

$$r(x \,|\, \theta_0, \theta_1) \equiv \frac{p(x \,|\, \theta_0)}{p(x \,|\, \theta_1)}$$

The value of gold

We can calculate the **joint likelihood ratio**

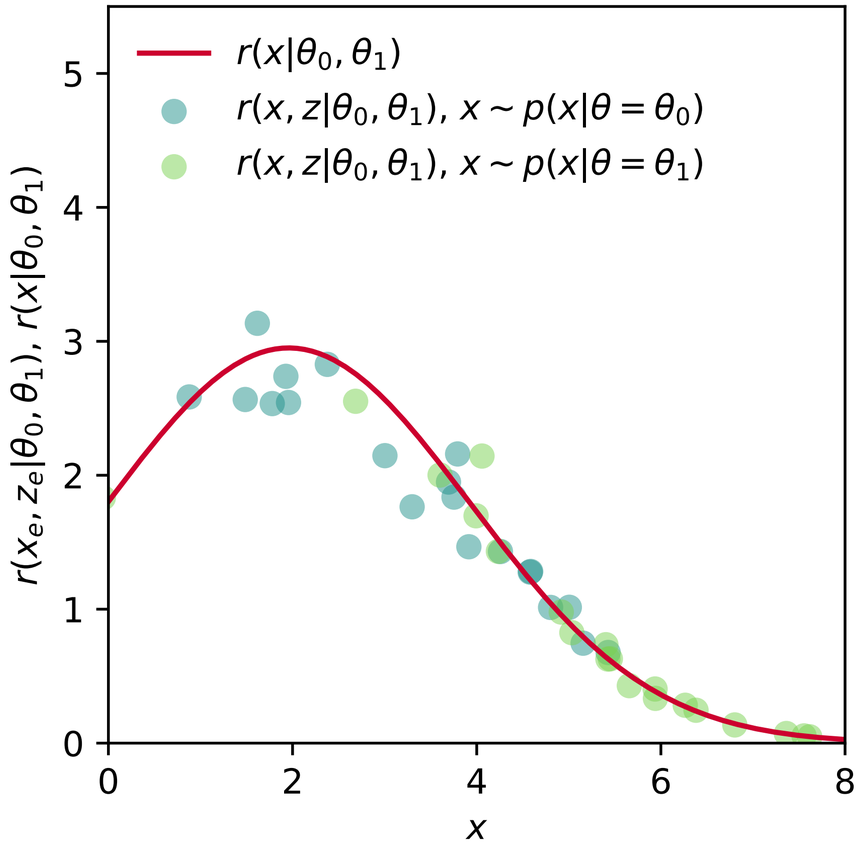
$$r(x, z \,|\, \theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p \,|\, \theta_0)}{p(x, z_d, z_s, z_p \,|\, \theta_1)}$$

We want the **likelihood ratio function**

$$r(x \,|\, \theta_0, \theta_1) \equiv \frac{p(x \,|\, \theta_0)}{p(x \,|\, \theta_1)}$$

$r(x, z \,|\, \theta_0, \theta_1)$ are scattered around $r(x \,|\, \theta_0, \theta_1)$

[1805.00020]



The value of gold

We can calculate the **joint likelihood ratio**

$$r(x, z \,|\, \theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p \,|\, \theta_0)}{p(x, z_d, z_s, z_p \,|\, \theta_1)}$$

We want the **likelihood ratio function**

$$r(x \,|\, \theta_0, \theta_1) \equiv \frac{p(x \,|\, \theta_0)}{p(x \,|\, \theta_1)}$$

With $r(x, z \,|\, \theta_0, \theta_1)$, we define the functional

$$L_r[\hat{r}(x \,|\, \theta_0, \theta_1)] = \int \mathrm{d}x \int \mathrm{d}z \, p(x, z \,|\, \theta_1) \, \left[\hat{r}(x \,|\, \theta_0, \theta_1) - r(x, z \,|\, \theta_0, \theta_1)\right]^2$$

One can show it is minimized by

$$\hat{r}(x \,|\, \theta_0, \theta_1) = r(x \,|\, \theta_0, \theta_1)$$

Machine learning = applied calculus

We can get a precise estimator by numerically minimizing a functional:

$$\hat{r}(x \mid \theta_0, \theta_1) = \underset{\hat{r}(x \mid \theta_0, \theta_1)}{\arg\min} \underbrace{\int \mathrm{d}x \int \mathrm{d}z \, p(x, z \mid \alpha_1) \left[\hat{r}(x \mid \theta_0, \theta_1) - r(x, z \mid \theta_0, \theta_1)\right]^2}_{L_r[\hat{r}(x \mid \theta_0, \theta_1)]}$$

Machine learning = applied calculus

We can get a precise estimator by numerically minimizing a functional:

$$\hat{r}(x \mid \theta_0, \theta_1) = \underset{\hat{r}(x \mid \theta_0, \theta_1)}{\arg\min} \underbrace{\int \mathrm{d}x \int \mathrm{d}z \, p(x, z \mid \alpha_1) \left[\hat{r}(x \mid \theta_0, \theta_1) - r(x, z \mid \theta_0, \theta_1)\right]^2}_{L_r[\hat{r}(x \mid \theta_0, \theta_1)]}$$

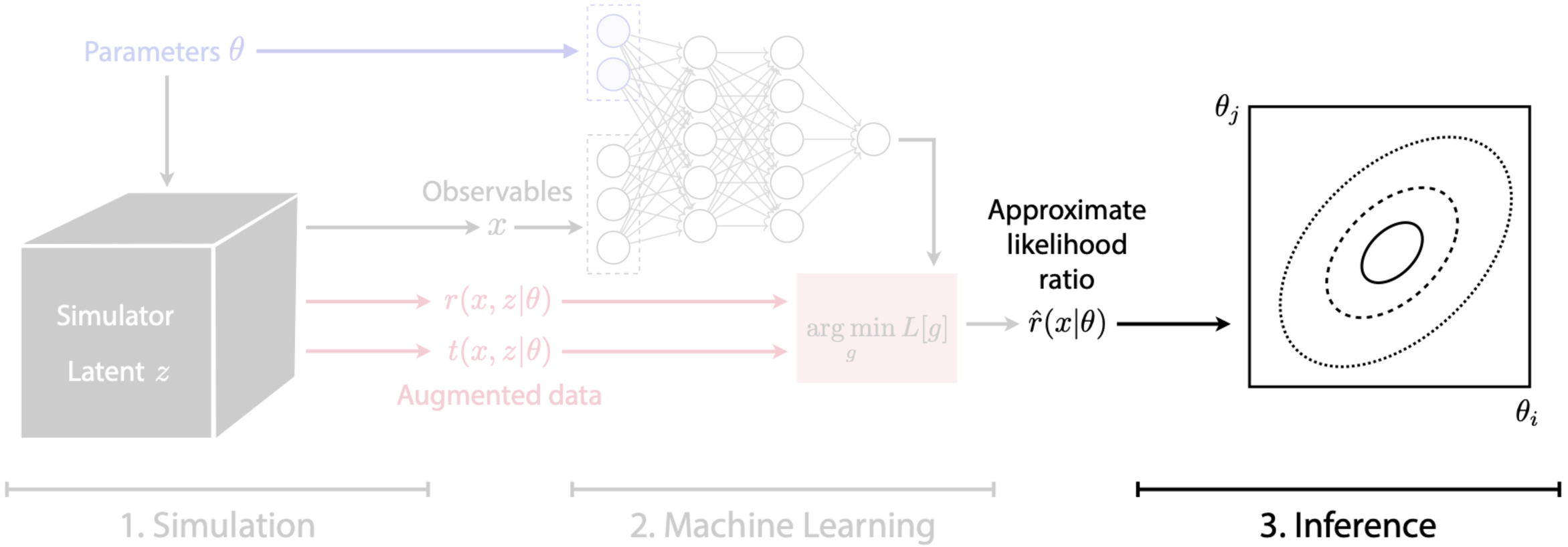
We do this via **Machine Learning:**

- Functional L_r → Loss function

$$\mathscr{L}_{\mathrm{MSE}} = \frac{1}{N} \sum_i \left[\hat{r}_\omega(x_i \mid \theta_0, \theta_1) - r(x_i, z_i \mid \theta_0, \theta_1)\right]^2$$

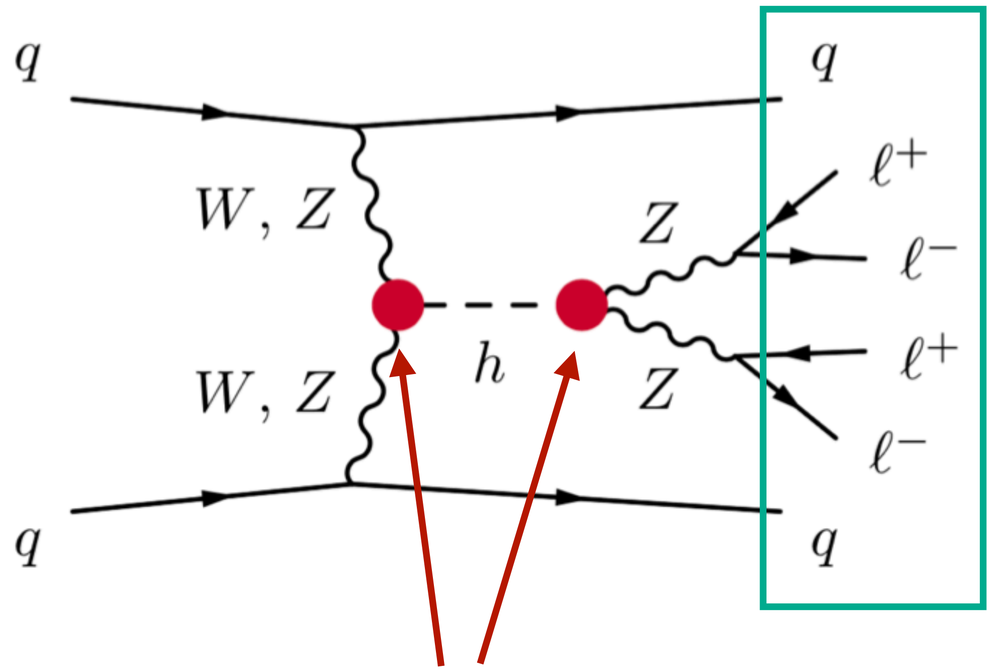
- Variational familiy $\hat{r}_\omega(x \mid \theta_0, \theta_1)$ → Flexible parametric function (e.g. neural network)
- Exact minimization → Numerical optimization algorithm (eg. Stochastic gradient descent)

The MadMiner approach



Constraining EFT parameters with ML

Higgs production in weak boson fusion:

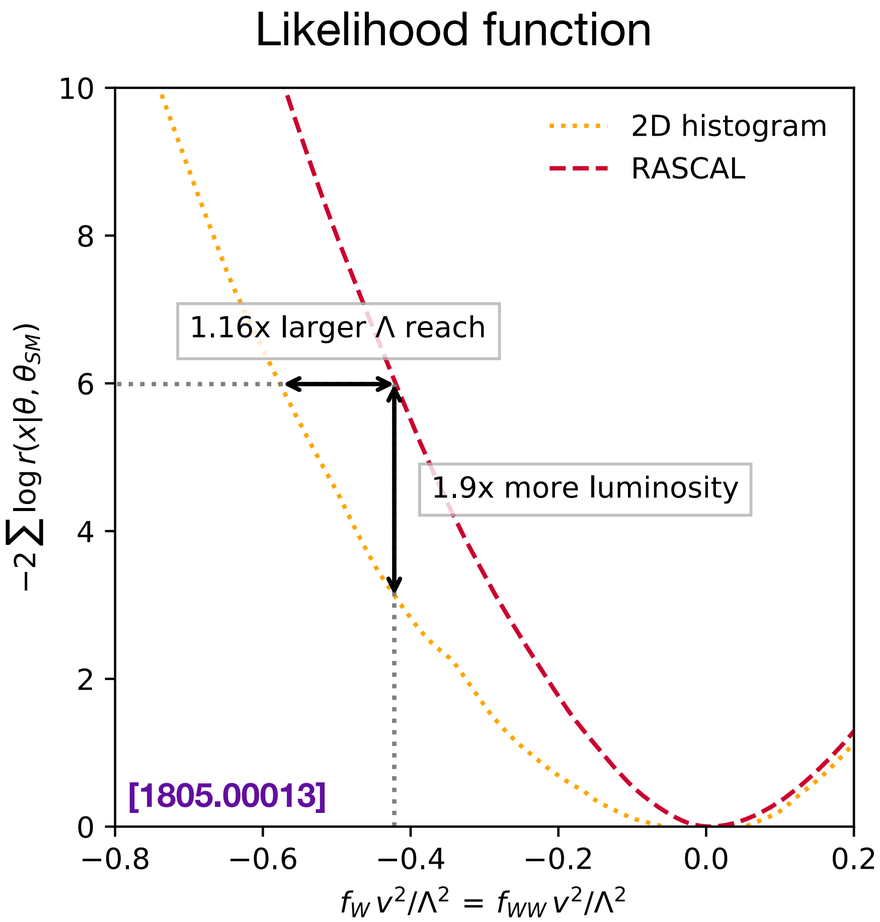


at least 16-dimensional observable space

Exciting new physics might hide here!
We parameterize it with two EFT coefficients

$$\mathcal{L} = \mathcal{L}_{\rm SM} + \boxed{\frac{f_W}{\Lambda^2}} \frac{ig}{2} (D^\mu \phi)^\dagger \sigma^a D^\nu \phi \, W^a_{\mu\nu} - \boxed{\frac{f_{WW}}{\Lambda^2}} \frac{g^2}{4} (\phi^\dagger \phi) \, W^a_{\mu\nu} W^{\mu\nu\, a}$$

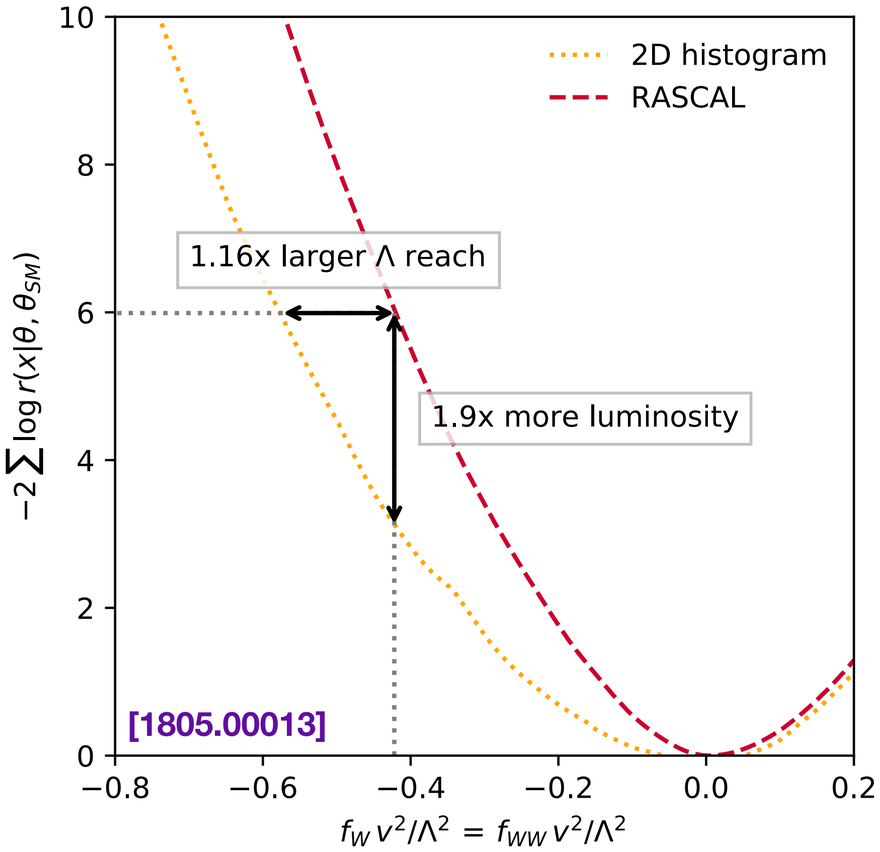
Better sensitivity to new physics



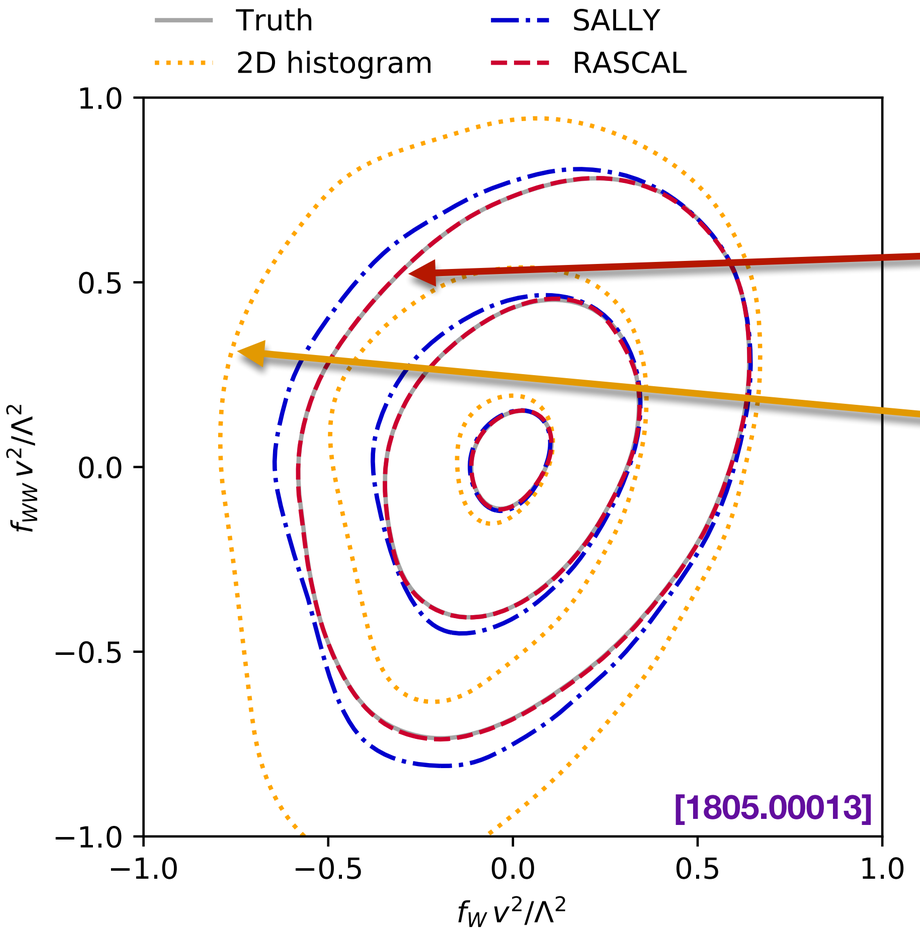
Results are based on 36 observed events, assuming SM

Better sensitivity to new physics

Likelihood function



Expected exclusion limits at 68%, 95%, 99.7% CL

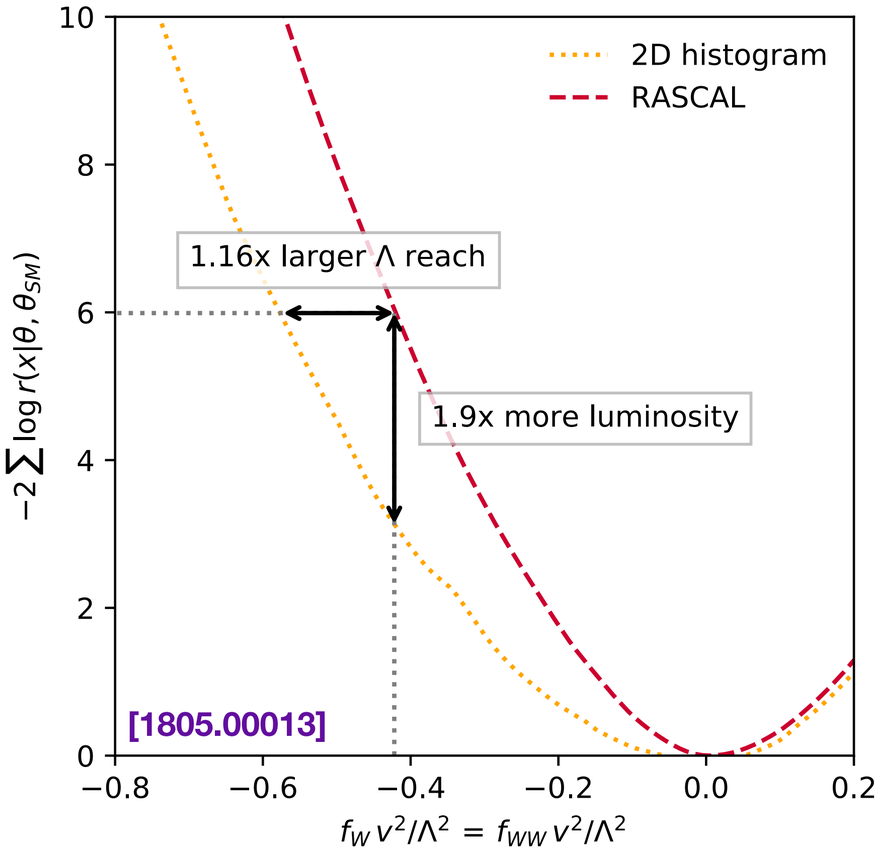


MadMiner enables stronger limits than **2D histogram**

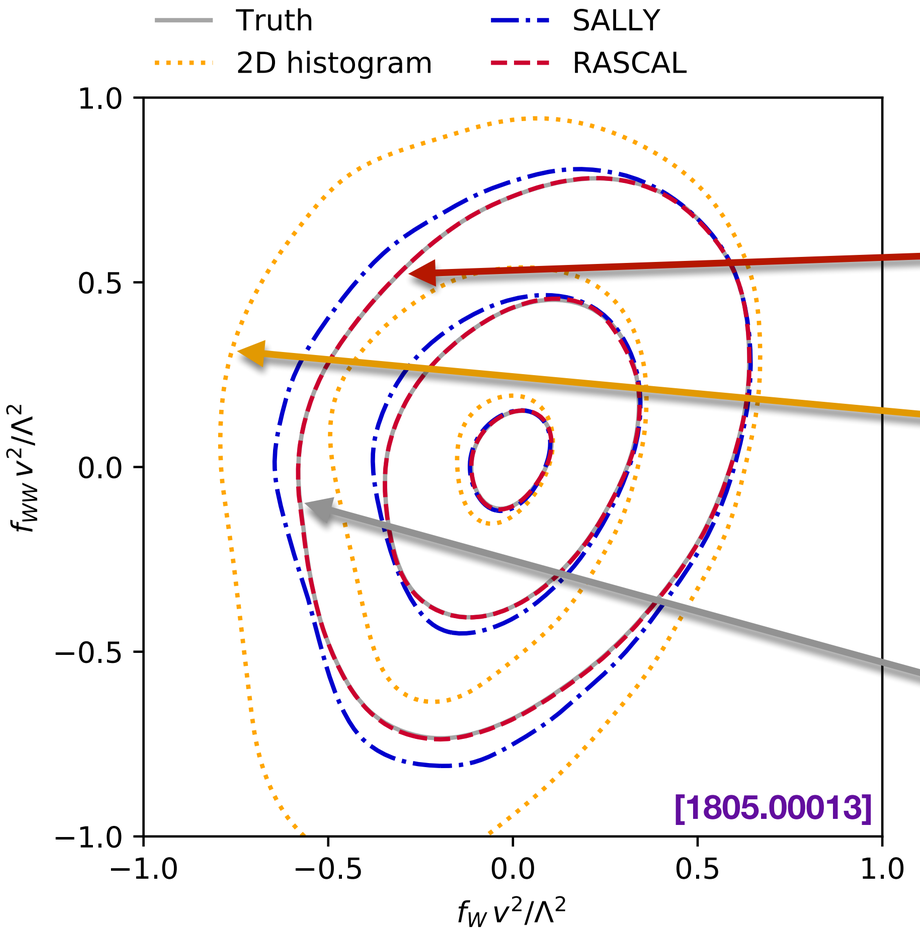
Results are based on 36 observed events, assuming SM

Better sensitivity to new physics

Likelihood function



Expected exclusion limits at 68%, 95%, 99.7% CL



MadMiner enables stronger limits than **2D histogram**

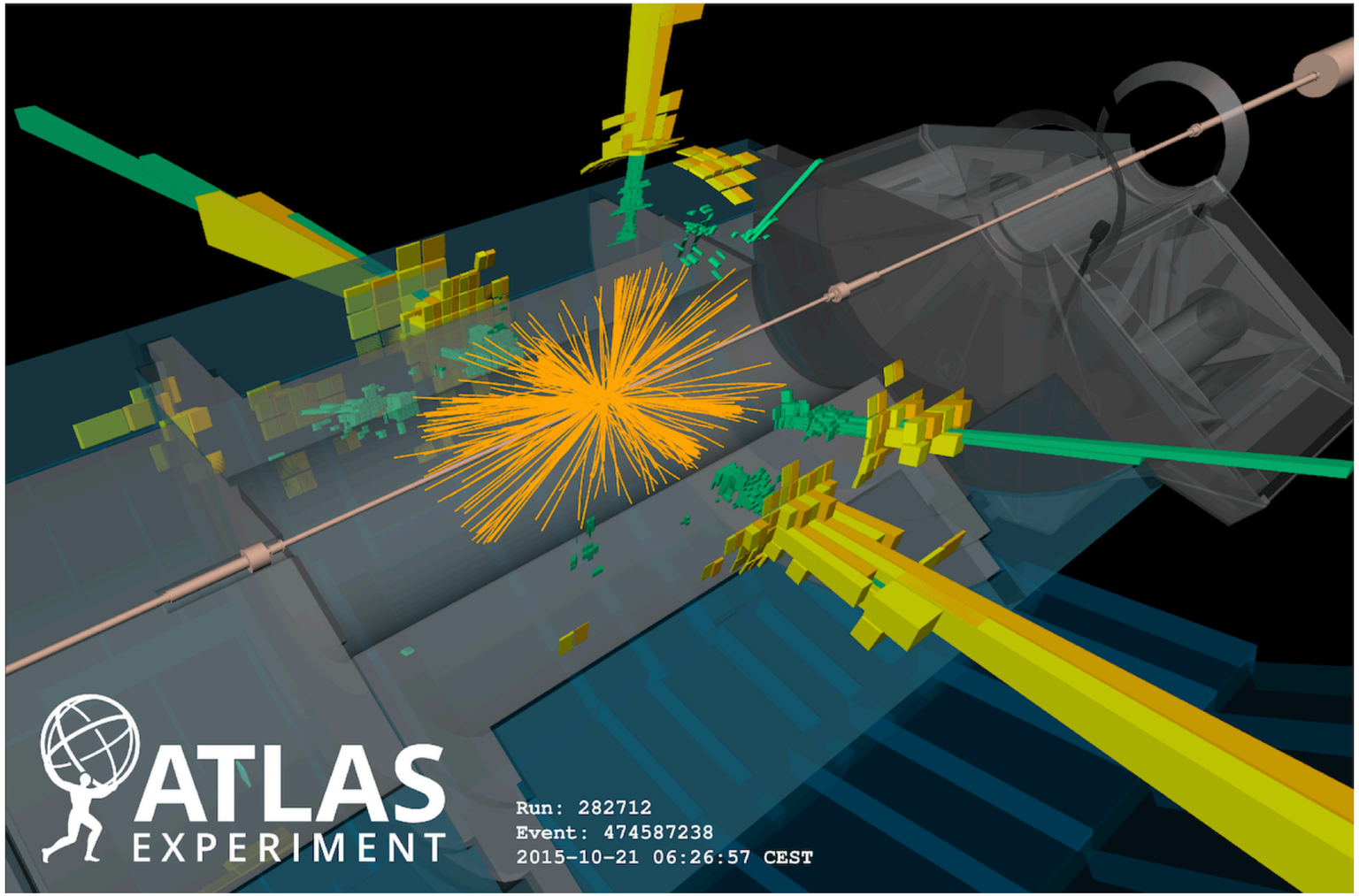
Limits from **MadMiner** indistinguishable from **true likelihood**
(usually we don't have that)

Results are based on 36 observed events, assuming SM

Example II

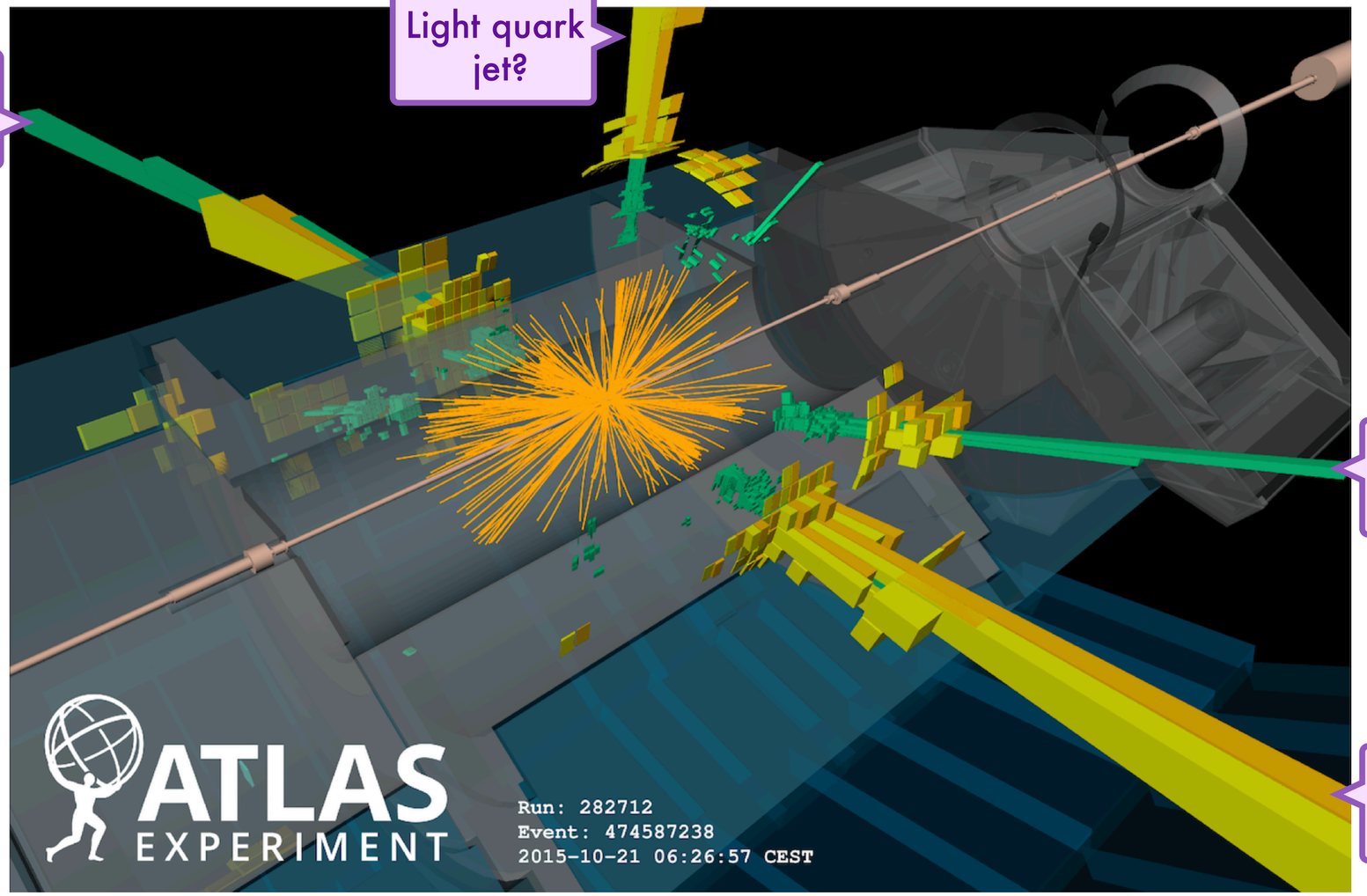
Classification with Top Tagging

What is jet tagging?



A jet is **collimated shower** of particles in the collider

What is jet tagging?



We want to know **which particle** produced the jet!

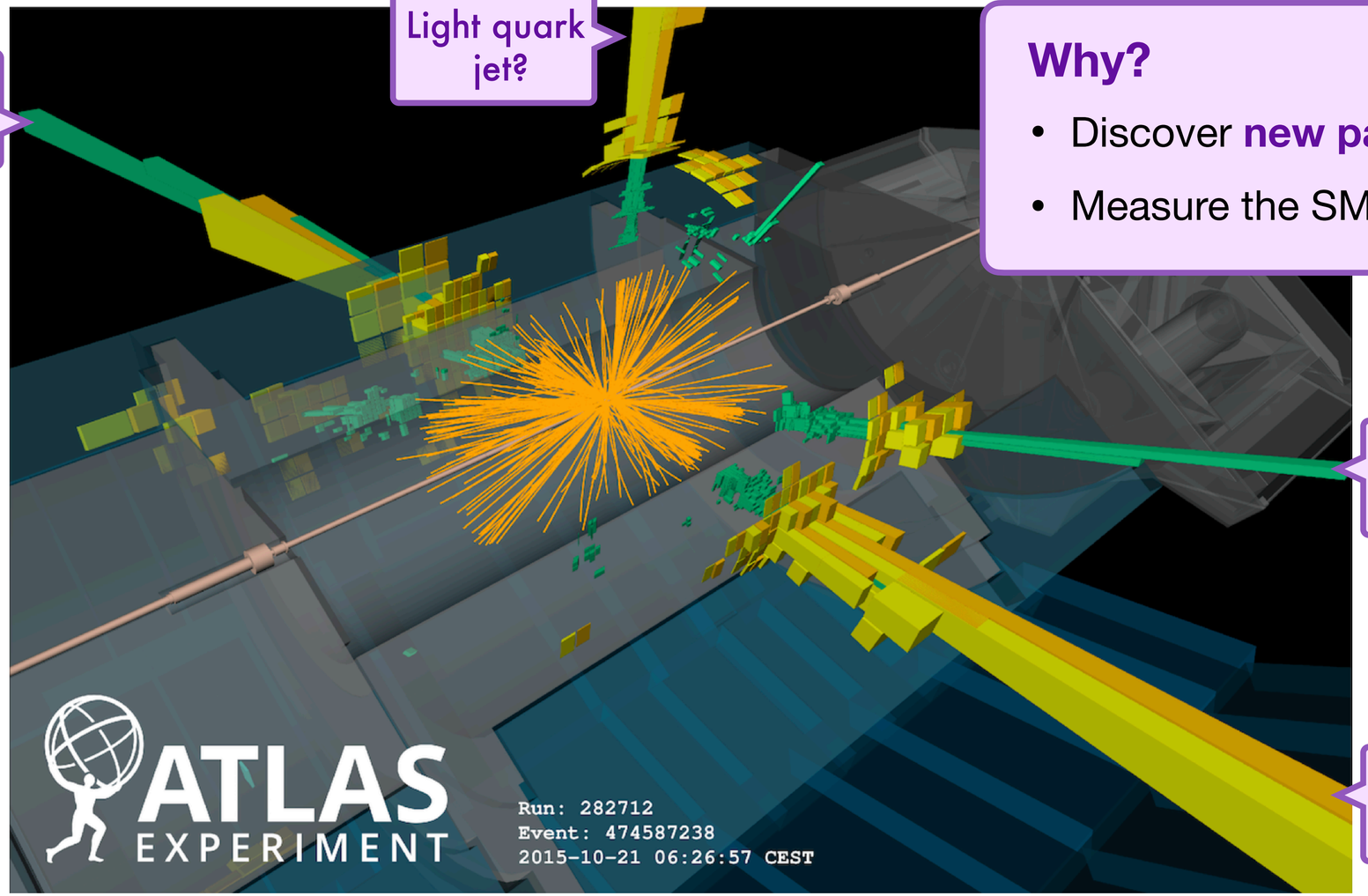
What is jet tagging?

Gluon jet?

Light quark jet?

Why?

- Discover **new particles**
- Measure the SM

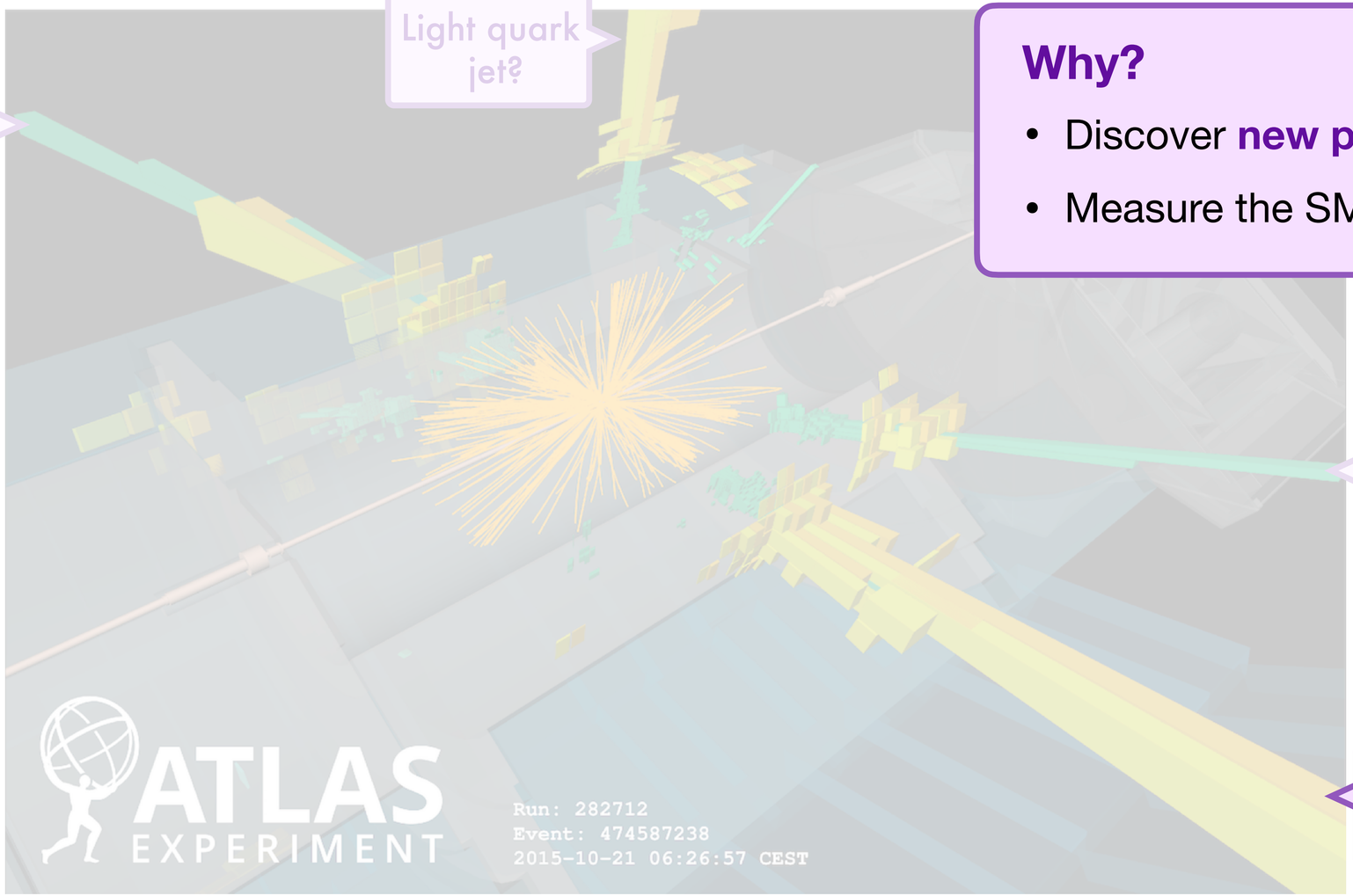


Bottom quark jet?

Top quark jet?

We want to know **which particle** produced the jet!

What is jet tagging?

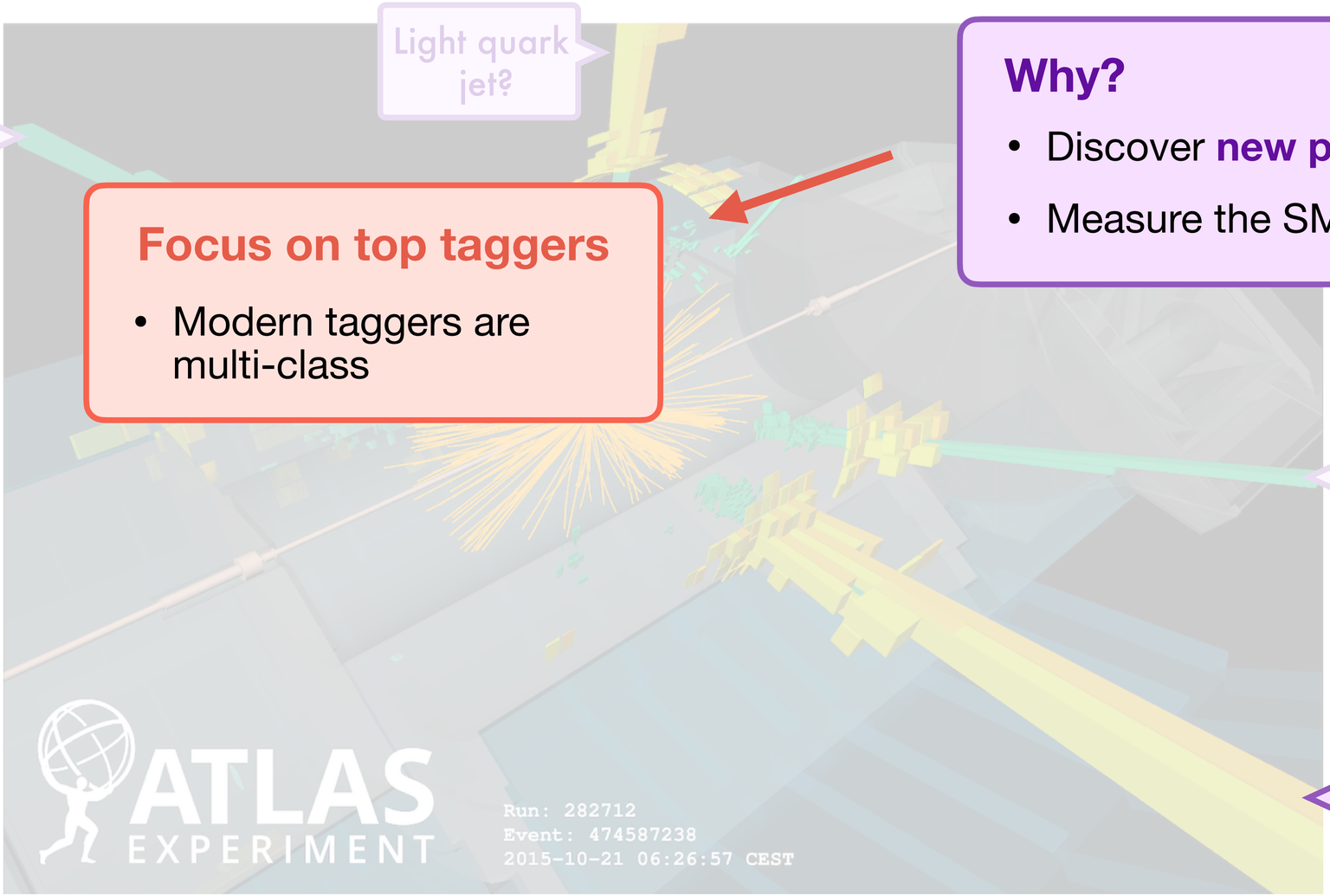


Why?

- Discover **new particles**
- Measure the SM

We want to know **which particle** produced the jet!

What is jet tagging?



Why?

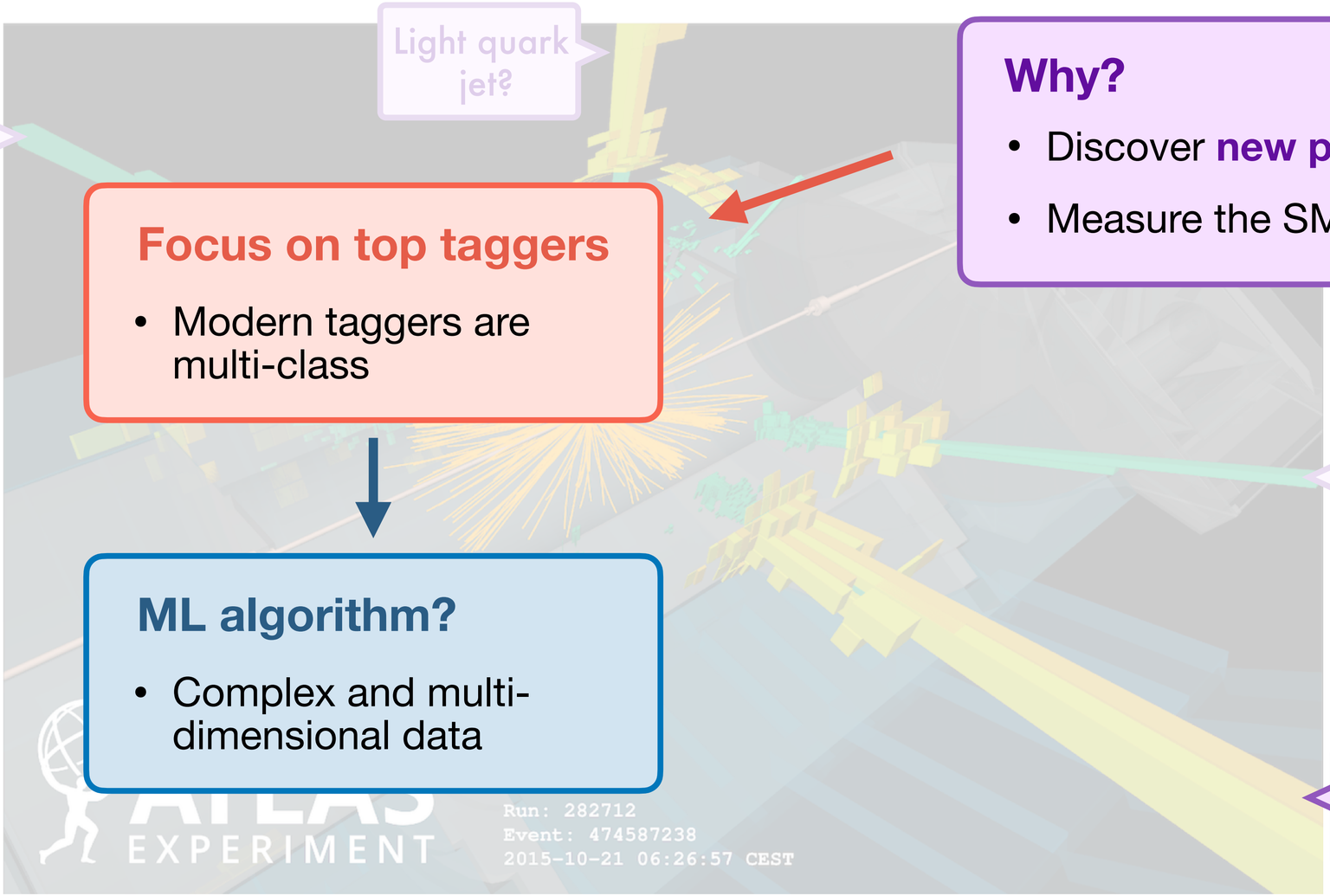
- Discover **new particles**
- Measure the SM

Focus on top taggers

- Modern taggers are multi-class

We want to know **which particle** produced the jet!

What is jet tagging?



Why?

- Discover **new particles**
- Measure the SM

Focus on top taggers

- Modern taggers are multi-class

ML algorithm?

- Complex and multi-dimensional data

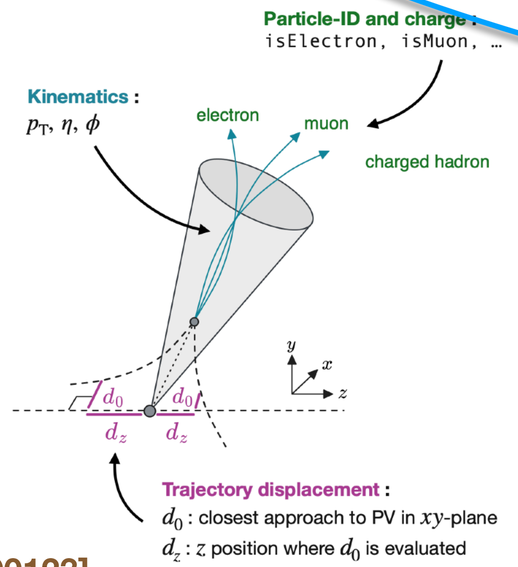
We want to know **which particle** produced the jet!

What is jet tagging?

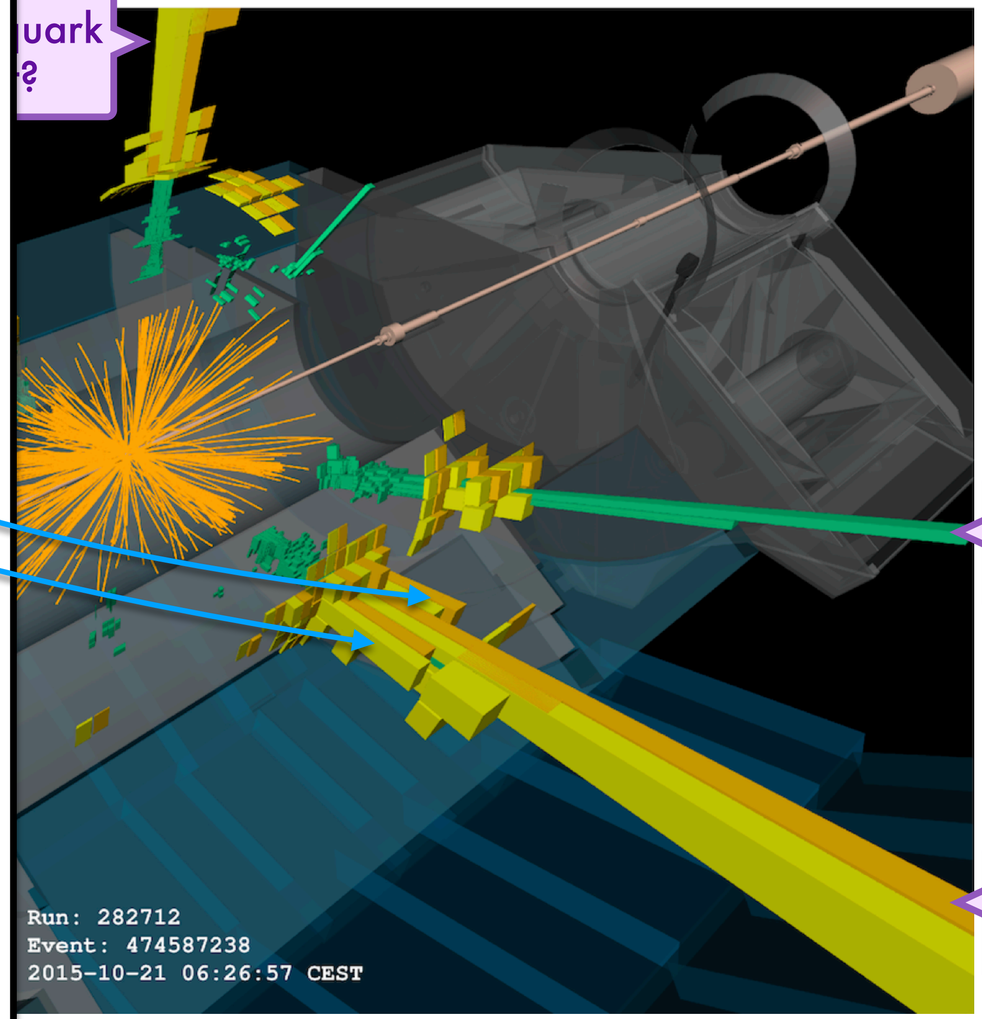
Data most naturally as **point cloud**

Each **input** (jet, event,…)
is a **set of k-dimensional vectors**
(individual particles, hits,…)

$$J_i = \{\vec{p}_1, \ldots, \vec{p}_n\}$$



[2312.00123]



uark
?

Bottom quark jet?

Top quark jet?

Landscape Dataset

- **Open dataset** for the development of better tagging algorithms for particle physics
- **2 million** simulated examples
- Perfect class labels:
  S=top jet or B=light quark/gluon jet
- Input: momentum sorted list of
  200 particles/jets with
  3 features/particle
  (p_x, p_y, p_z)

SciPost Physics Submission

The Machine Learning Landscape of Top Taggers

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

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

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

July 24, 2019

Abstract

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

[1902.09914]

Landscape Dataset

- Open d
  of bette
  for parti
- 2 millio
- Perfect
  S=top j
- Input: m
  200 par
  3 featur
  (p_x, p_y, p_z)

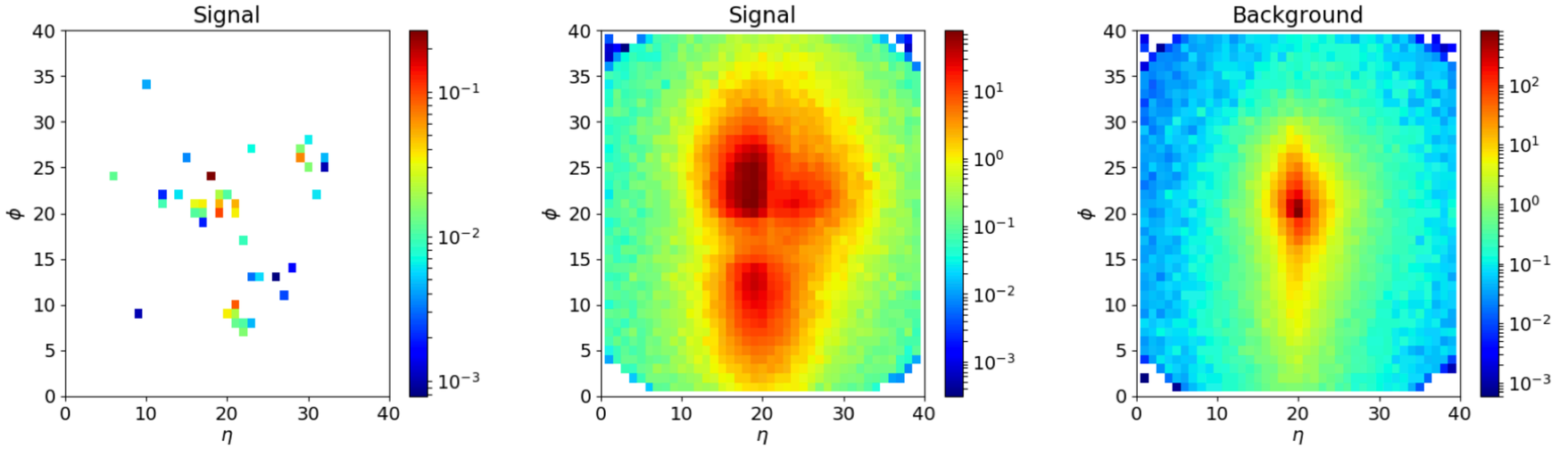


Figure 1: Left: typical single jet image in the rapidity vs azimuthal angle plane for the top signal after pre-processing. Center and right: signal and background images averaged over 10,000 individual images.

[1902.09914]

SciPost Physics Submission

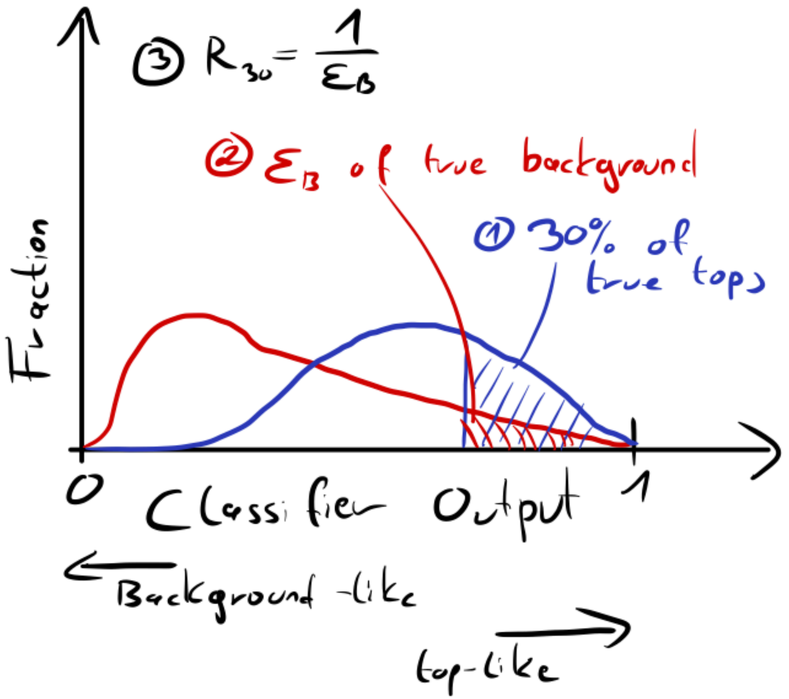
Based on the established task of identifying boosted, hadronically decaying top quarks, we compare a wide range of modern machine learning approaches. Unlike most established methods they rely on low-level input, for instance calorimeter output. While their network architectures are vastly different,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

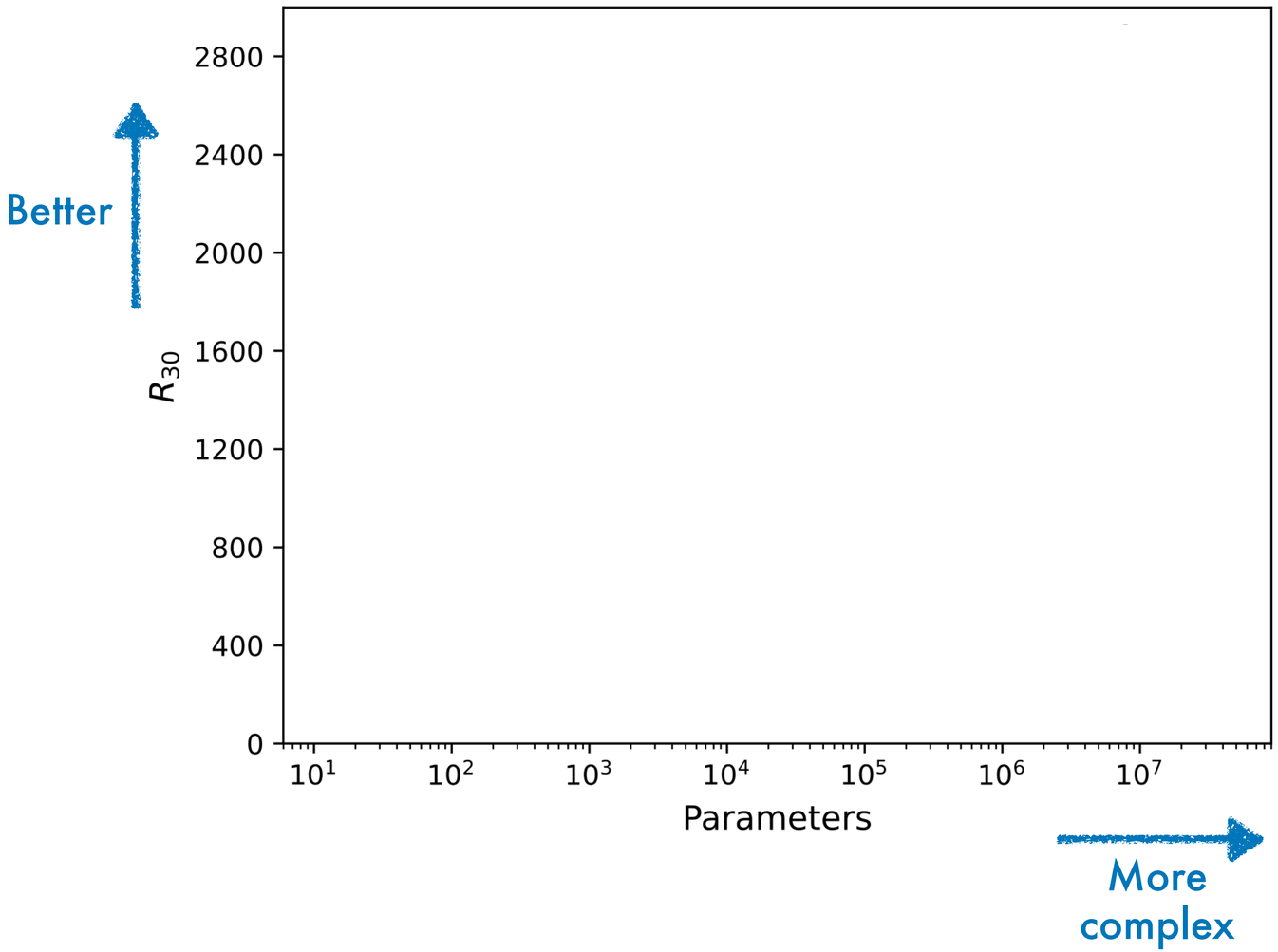
[1902.09914]

Evaluation metrics and results

Cut so that **30%** of top quarks pass selection:

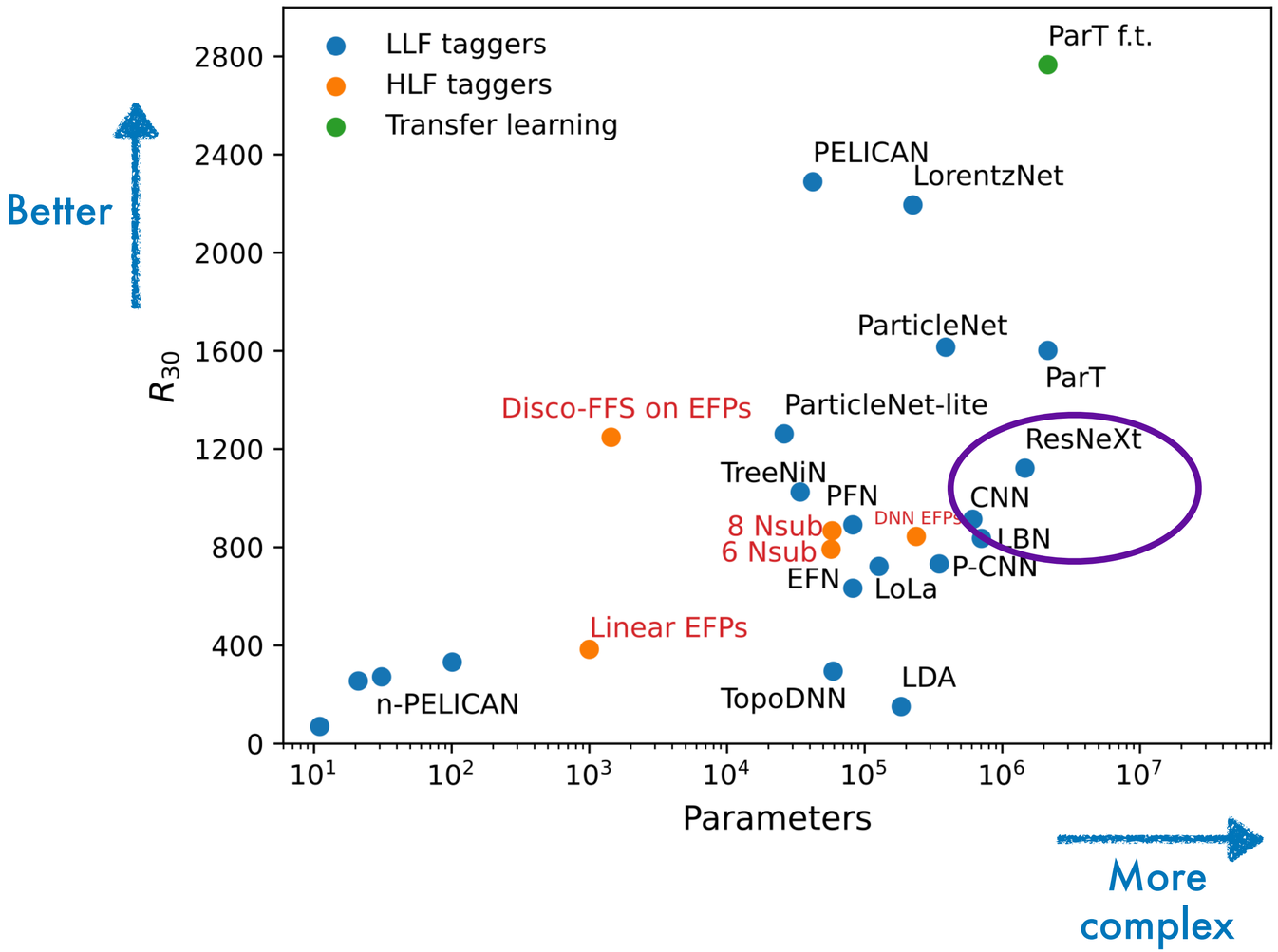
R_{30} is the inverse of the number of background jets that also pass





Evaluation metrics and results

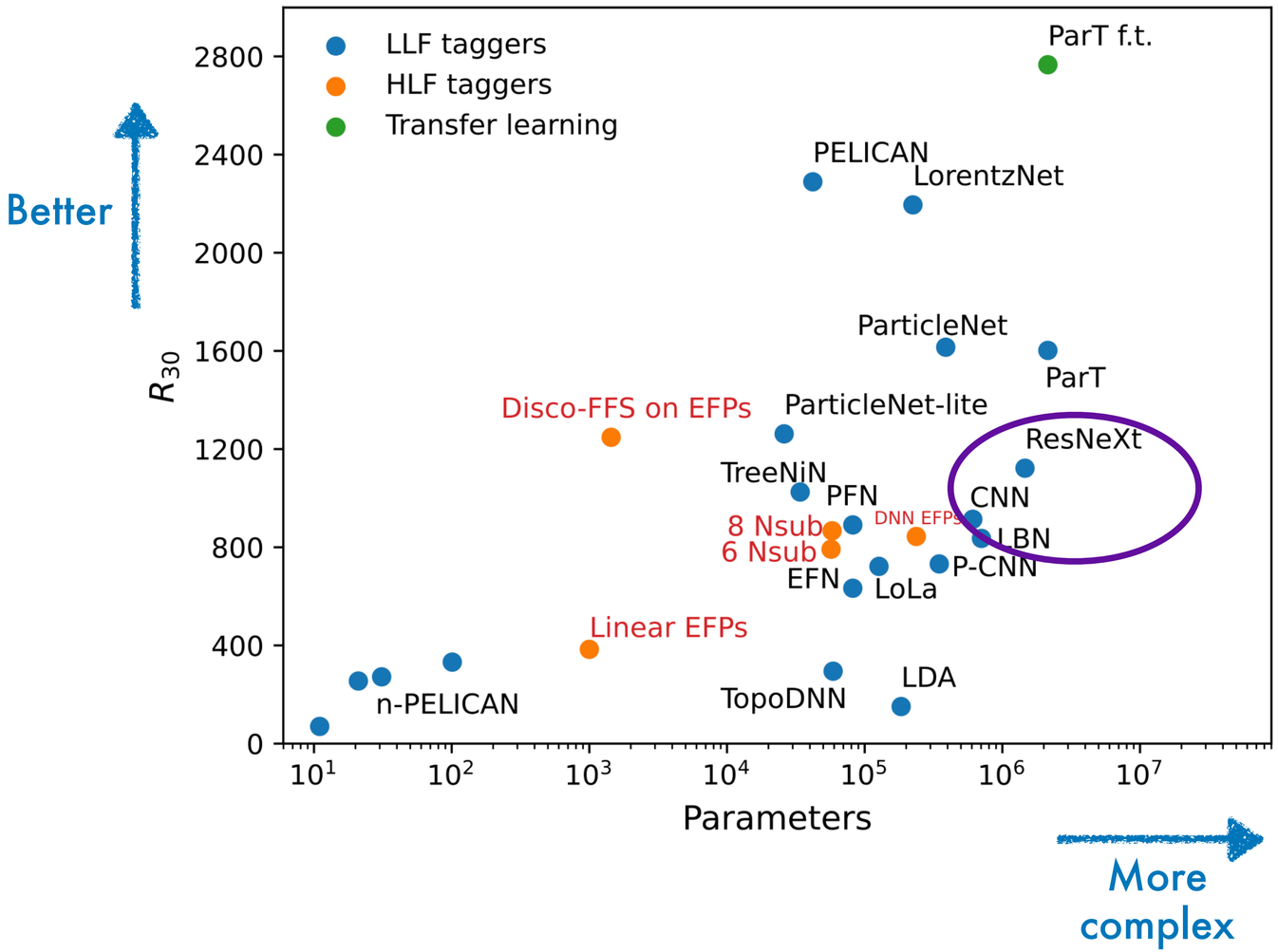
[2212.00046] + G. Kasieczka



Evaluation metrics and results

- **Jets as images**
  [1701.08784, 1803.00107]

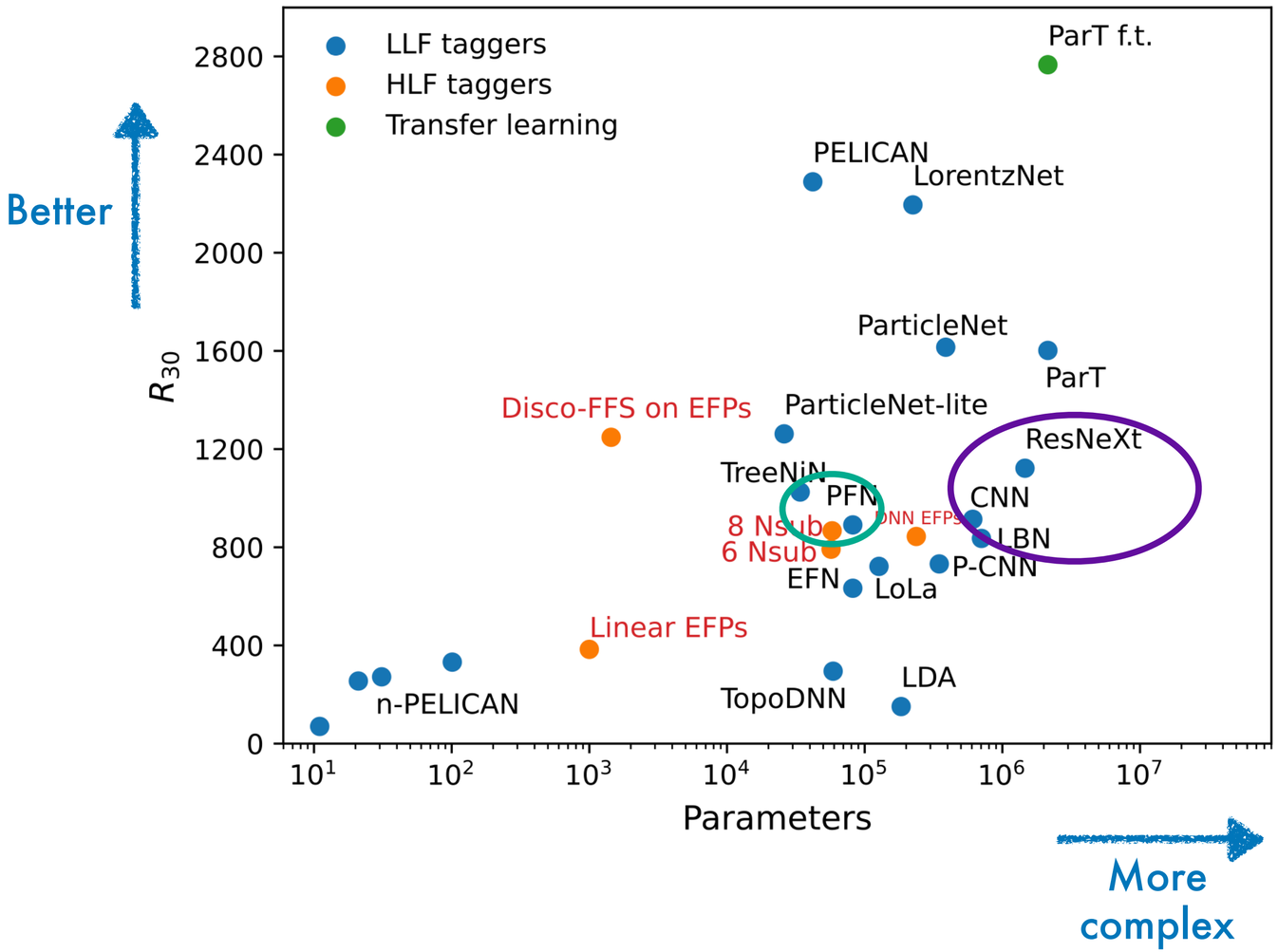
[2212.00046] + G. Kasieczka



Evaluation metrics and results

- **Jets as images**
  [1701.08784, 1803.00107]
- **Point cloud**
  [PFN 1810.05165]

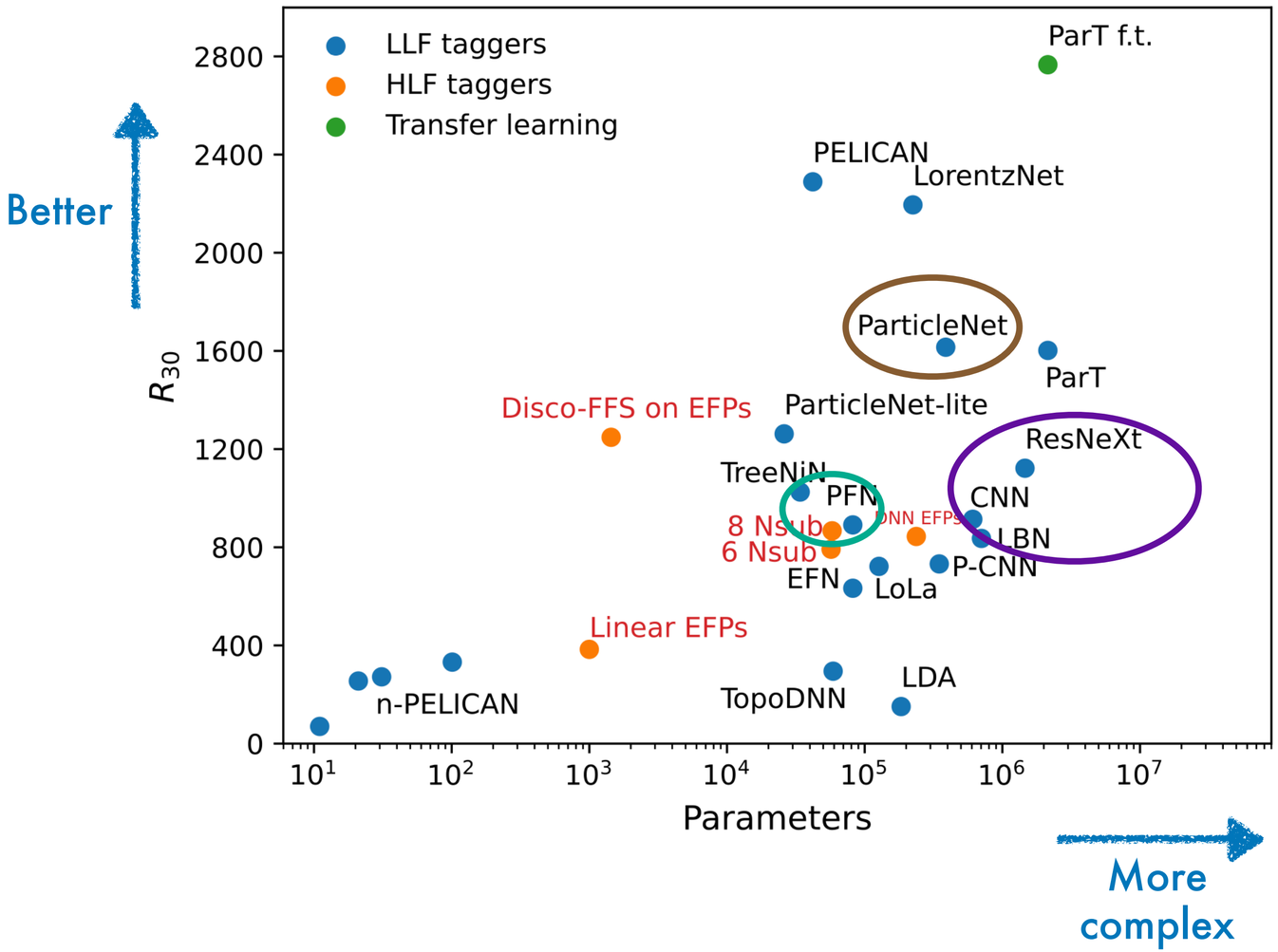
[2212.00046] + G. Kasieczka



Evaluation metrics and results

- **Jets as images**
  [1701.08784, 1803.00107]
- **Point cloud**
  [PFN 1810.05165]
- **Point cloud with Graphs**
  [1902.08570, 2007.13681]

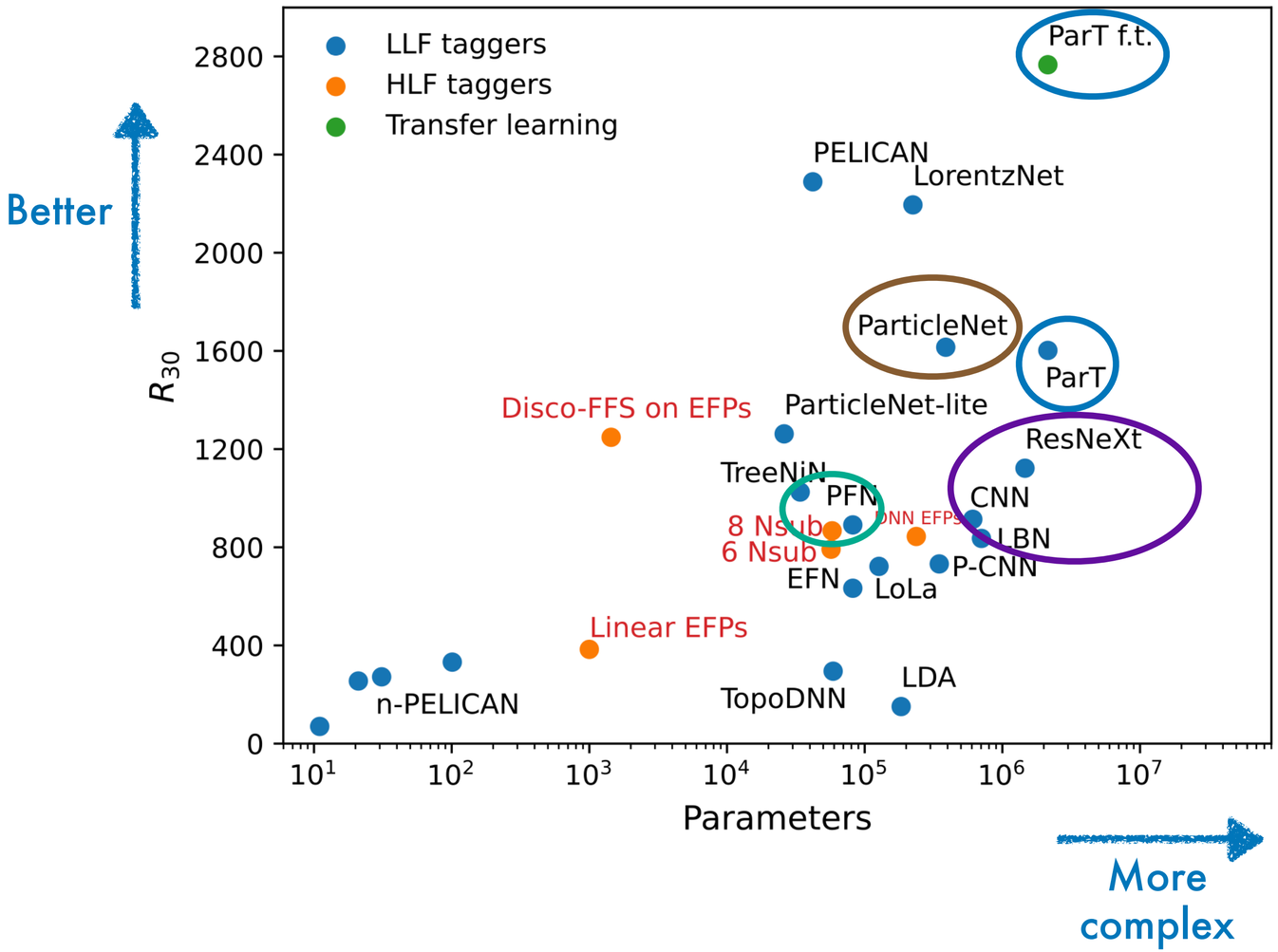
[2212.00046] + G. Kasieczka



Evaluation metrics and results

- **Jets as images**
  [1701.08784, 1803.00107]
- **Point cloud**
  [PFN 1810.05165]
- **Point cloud with Graphs**
  [1902.08570, 2007.13681]
- **Point cloud with attention**
  [2202.03772]

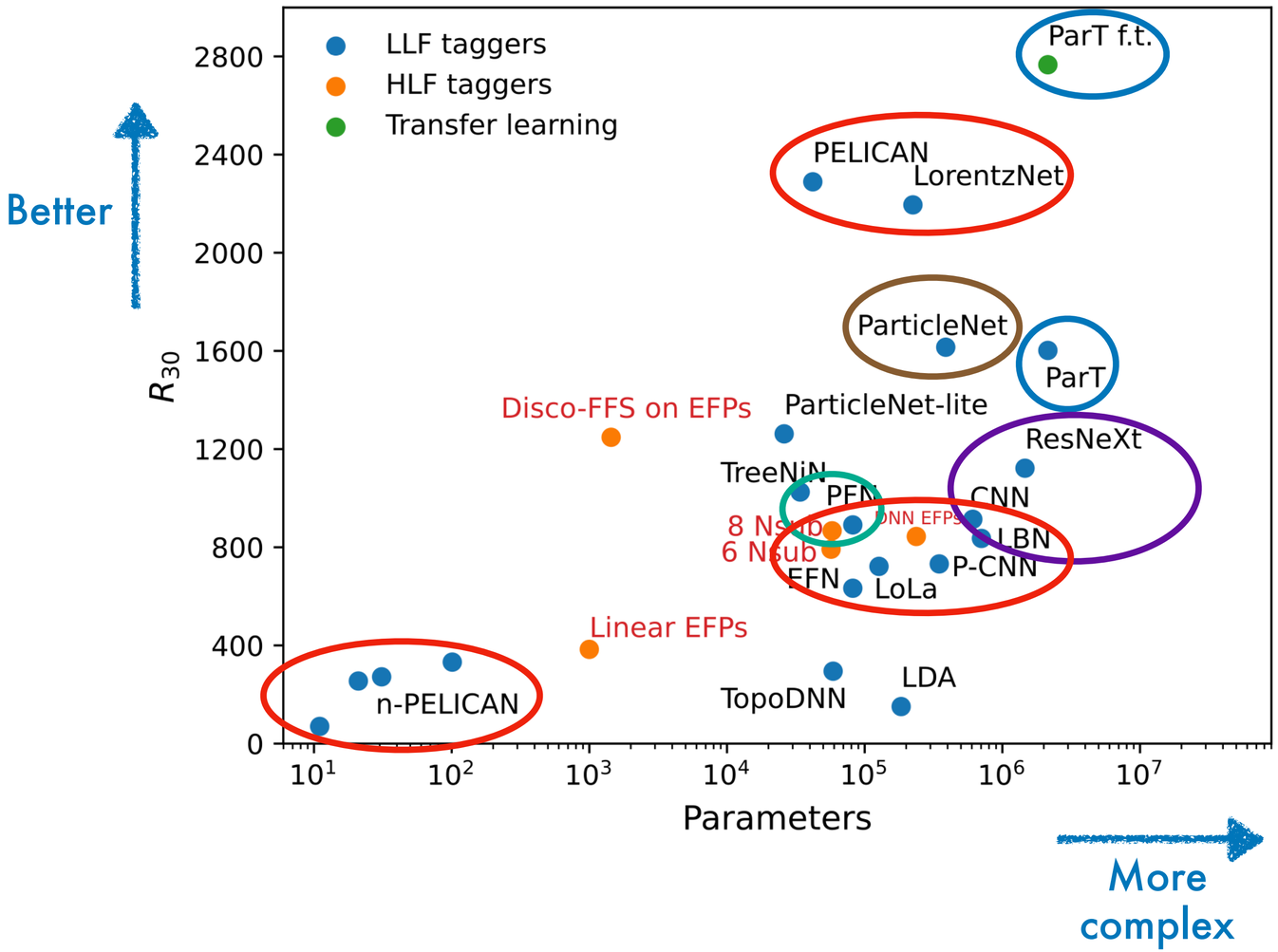
[2212.00046] + G. Kasieczka



Evaluation metrics and results

- **Jets as images**
  [1701.08784, 1803.00107]
- **Point cloud**
  [PFN 1810.05165]
- **Point cloud with Graphs**
  [1902.08570, 2007.13681]
- **Point cloud with attention**
  [2202.03772]
- **Lorentz symmetries**
  [1707.08966, 2201.08187, 2211.00454]

[2212.00046] + G. Kasieczka



Appendix

Bonus material

Variational calculus

$$L[\hat{g}(x)] = \int \mathrm{d}x \int \mathrm{d}z\, p(x,z\,|\,\theta)\big[g(x,z) - \hat{g}(x)\big]$$

$$= \int \mathrm{d}x \underbrace{\left[\hat{g}^2(x)\int \mathrm{d}z\, p(x,z\,|\,\theta) - 2\hat{g}(x)\int \mathrm{d}z\, p(x,z\,|\,\theta)\, g(x,z) + \int \mathrm{d}z\, p(x,z\,|\,\theta)\, g^2(x,z)\right]}_{=F(x)}$$

$$0 = \left.\frac{\delta F}{\delta \hat{g}}\right|_{g^*} = 2\hat{g}\underbrace{\int \mathrm{d}z\, p(x,z\,|\,\theta)}_{=p(x|\theta)} - 2\int \mathrm{d}z\, p(x,z\,|\,\theta)\, g(x,z)$$

$$g^*(x) = \frac{1}{p(x\,|\,\theta)}\int \mathrm{d}z\, p(x,z,\,|\,\theta)\, g(x,z)$$

Choose

$$g(x,z) = \frac{p(x,z,\,|\,\theta_0)}{p(x,z,\,|\,\theta_1)}$$