

Machine Learning for HEP

Lecture I – Introduction to Machine Learning



Midjourney AI



UNIVERSITÀ
DEGLI STUDI
DI MILANO

BND Graduate School – Blankenberge 2024

Ramon Winterhalder

Why care about ML in HEP?

Why ML in HEP?

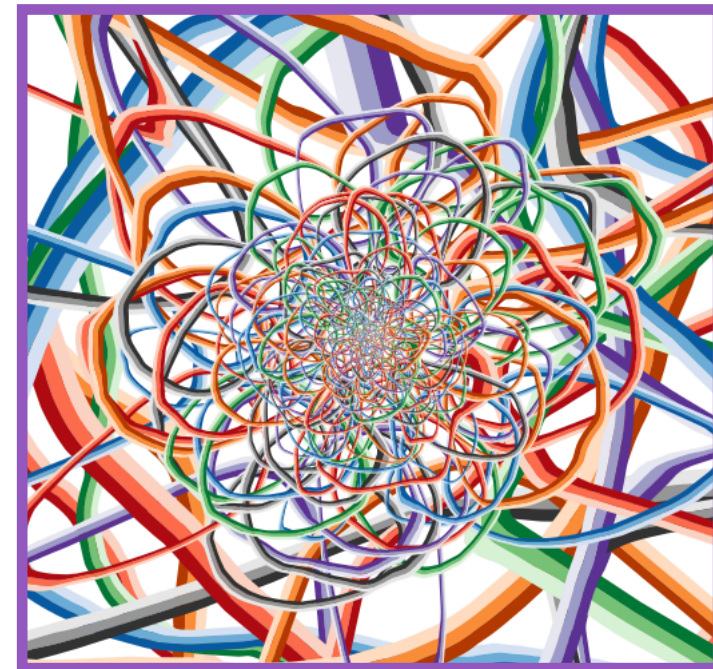
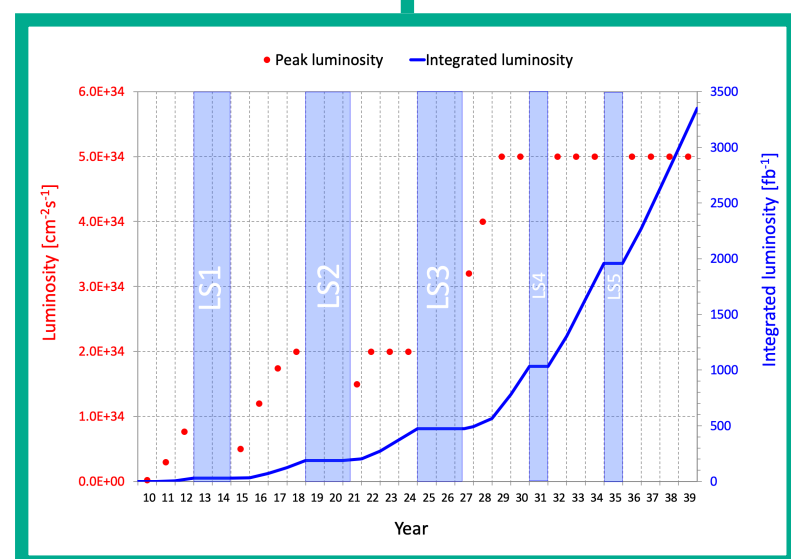
Data volume

Large amounts of data

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

ML wants lots of data

1



2

Complexity

High-dimensional & highly correlated data structure

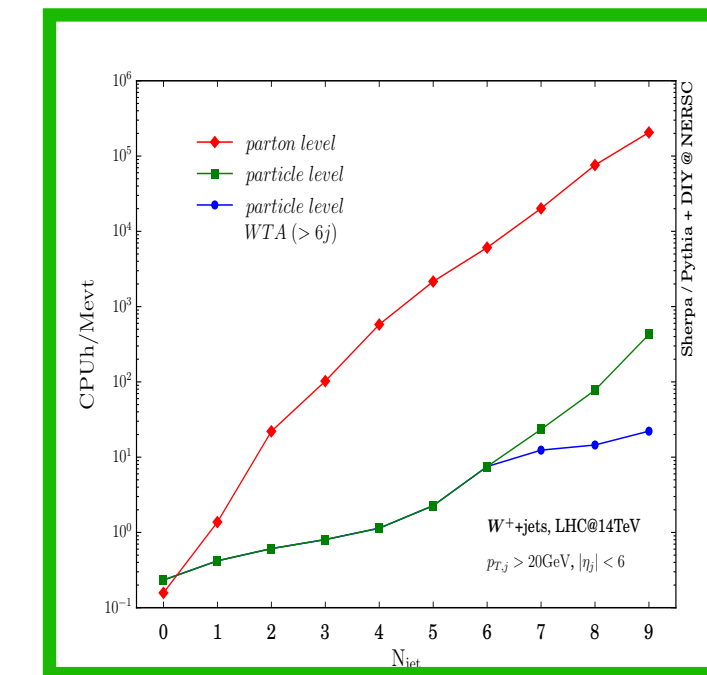
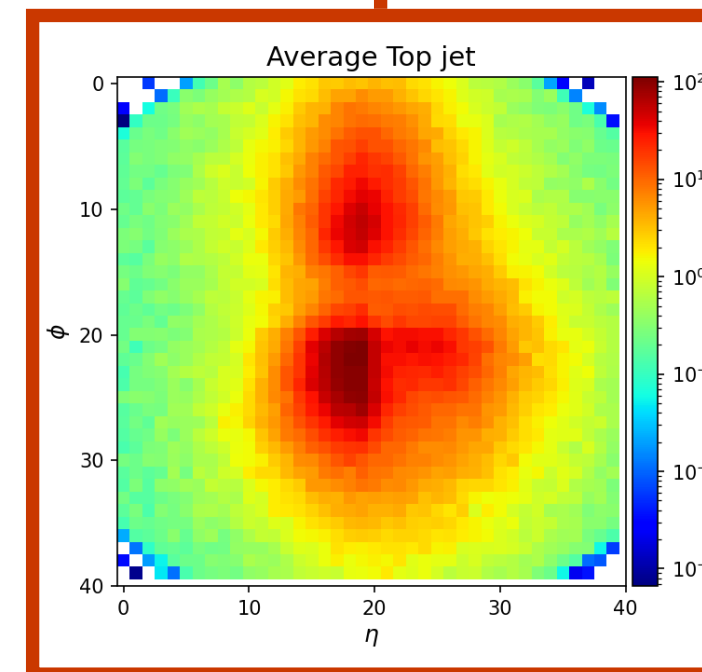
ML is expressive and flexible

Signal detection

Rare and elusive signals among large backgrounds

ML has high accuracy and sensitivity

3



4

Computing Budget

Simulation & analysis is computationally expensive

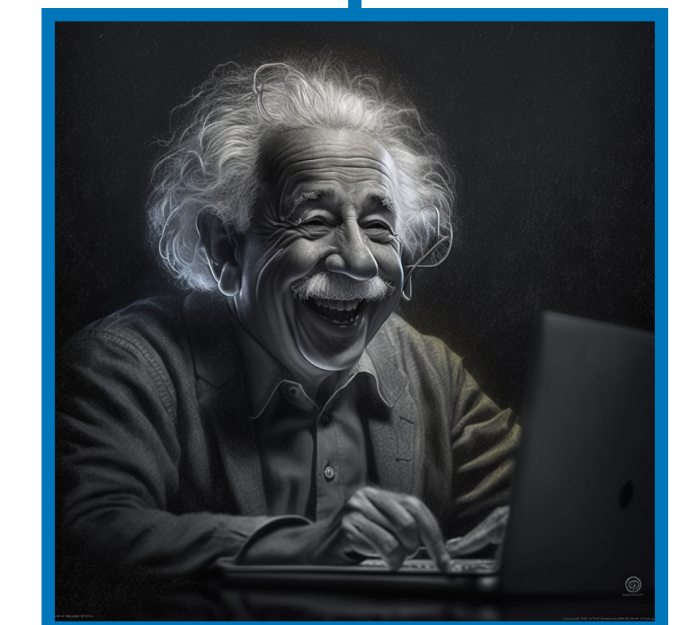
ML is fast

Increasing interest

> 150 paper/year
Future of HEP?

ML is fun!

5



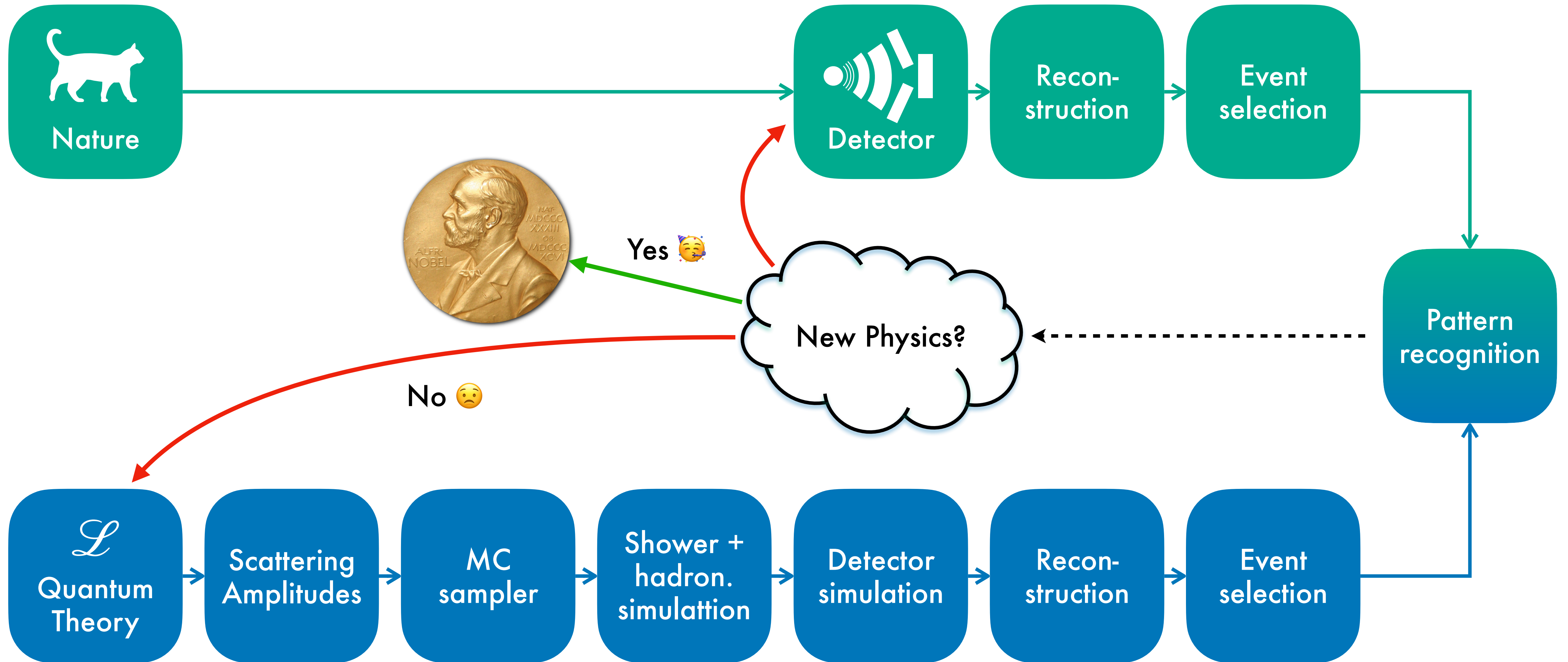
2024

1987

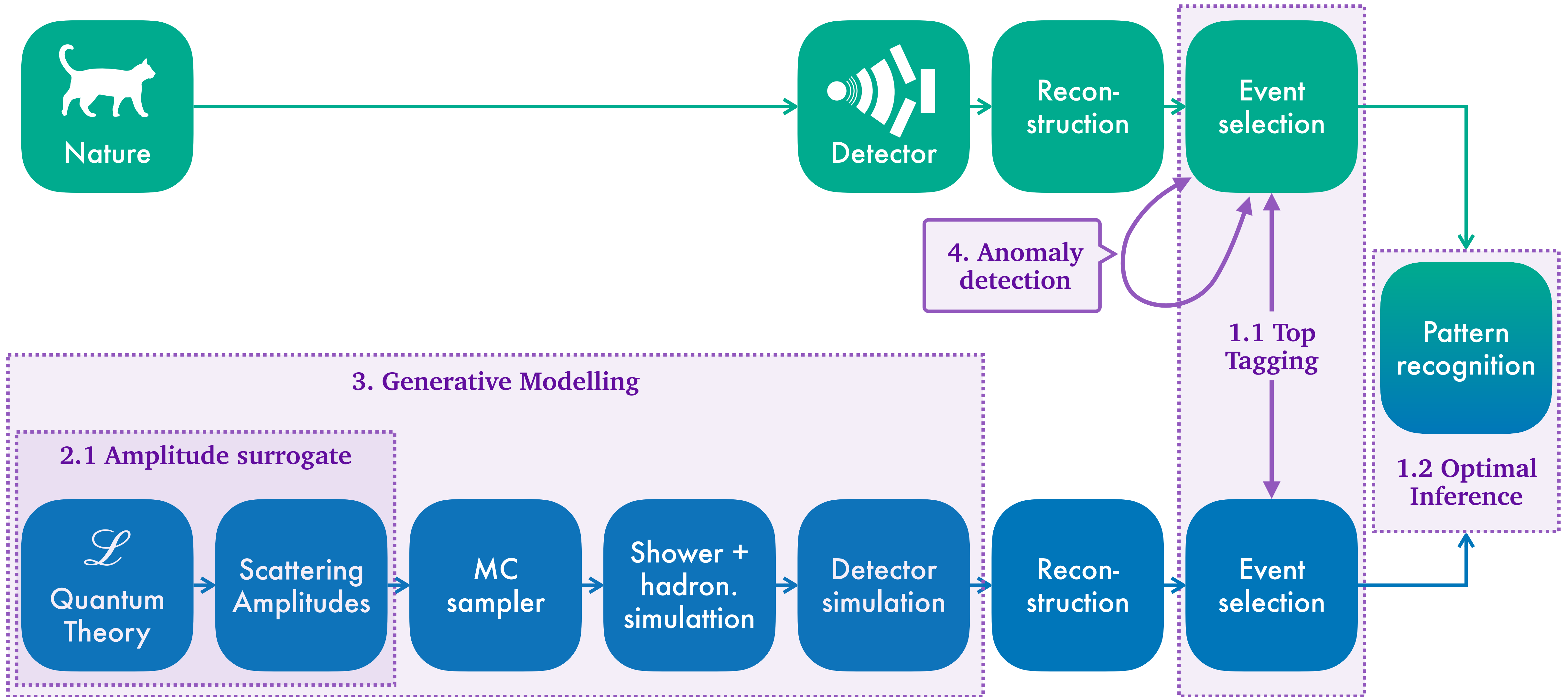
2014



LHC analysis (oversimplified)



LHC analysis + ML



Plan of attack



Wednesday

1. Introduction to Machine Learning
 - Basic concepts of machine learning
 2. Bayesian Neural Networks
 - Uncertainty estimation
-

Thursday

3. Generative Models for the LHC
 - GANs, normalizing flows & diffusion models
4. Anomaly Detection
 - Autoencoders, CWoLa,...

Lecture I

Introduction to Machine Learning

What is machine learning?

What is machine learning?



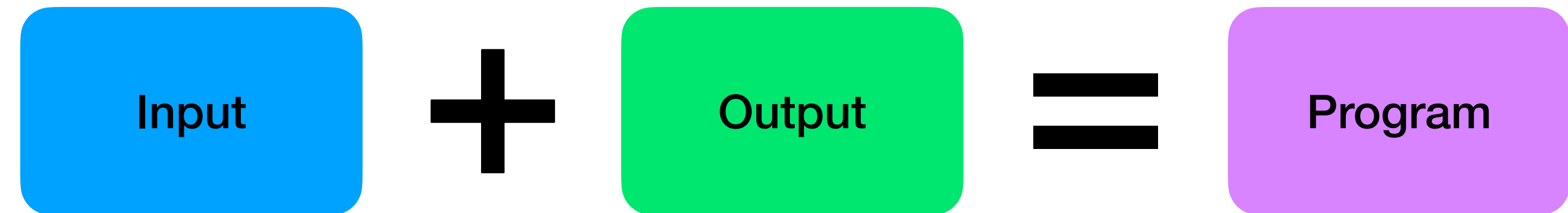
Tom Mitchell
ML pioneer

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

Traditional approach



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

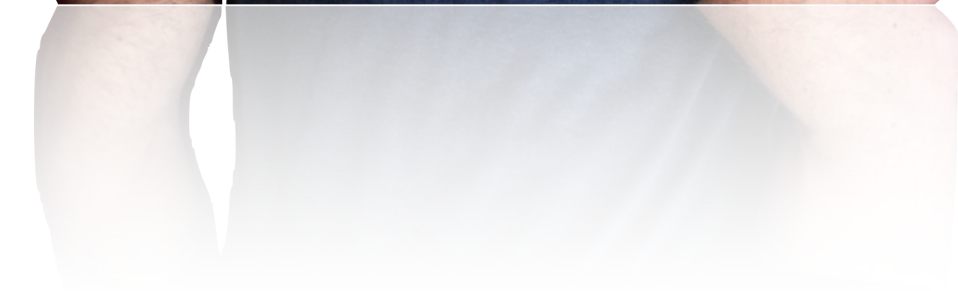
What is machine learning?



“Machine learning is just statistics on steroids.
Lots and lots of steroids”

Ilya Sutskever

Co-founder of OpenAI

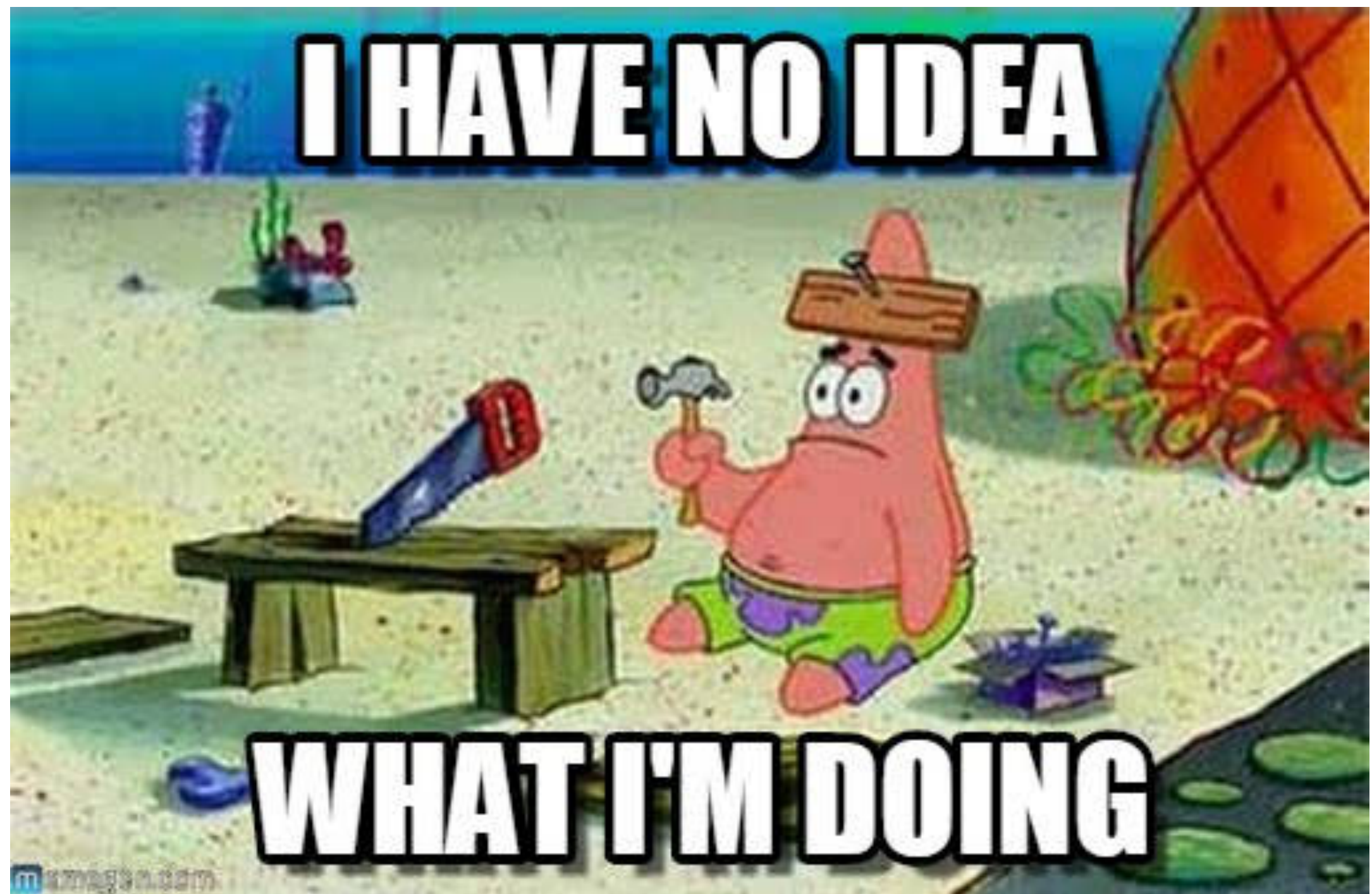


What is machine learning?

What it feels like... (sometimes)

Aim of the lectures:

Giving you the tools and ideas to use it right!



What is machine learning?

13

What it feels like... (sometimes)

Aim of the lectures:

Giving you the tools and ideas to use it right!

Be aware!

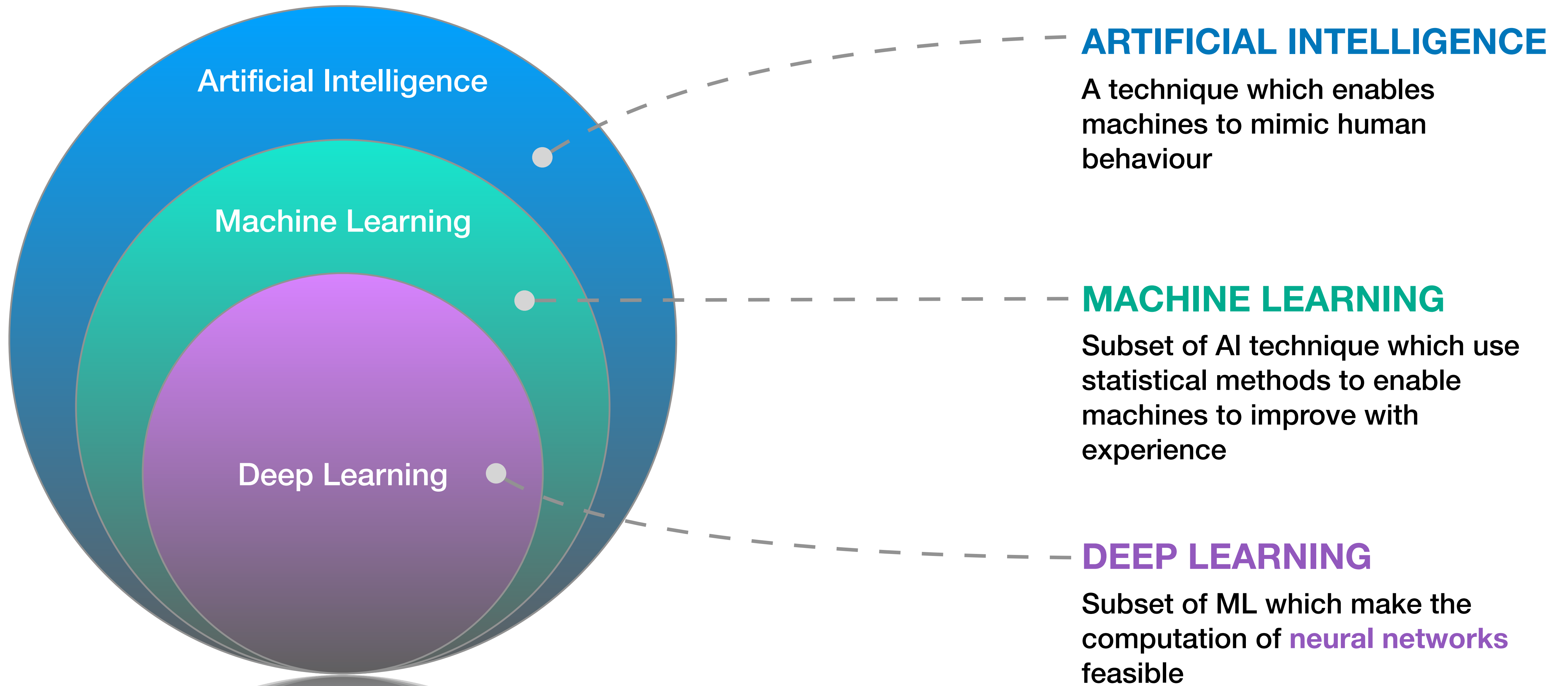
The core of machine learning is to find structure in data - **no more no less!**



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

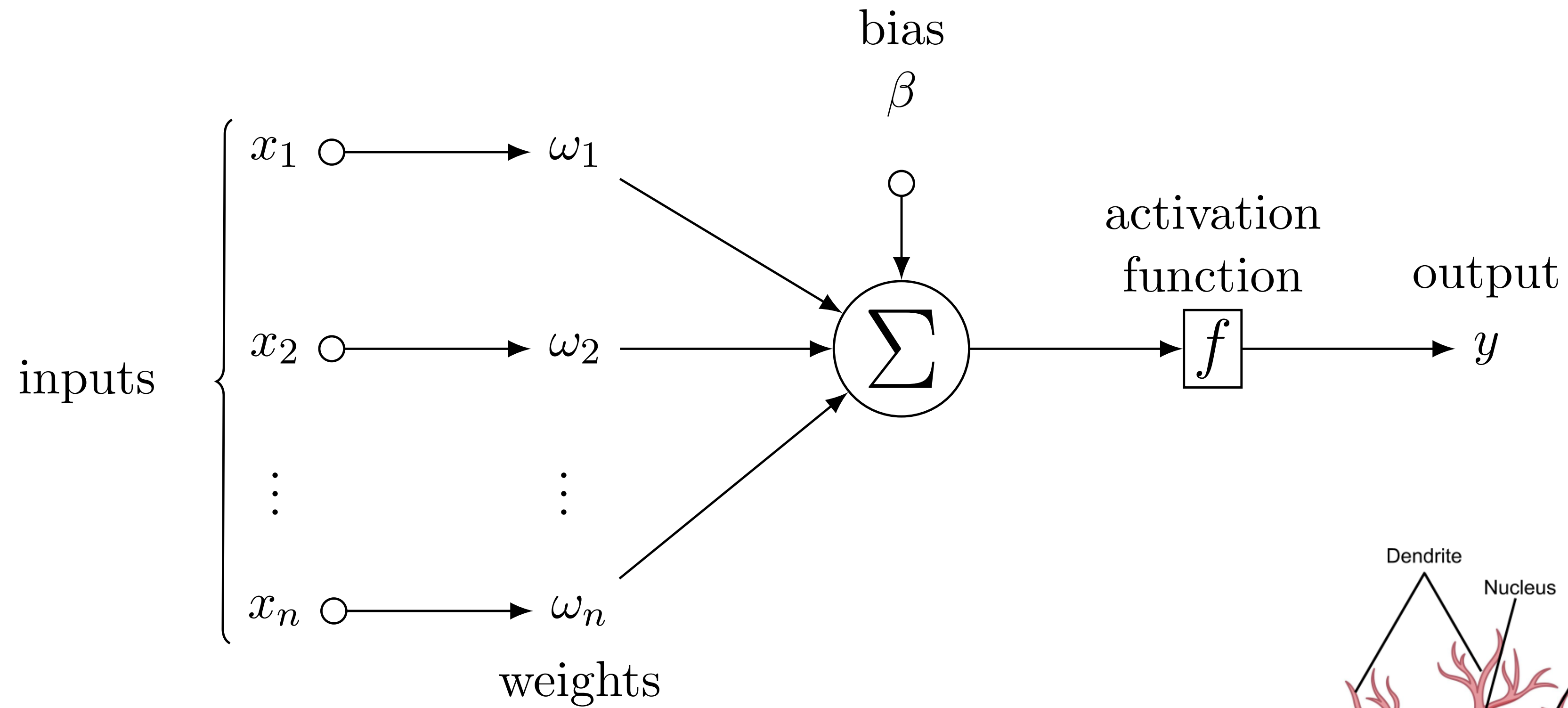
What is deep learning?

What is deep learning?

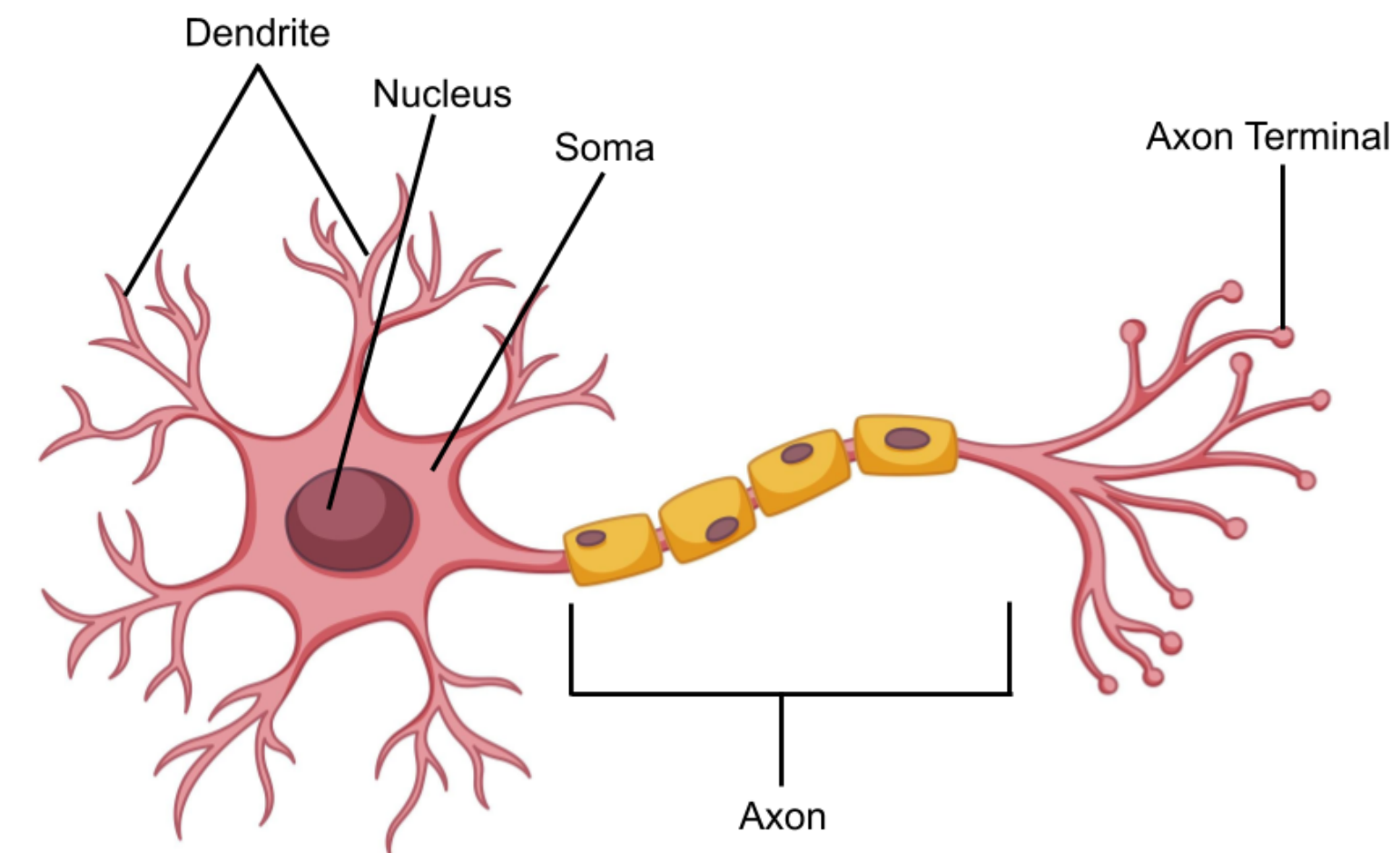


What are neural networks?

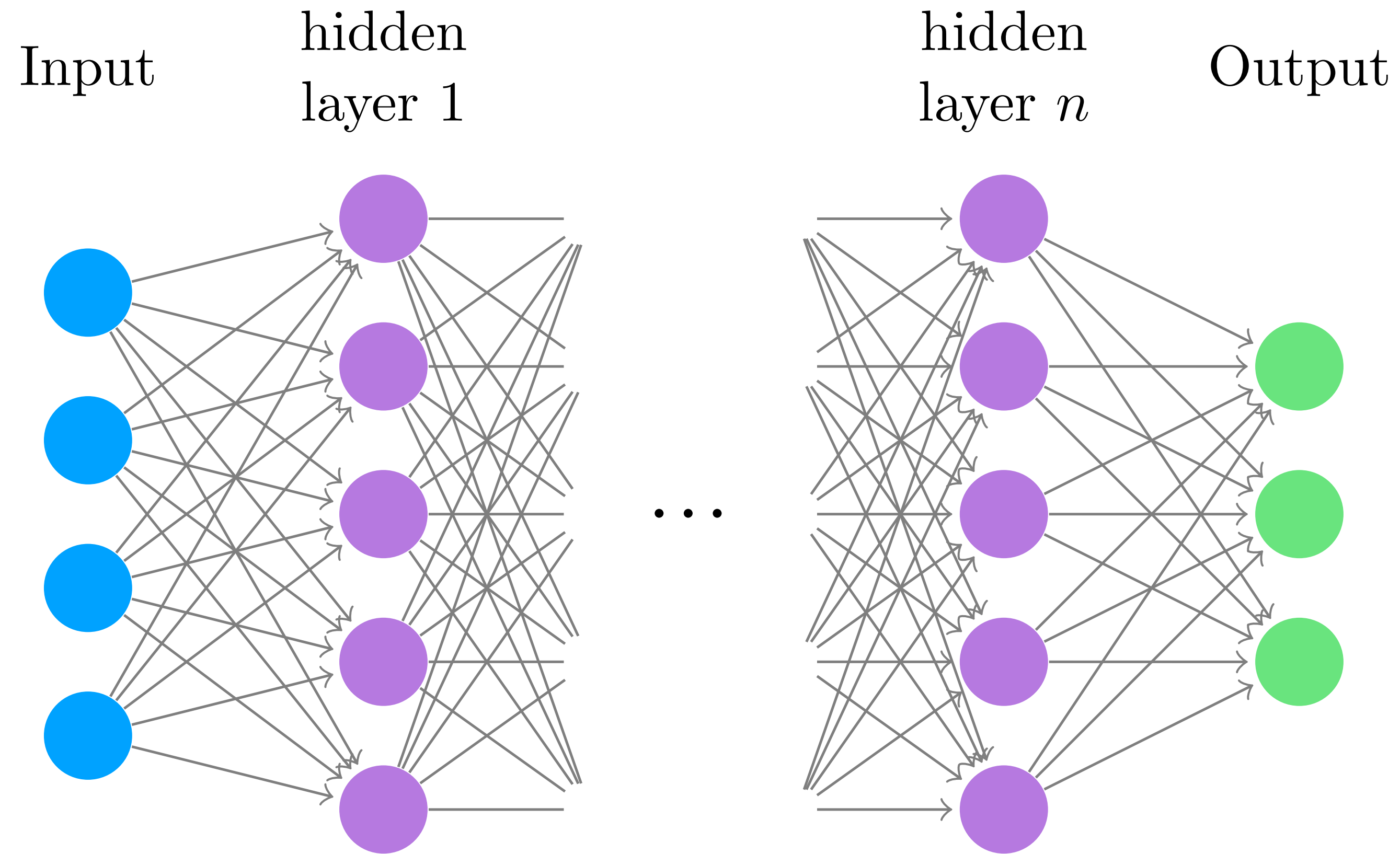
Components of artificial neurons



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



Deep neural network



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

Training objective and optimization



Performance measure

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

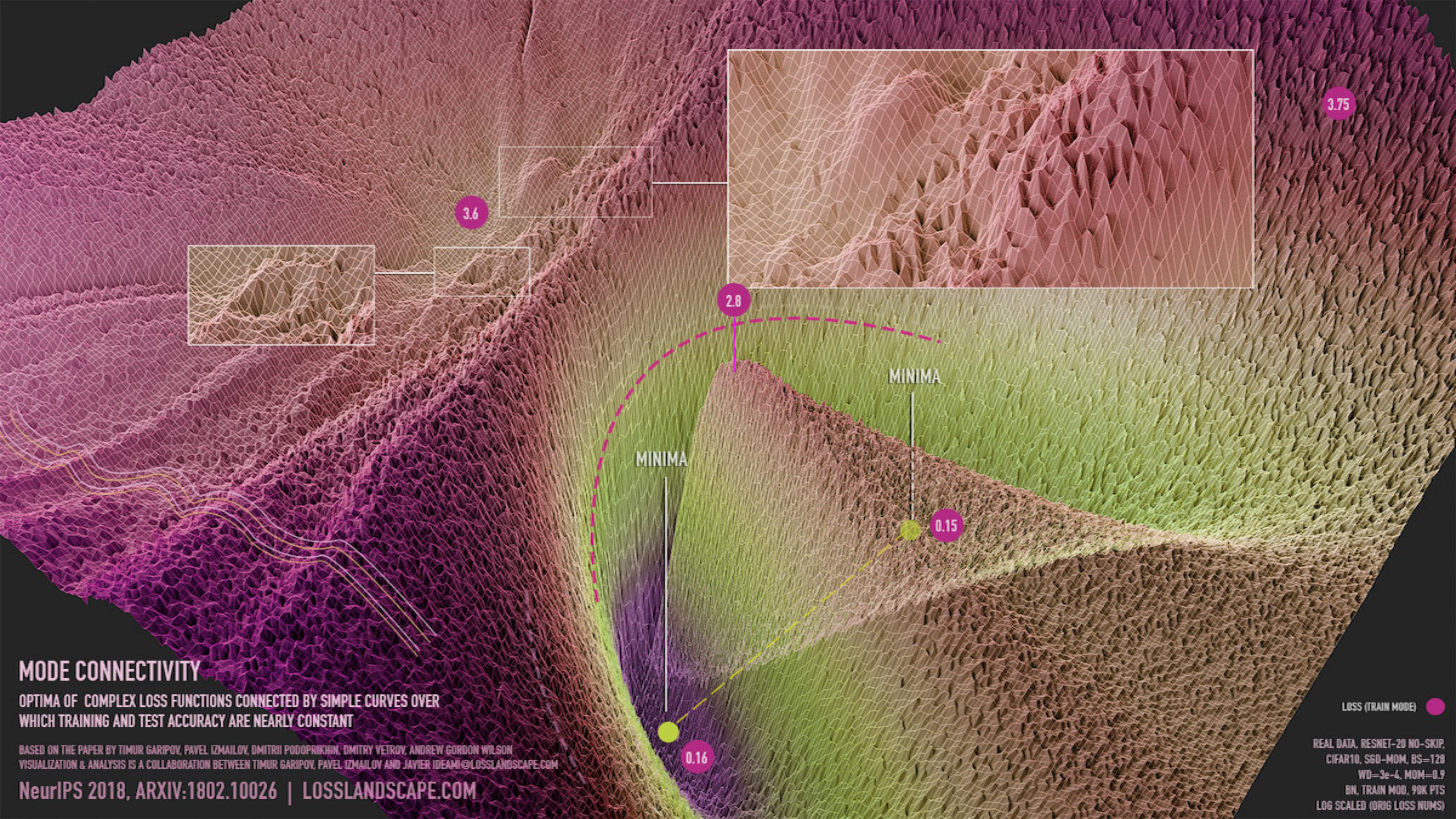
$$\mathcal{L}_{\text{tot}} = \sum_i^N \mathcal{L}(f_\omega(x_i), y_i)$$

Backpropagation

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

Stochastic gradient descent

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



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

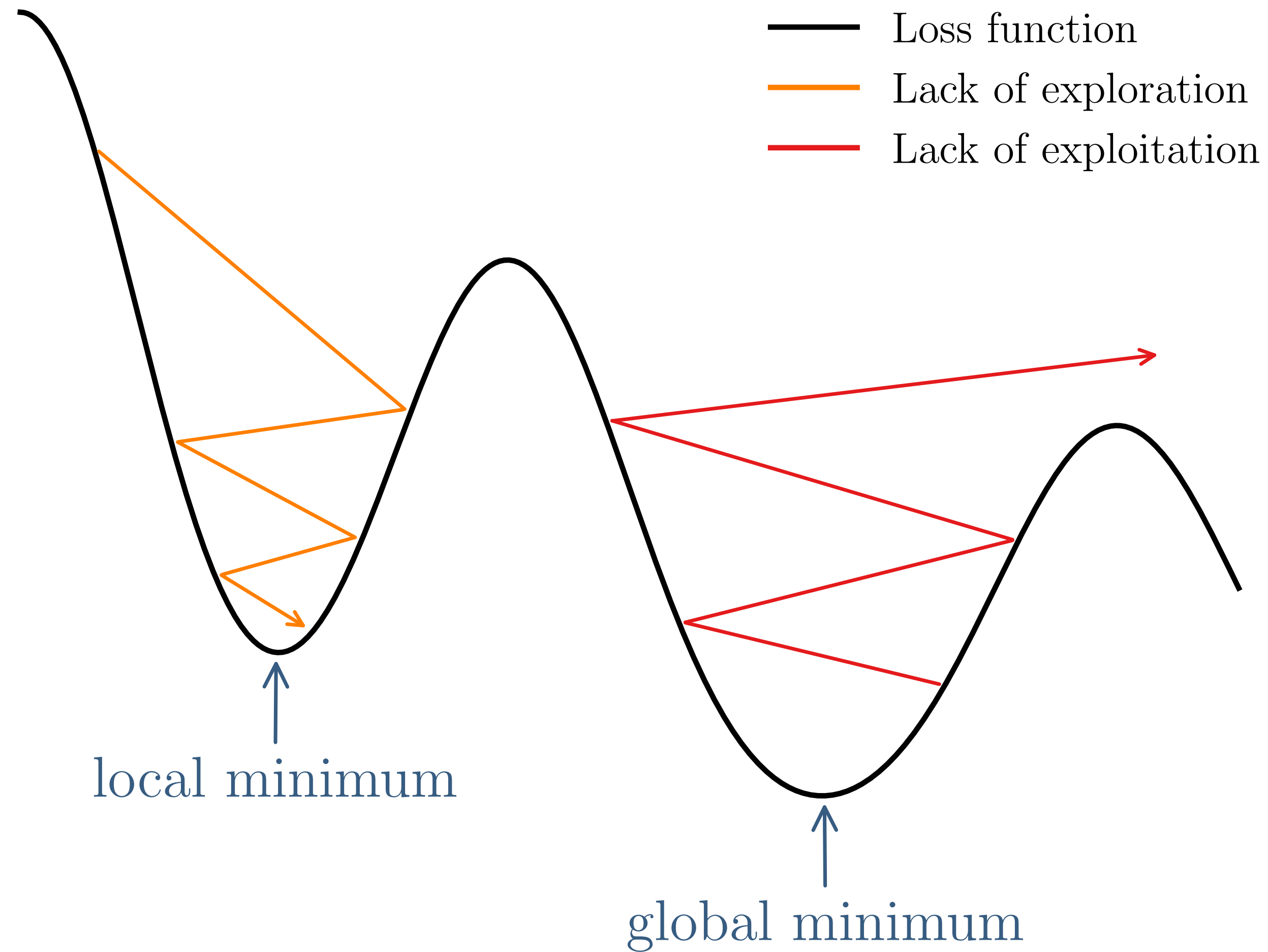
NeurIPS 2018, ARXIV:1802.10026 | LOSSLANDSCAPE.COM

LOSS (TRAIN MODE) ●

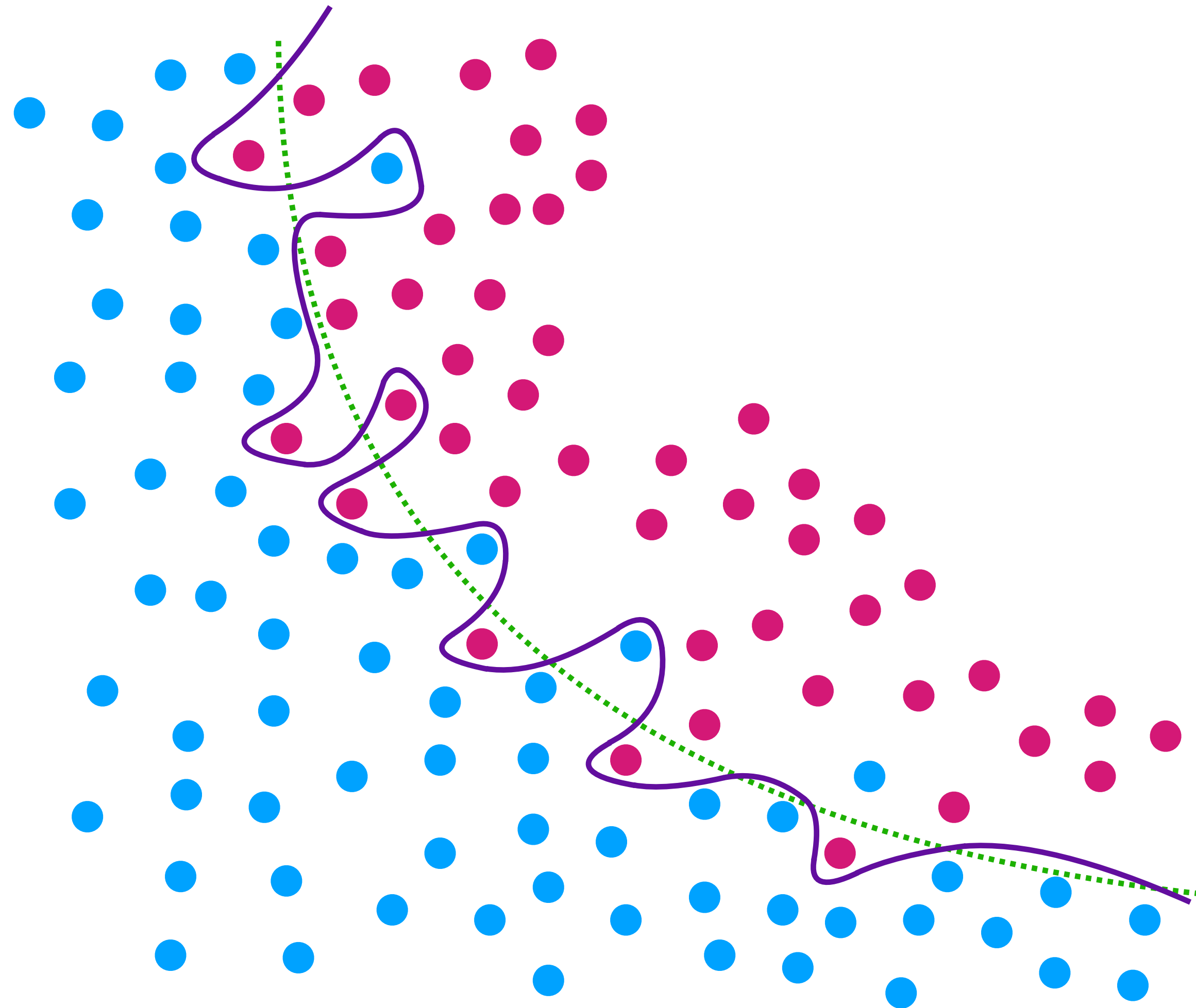
REAL DATA, RESNET-20 NO-SKIP,
CIFAR10, SGD-MOM, BS=128
WD=3e-4, MOM=0.9
BN, TRAIN MOD, 90K PTS
LOG SCALED (ORIG LOSS NUMS)

Optimizer and loss landscape

Finding the global minimum in a sea of local minima



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

$$p(x|\omega) = \prod_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{|f_j - f_{\omega}(x_j)|^2}{2\sigma_j}\right)$$
$$\Rightarrow \log p(x|\omega) = -\sum_j \frac{|f_j - f_{\omega}(x_j)|^2}{2\sigma_j} + \text{const}$$

Assumes Gaussian probability distribution

Loss function of the fit

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

Minimize negative log-likelihood

Regression in Machine Learning

Loss function in fits

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

Mini-batches require mean instead of sum
if error σ_j unknown or same for all

Typical ML Regression loss

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

Puts more weight solely on deviations of the function values!

Preprocessing of the training data

$$f_j \rightarrow \log f_j \quad f_j \rightarrow f_j - \langle f_j \rangle \quad f_j \rightarrow \frac{f_j}{\langle f_j \rangle} \dots$$

Mean-absolute error

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

What about Classification?

Approximate data probability

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

Optimal test
statistic

Kullback-Leibler divergence

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

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

$$\begin{aligned}\mathcal{L}_{\text{class}} &= \sum_{j=S,B} D_{\text{KL}}(p_{\text{data},j} | p_{\omega,j}) \\ &= - \sum_{\{x\}} [p_{\text{data},S} \log p_{\omega,S} + p_{\text{data},B} \log p_{\omega,B}] + \text{const}\end{aligned}$$



Using $p_B = 1 - p_S$

Binary cross-entropy loss

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



Generalization (multi-class)

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

Useful libraries + algorithms

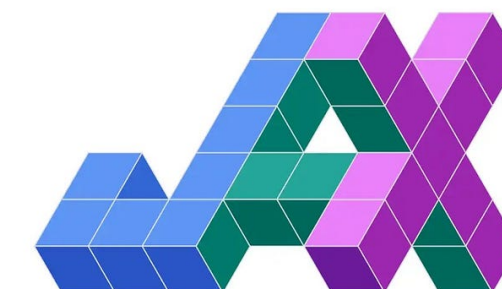
- **scikit-learn:** For most of the “basic” algorithms like Linear Regression and Boosted Decision Trees. Also useful for preprocessing, model combination, etc.



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



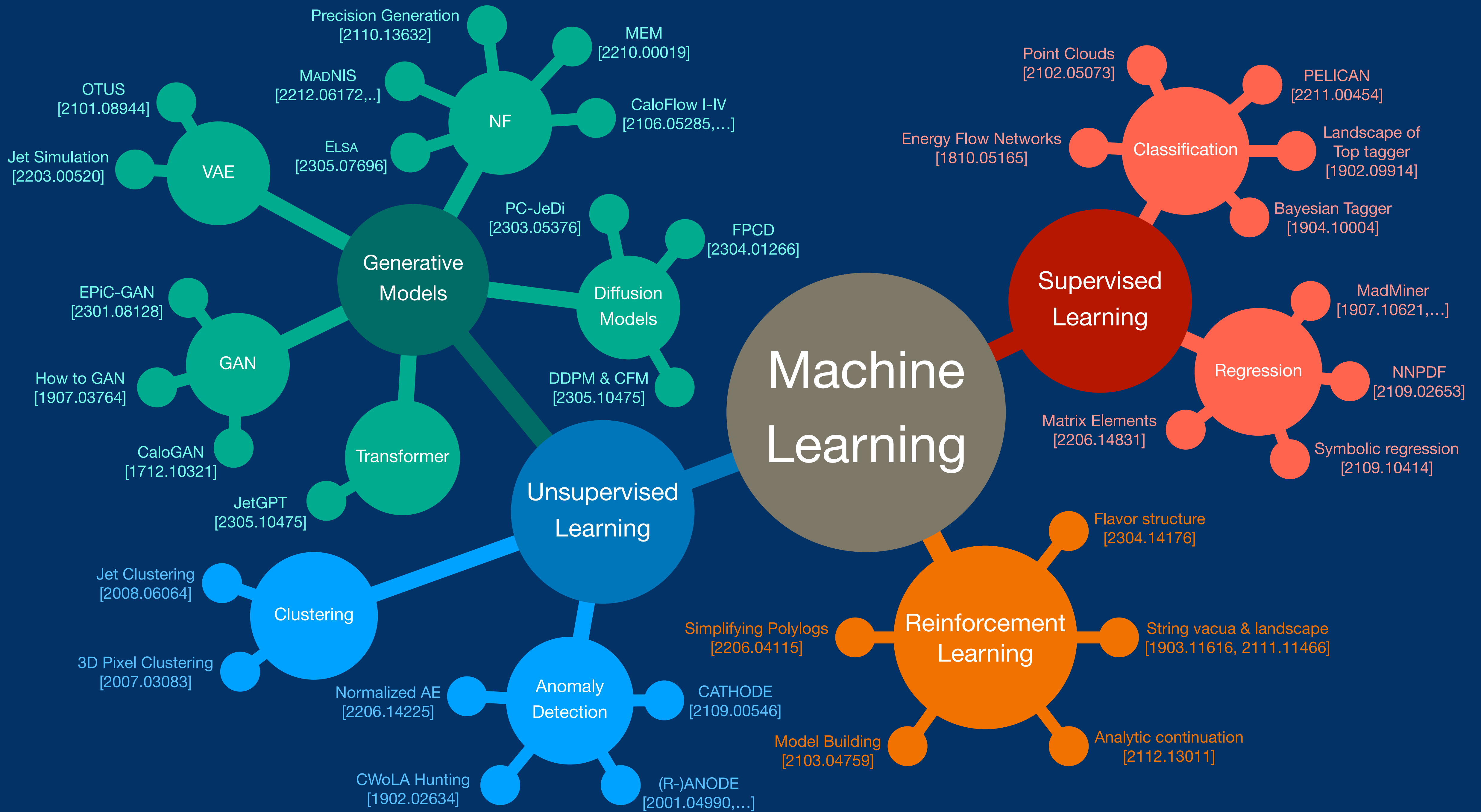
- **TensorFlow, PyTorch, Jax:** For deep learning models

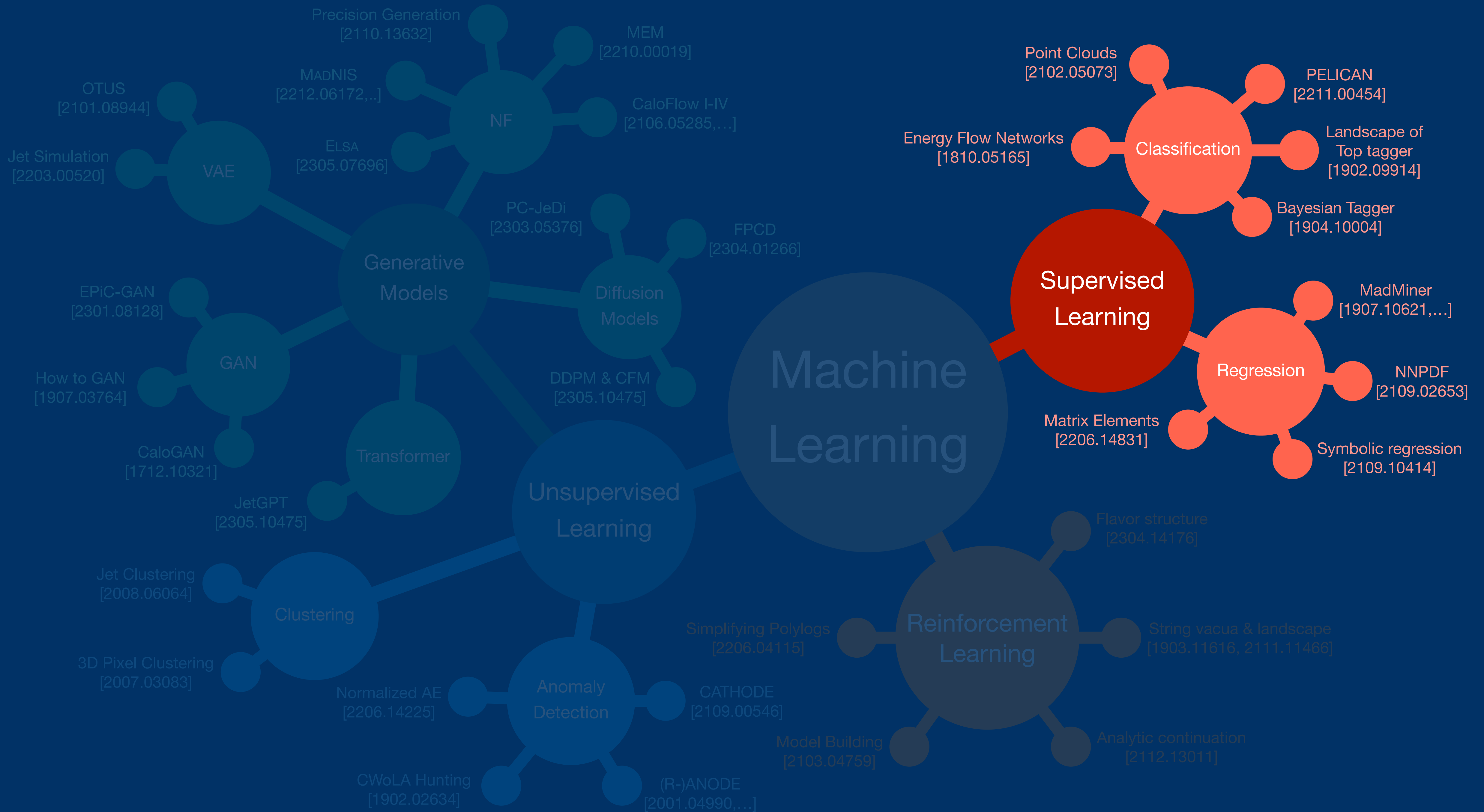




Questions?

The Landscape of Machine Learning

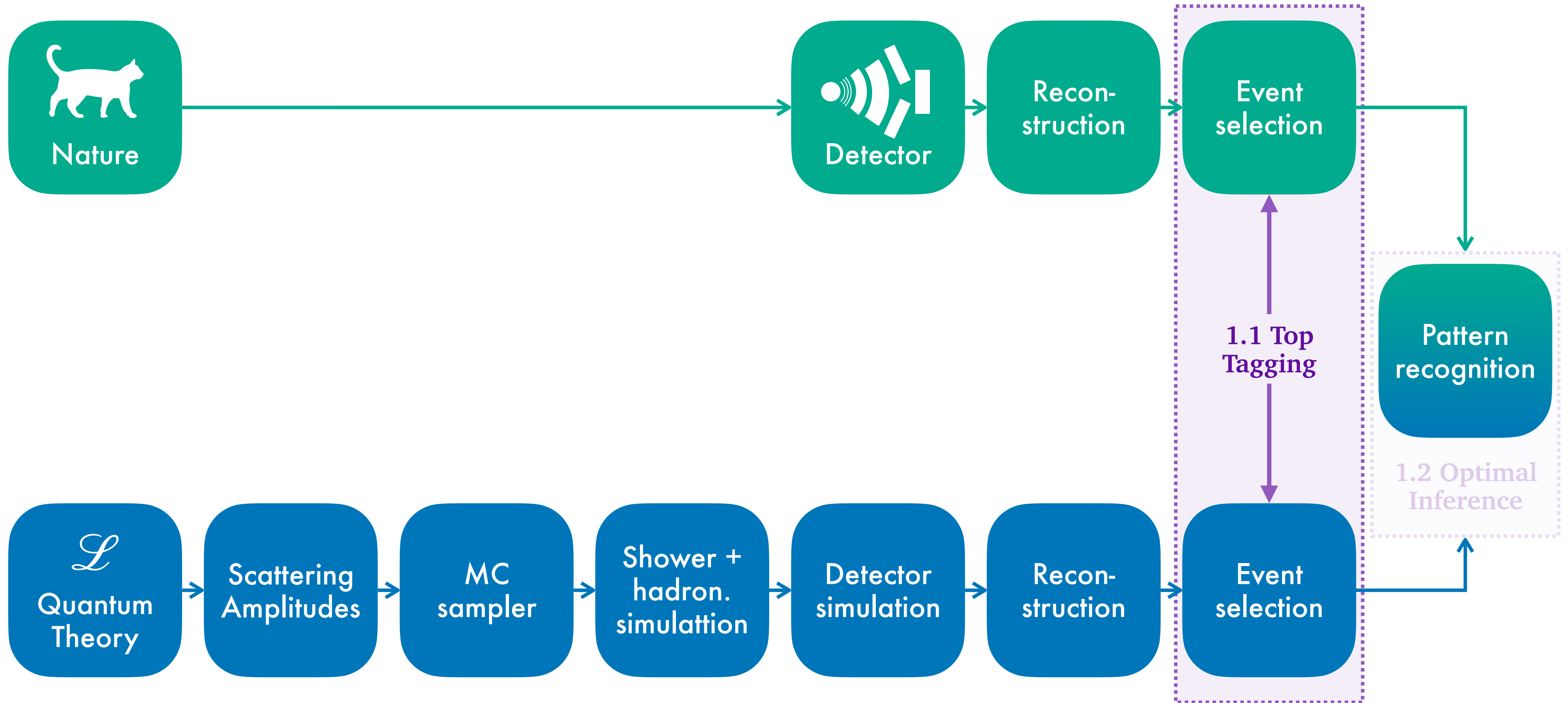




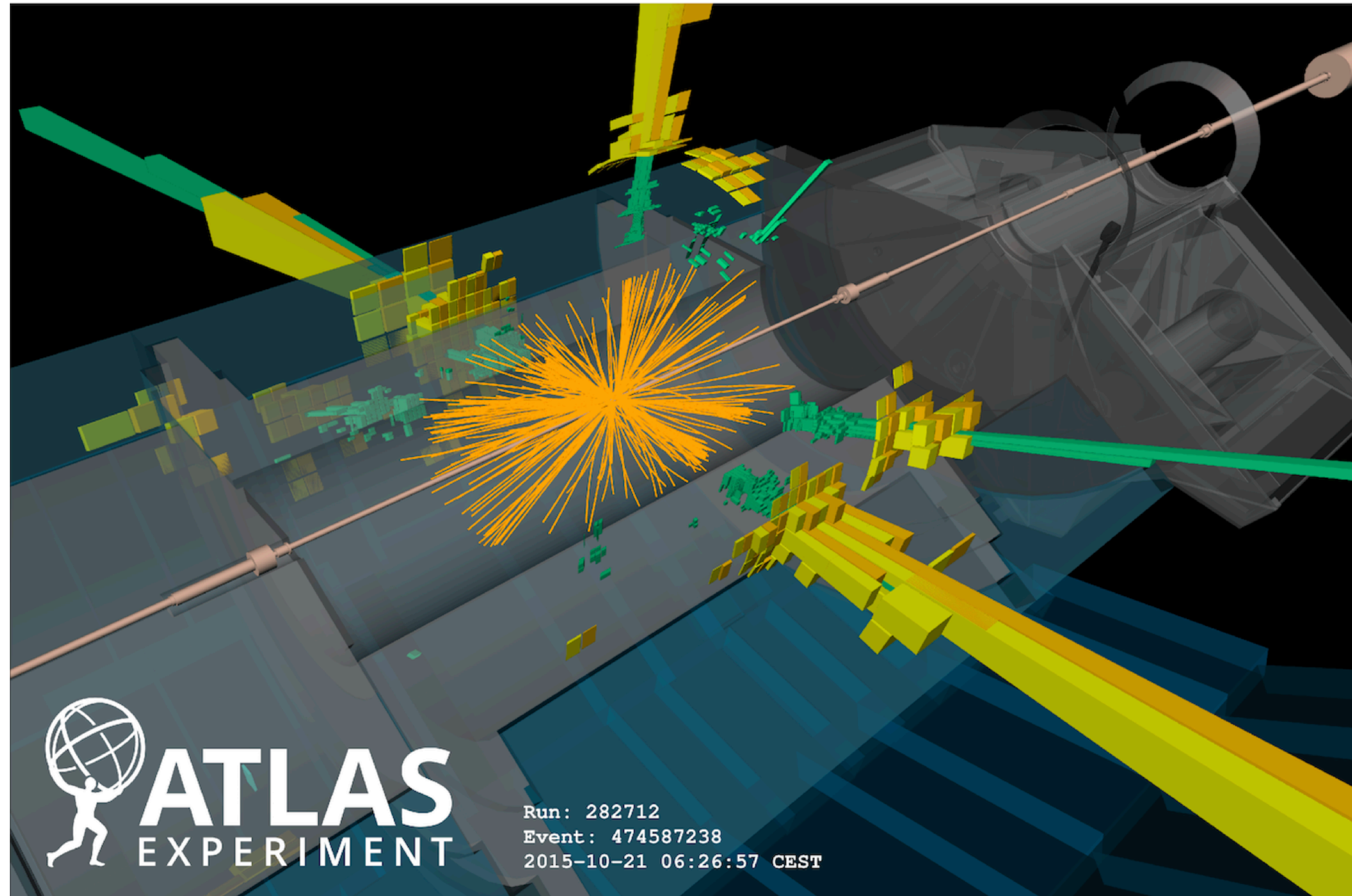
Example I

Classification with Top Tagging

LHC analysis + ML



What is jet tagging?



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

What is jet tagging?

The image shows a 3D visualization of a particle collision event. A central orange starburst represents the collision point. Several jets are shown as colored cones extending from the center: a green jet pointing towards the top-left, a yellow jet pointing towards the top-right, a cyan jet pointing towards the right, and a yellow jet pointing towards the bottom-right. Callout boxes with purple borders and white text point to these jets with lines. The ATLAS Experiment logo is in the bottom-left, and event details are in the bottom-center.

Gloun jet?

Light quark jet?

Why?

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

Bottom quark jet?

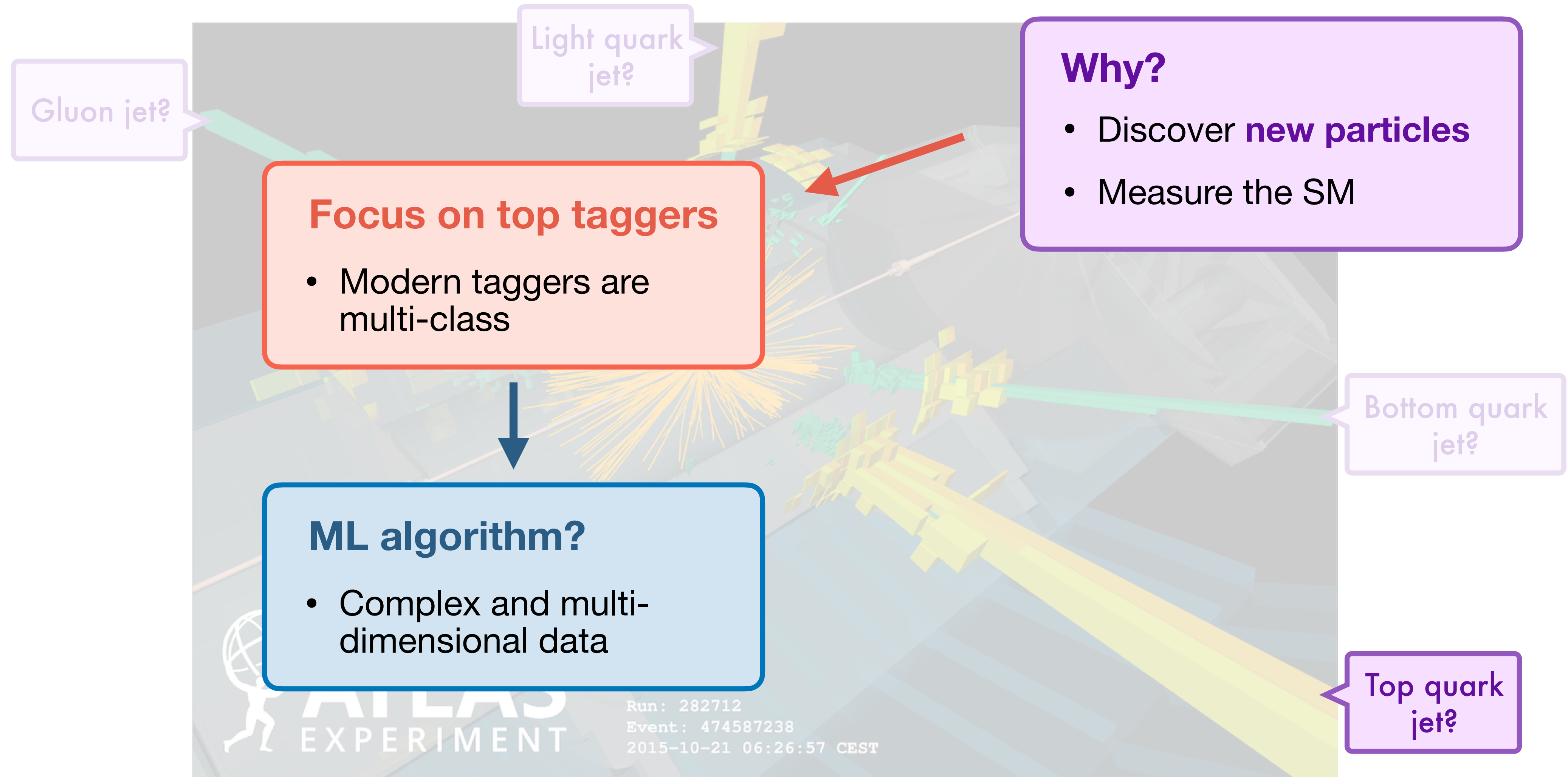
Top quark jet?

ATLAS EXPERIMENT

Run: 282712
Event: 474587238
2015-10-21 06:26:57 CEST

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

What is jet tagging?



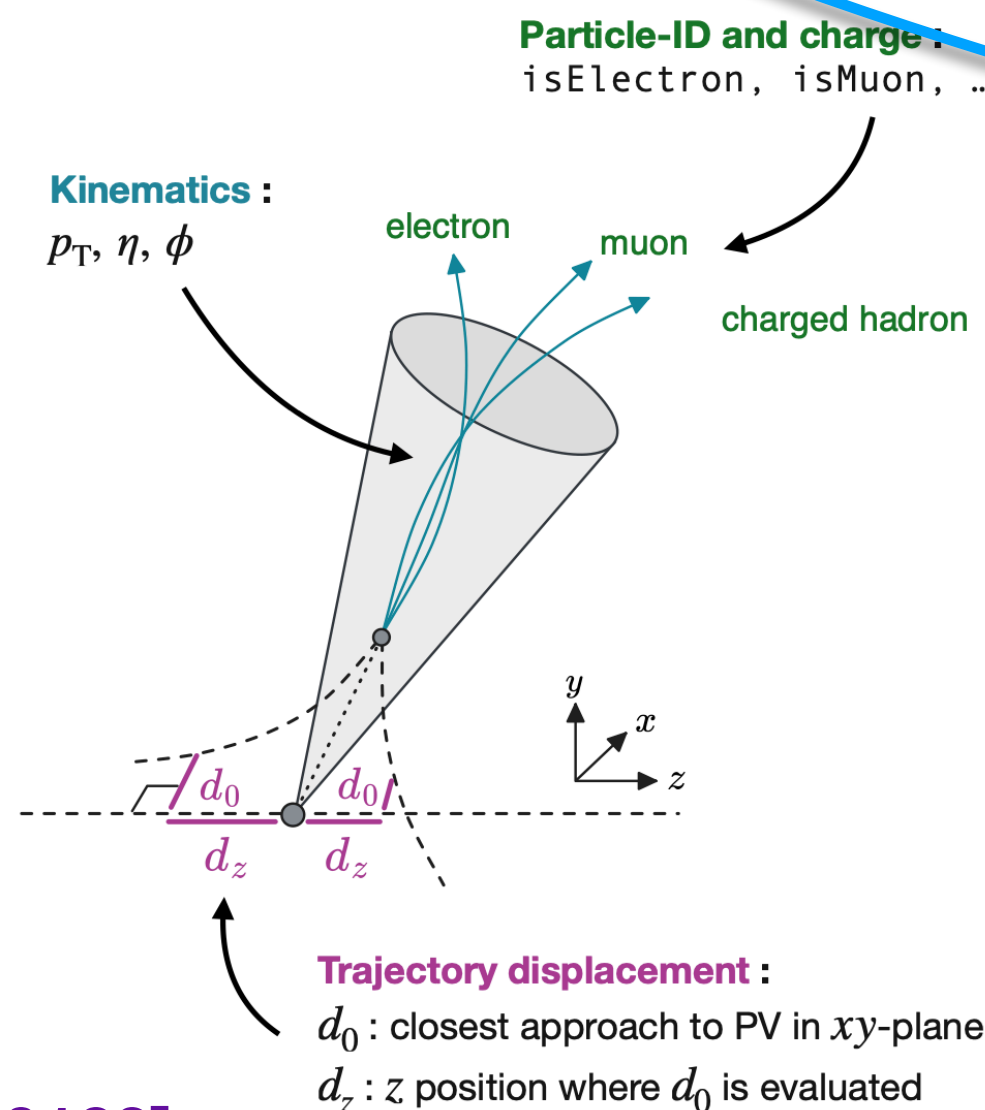
We want to know **which particle** produced the jet!

What is jet tagging?

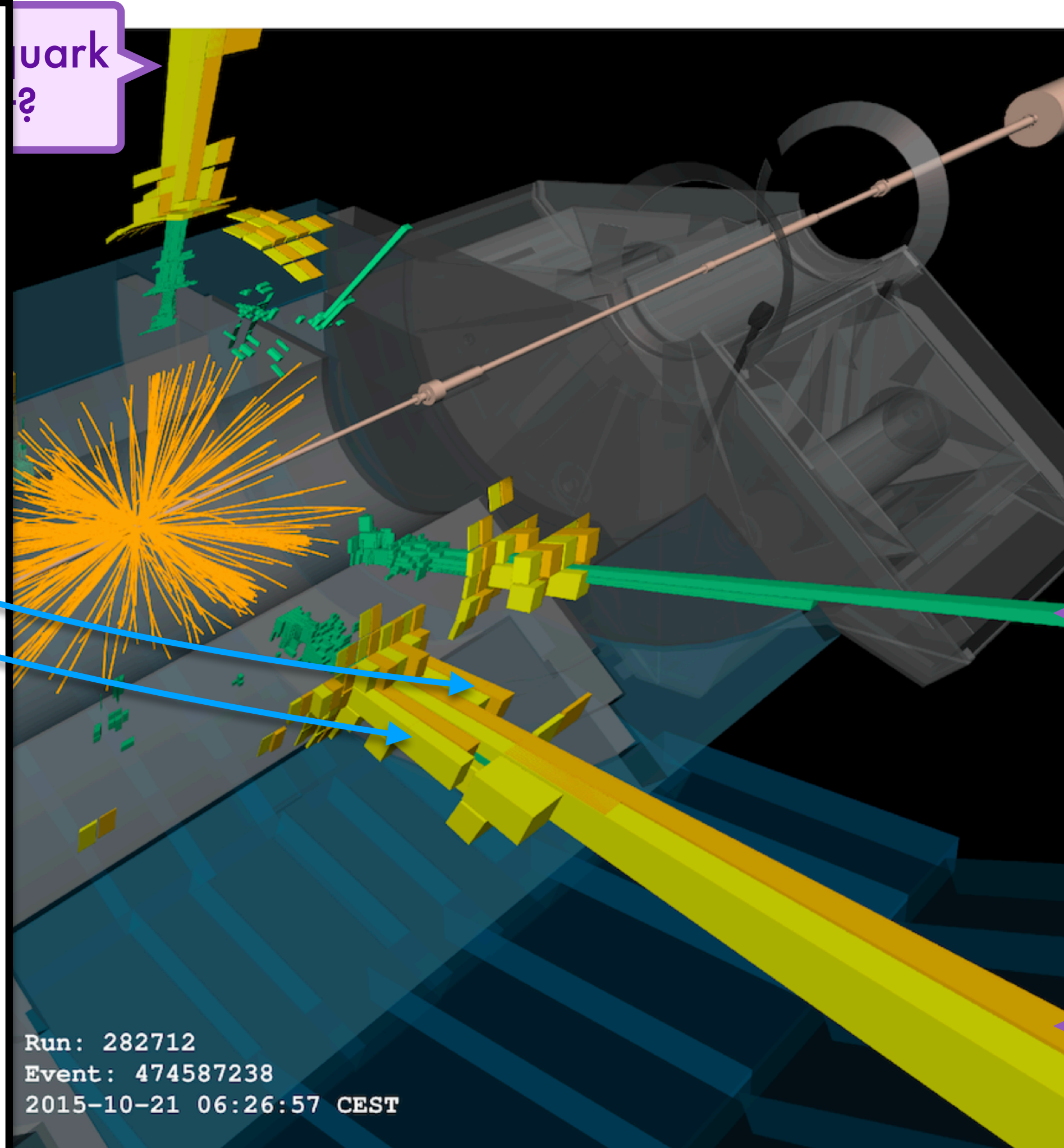
Data most naturally as **point cloud**

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

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



[2312.00123]



quark?
?

Bottom quark jet?

Top quark jet?

Run: 282712
Event: 474587238
2015-10-21 06:26:57 CEST

Landscape Dataset

39

- **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. Butter², K. Cranmer³, D. Debnath⁴, B. M. Dillon⁵, M. Fairbairn⁶, D. A. Faroughy⁵, W. Fedorko⁷, C. Gay⁷, L. Gouskos⁸, J. F. Kamenik^{5,9}, P. T. Komiske¹⁰, S. Leiss¹, A. Lister⁷, S. Macaluso^{3,4}, E. M. Metodiev¹⁰, L. Moore¹¹, B. Nachman,^{12,13} K. Nordström^{14,15}, J. Pearkes⁷, H. Qu⁸, Y. Rath¹⁶, M. Rieger¹⁶, D. Shih⁴, J. M. Thompson², and S. Varma⁶

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, Universitèx 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

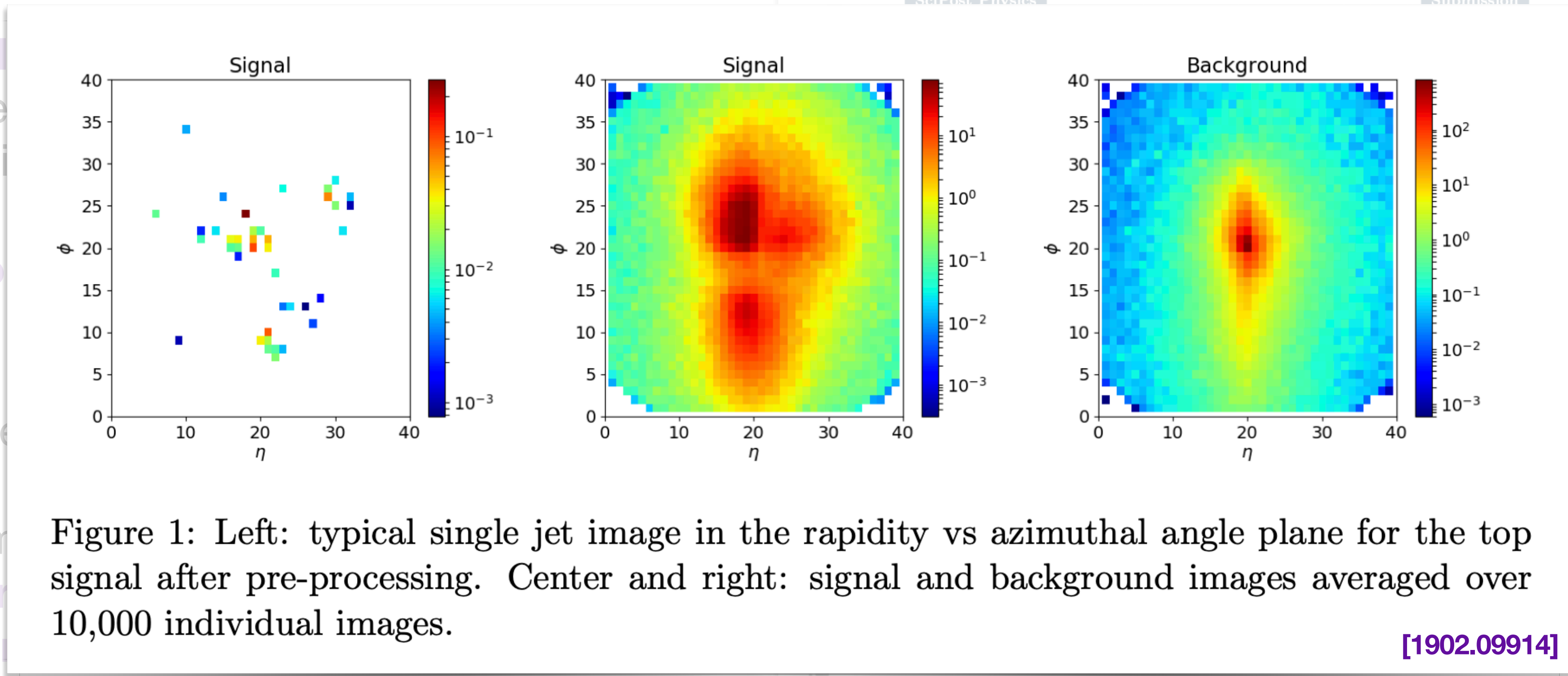
Based on the established task of identifying boosted, hadronically decaying top quarks, we compare a wide range of modern machine learning approaches. Unlike most established methods they rely on low-level input, for instance calorimeter output. While their network architectures are vastly different,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

[1902.09914]

arXiv:1902.09914v3 [hep-ph] 23 Jul 2019

Landscape Dataset

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



[1902.09914]

arX

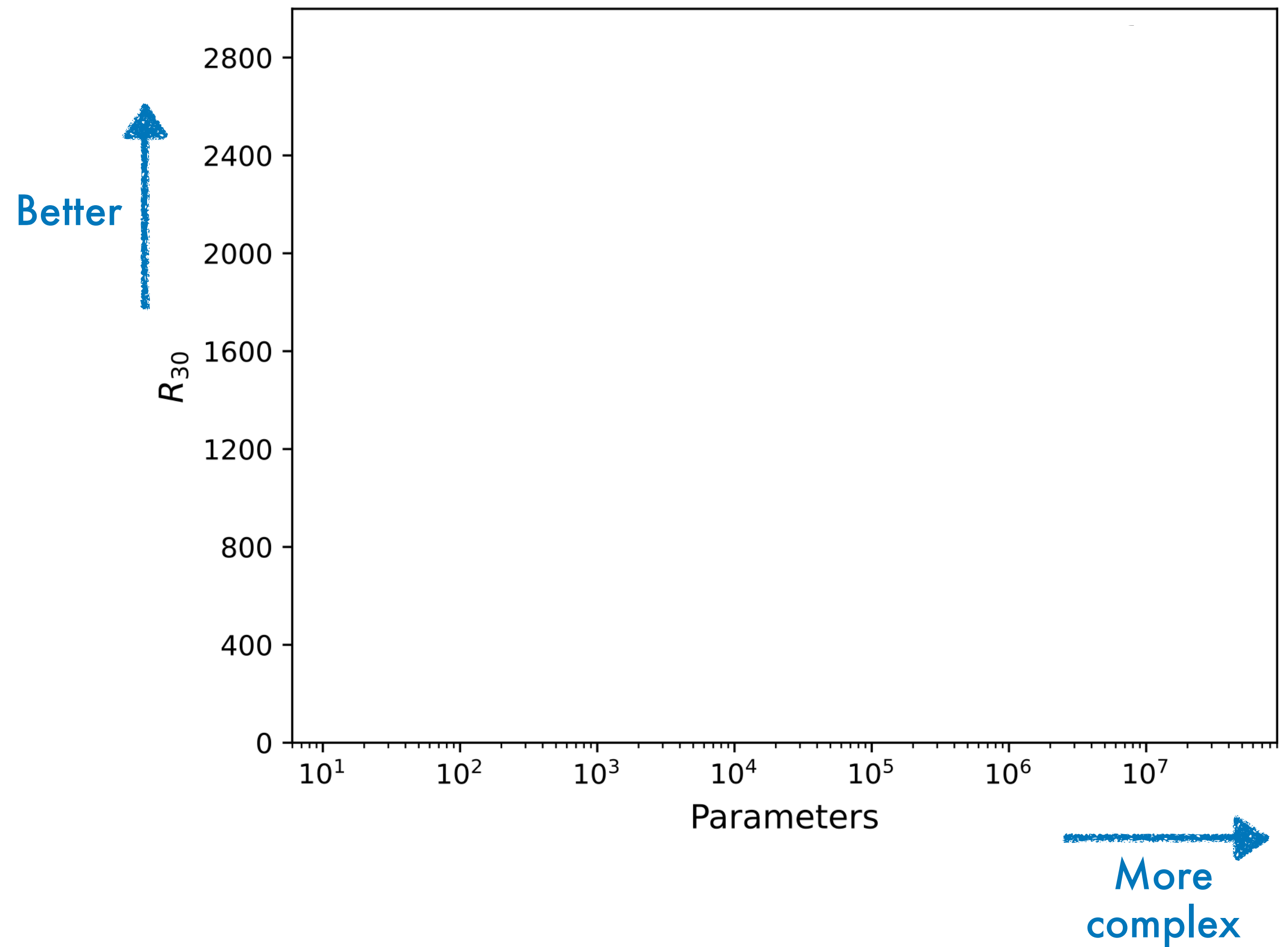
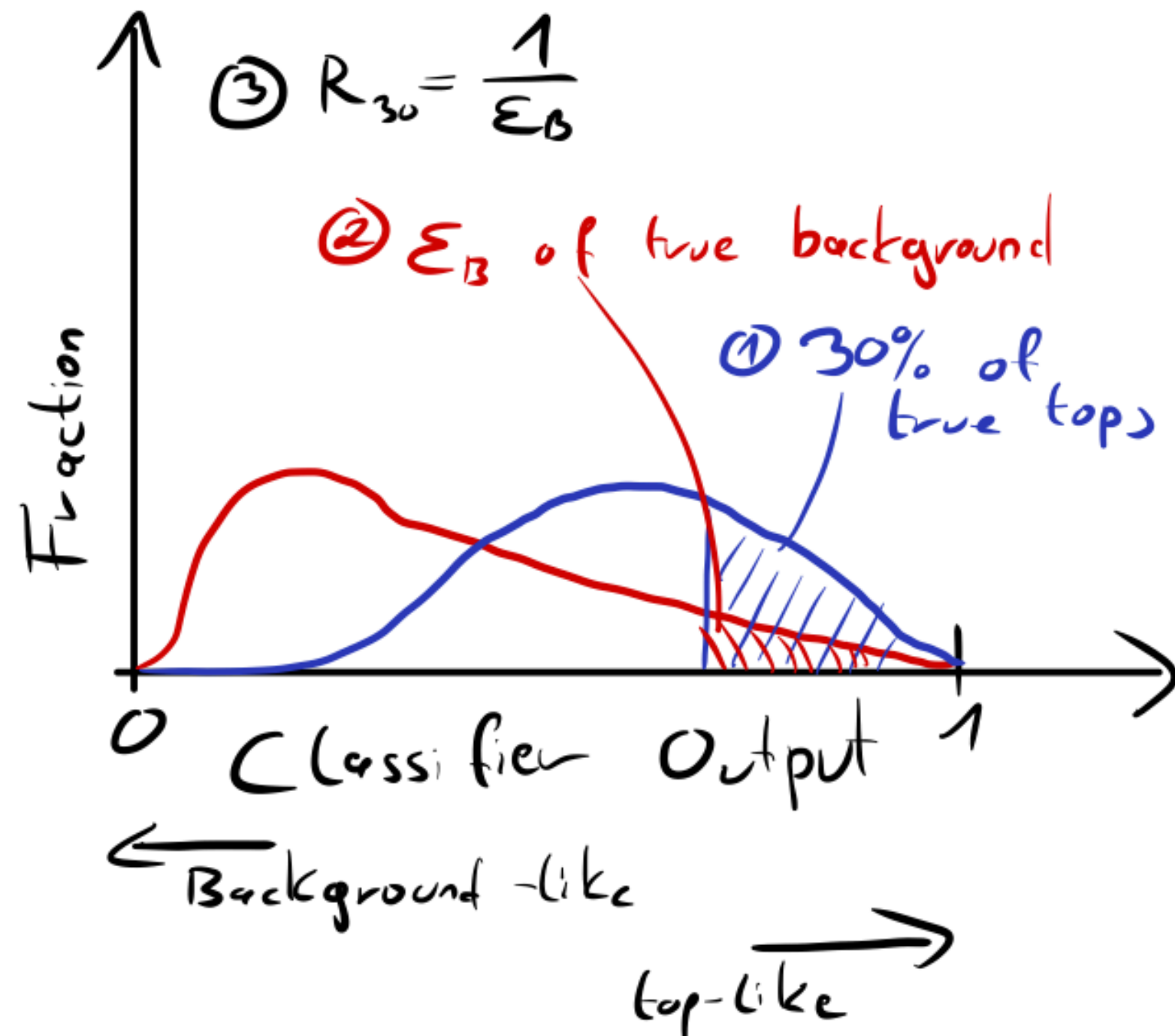
Based on the established task of identifying boosted, hadronically decaying top quarks, we compare a wide range of modern machine learning approaches. Unlike most established methods they rely on low-level input, for instance calorimeter output. While their network architectures are vastly different, 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

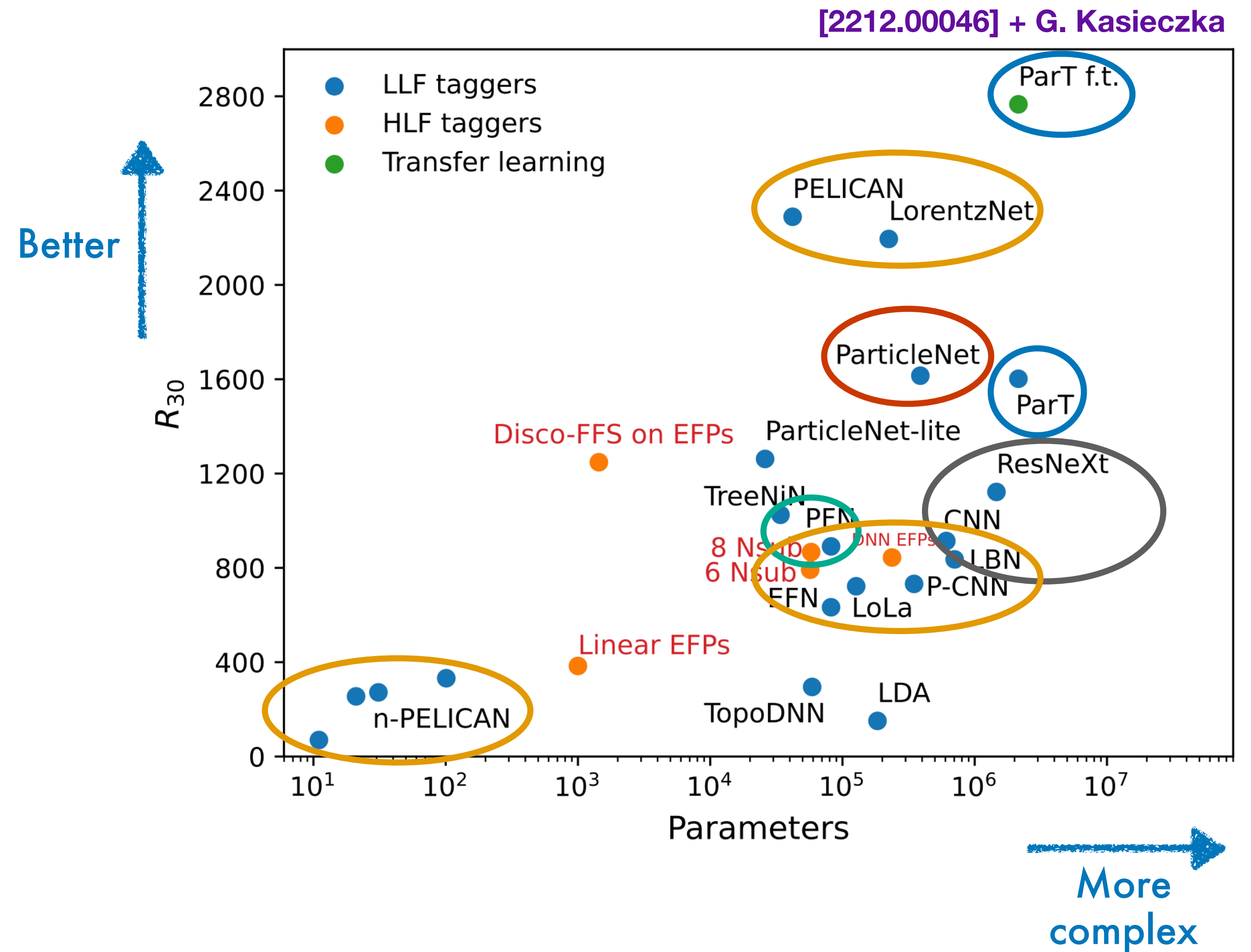
Cut so that **30%** of top quarks pass selection:

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



Evaluation metrics and results

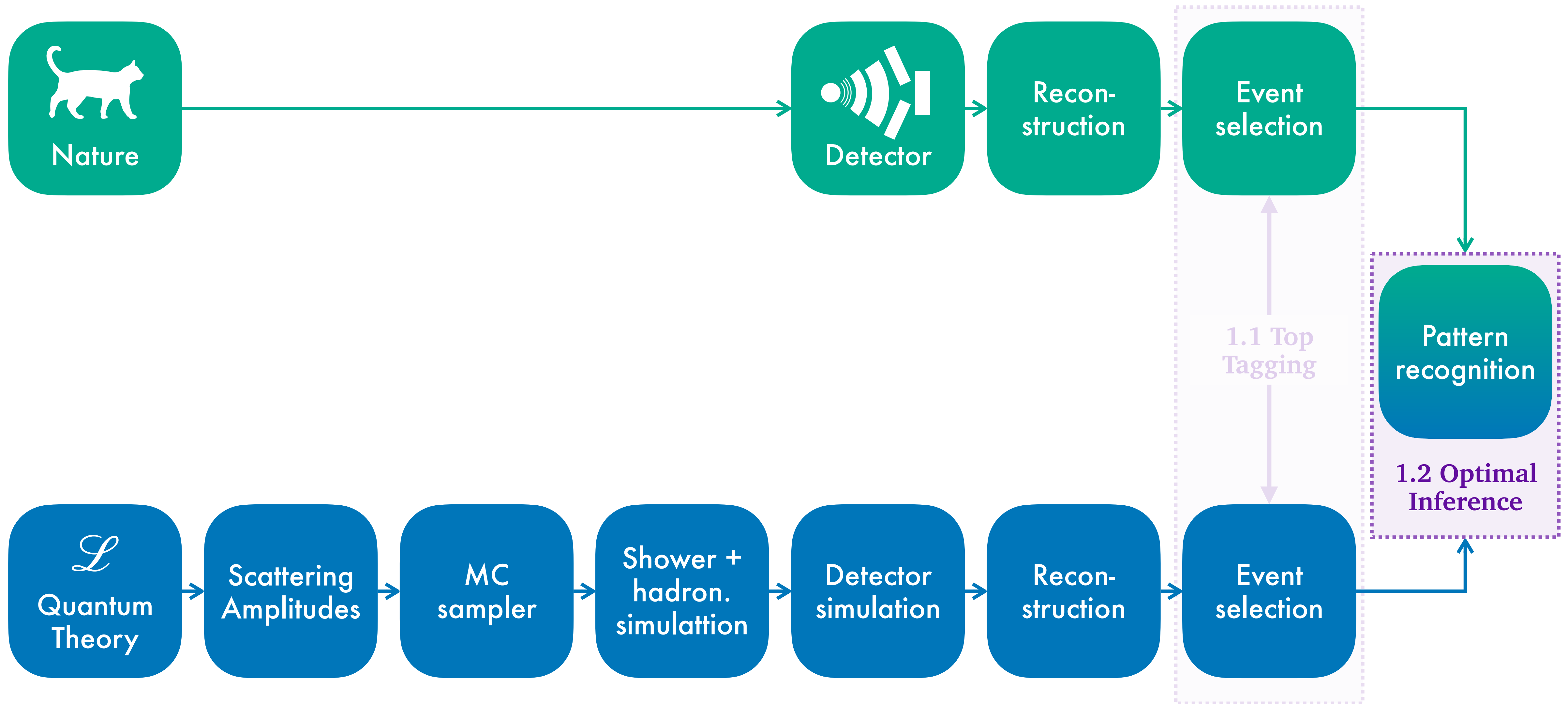
- **Jets as images**
[1701.08784, 1803.00107]
- **Point cloud**
[PFN 1810.05165]
- **Point cloud with Graphs**
[1902.08570, 2007.13681]
- **Point cloud with attention**
[2202.03772]
- **Lorentz symmetries**
[1707.08966, 2201.08187, 2211.00454]



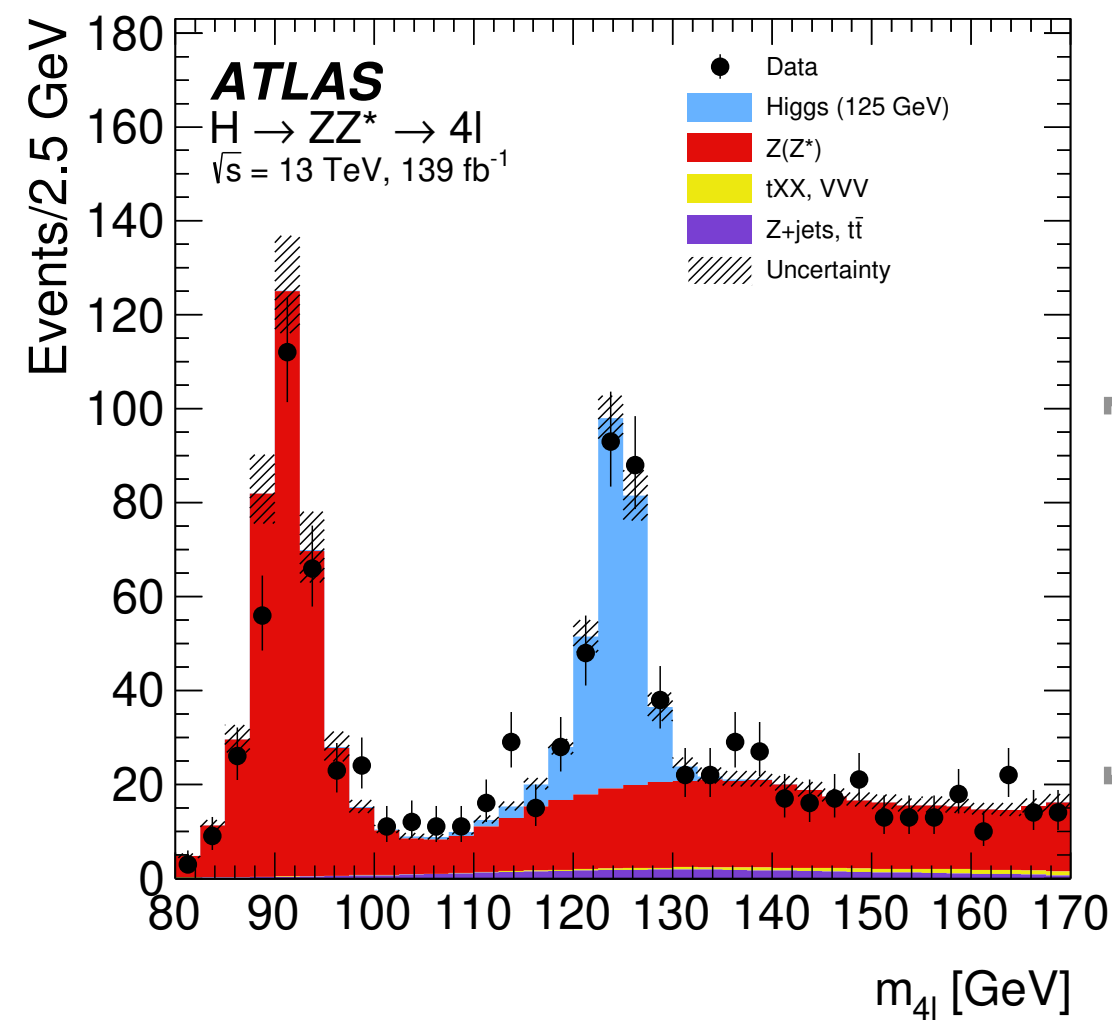
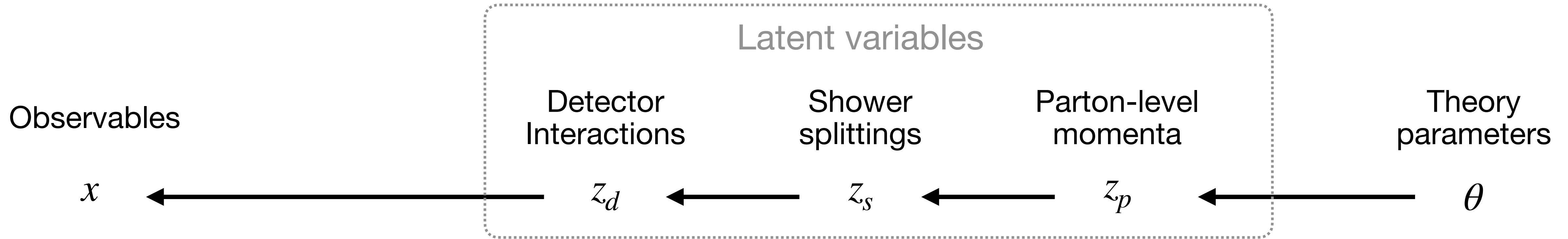
Example II

Regression with MadMiner

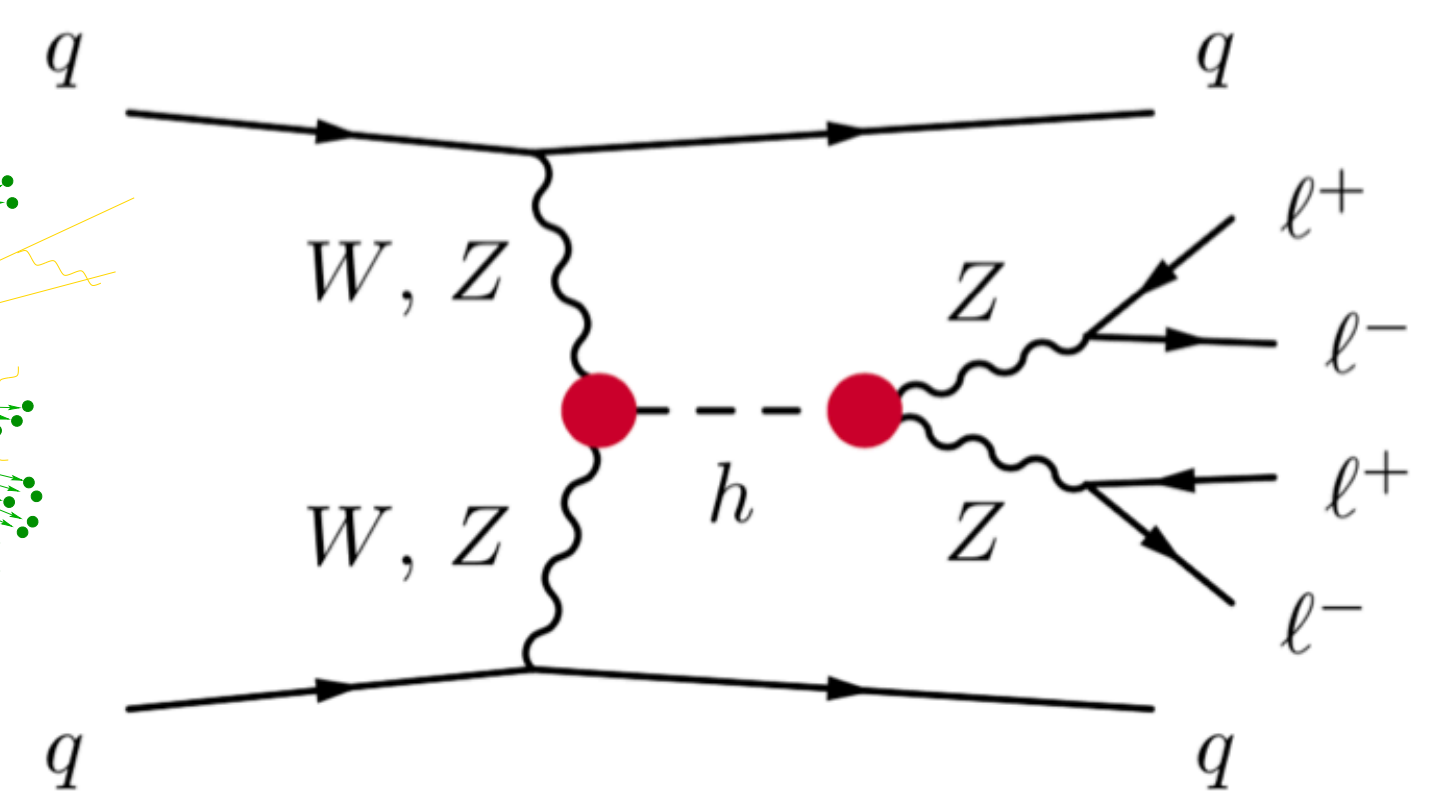
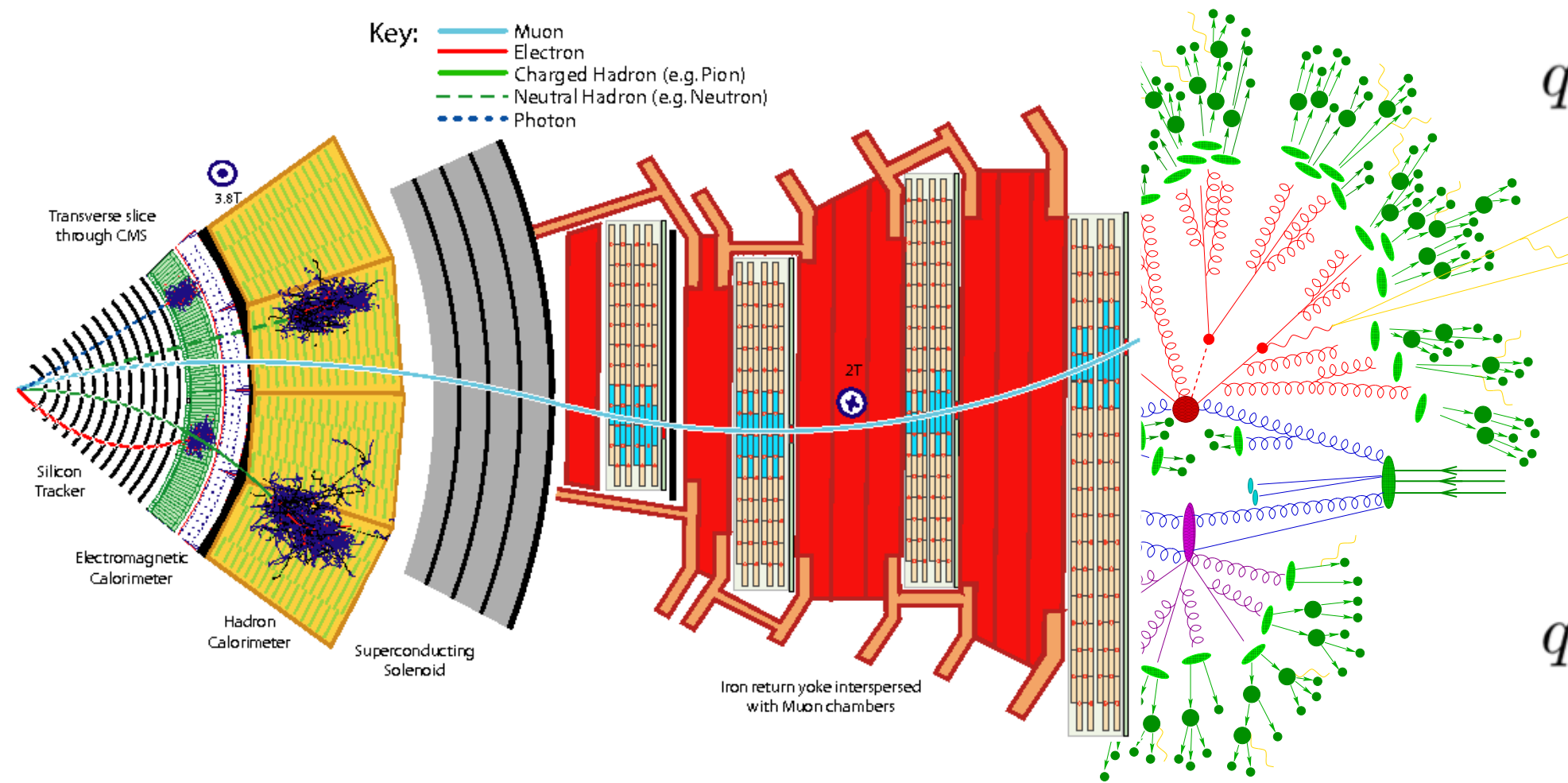
LHC analysis + ML



Recap — HEP analysis

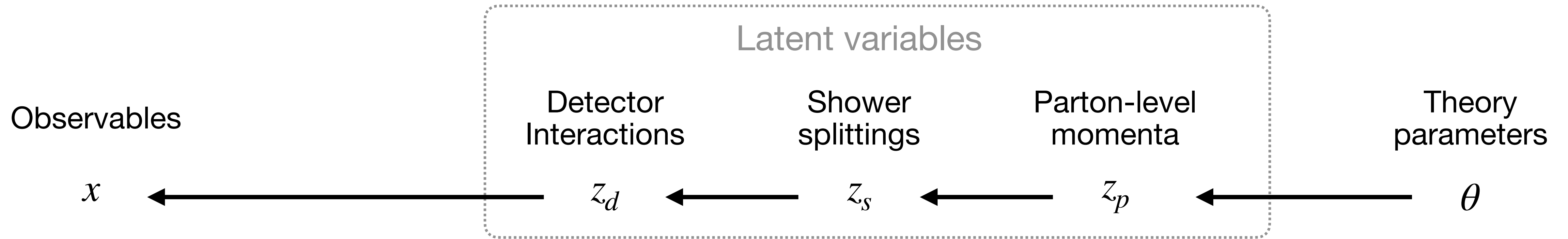


ATLAS [1712.02304]



← Evolution →

Recap – HEP analysis



Sample from

$p(x | z_d)$

$p(z_d | z_s)$

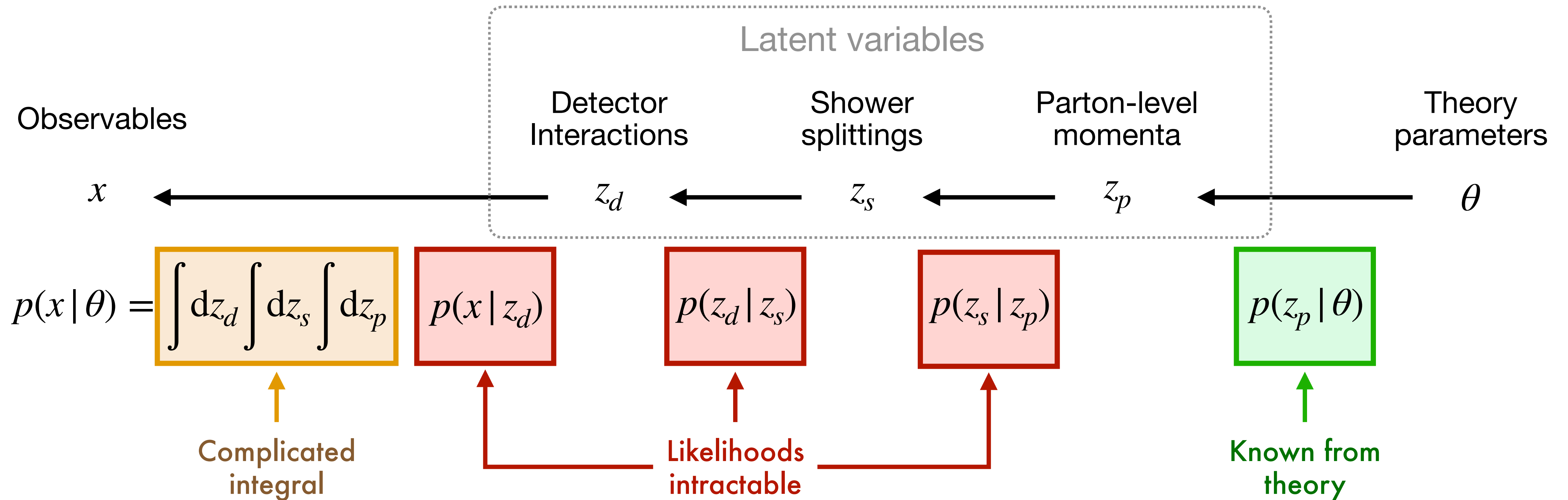
$p(z_s | z_p)$

$p(z_p | \theta)$



← Prediction (Simulation)

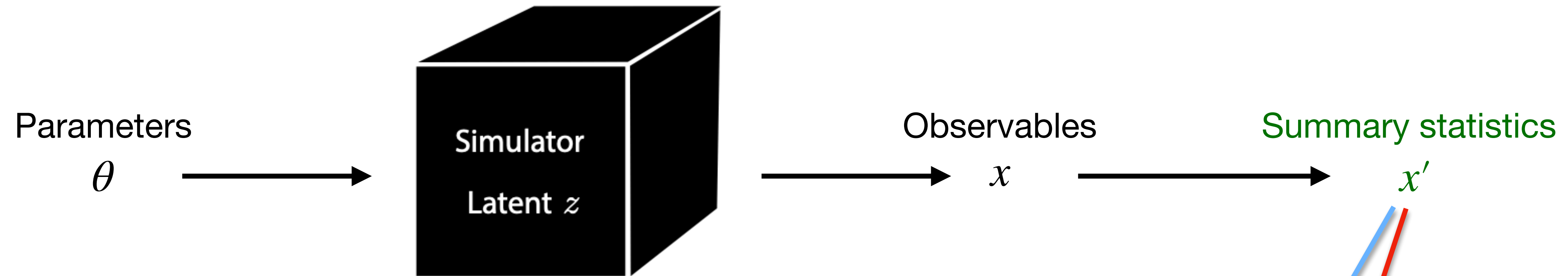
Recap — HEP analysis



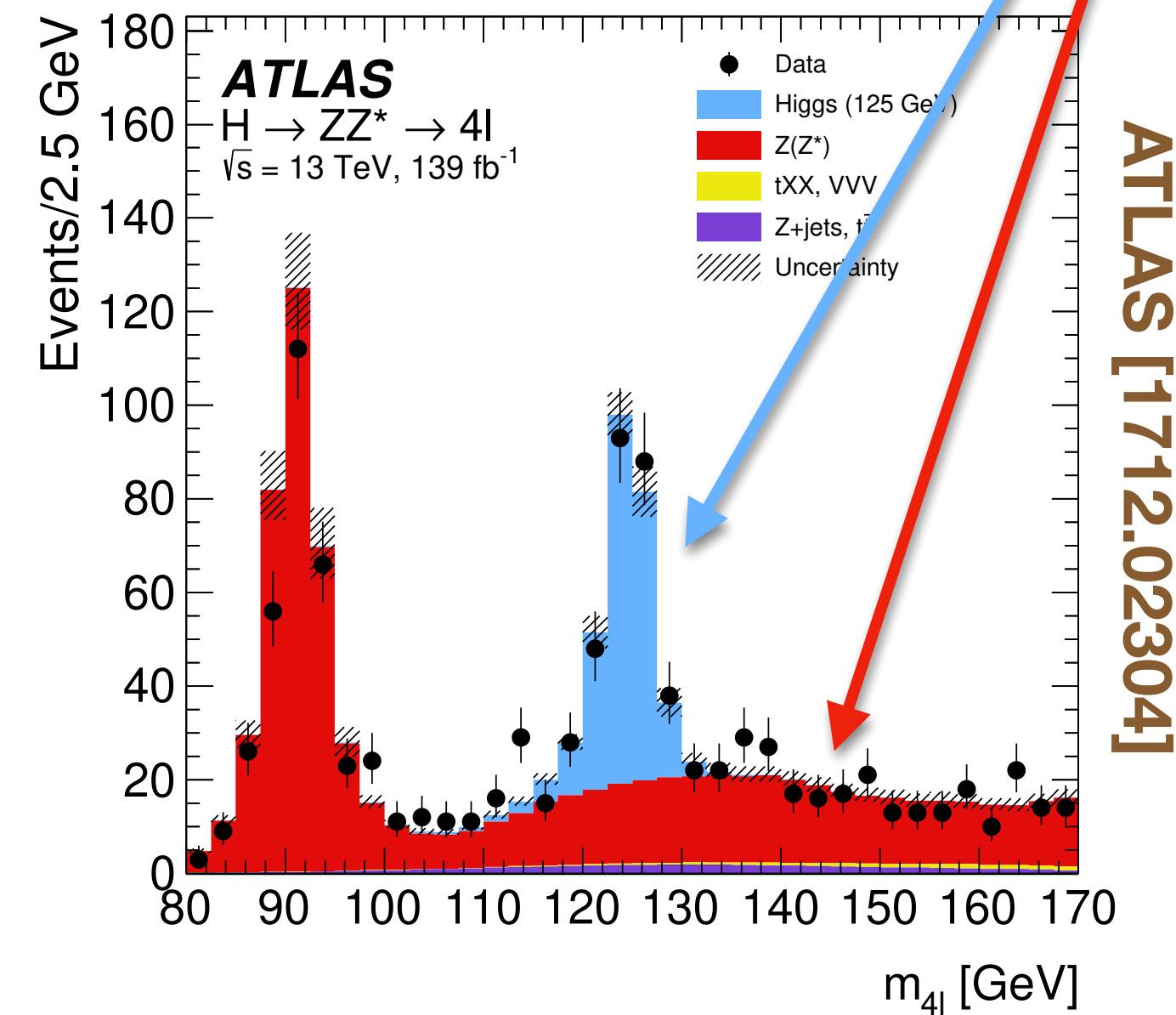
Why has that not stopped us so far?

→ Inference

Solve it by histogramming summary statistics



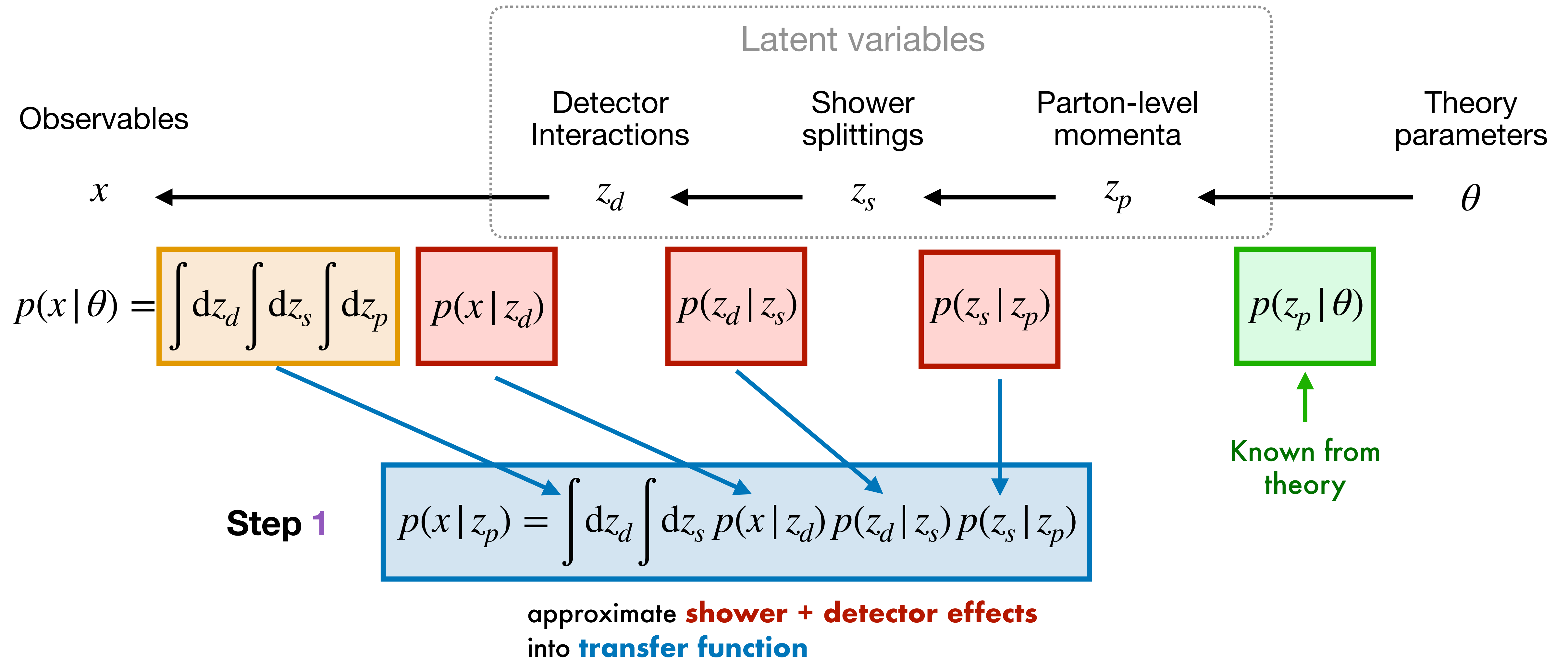
$$\hat{p}(x | \theta) = p(x' | \theta) =$$



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

Observables

Detector Interactions

Shower splittings

Parton-level momenta

Theory parameters

Latent variables

Matrix Element Method [K. Kondo 1988] + **ML** [2210.00019, 2310.07752]

Lecture III

θ

$$p(x | \theta) = \int dz_p$$

(Less) complicated integral

$$p(x | z_p)$$

Likelihood still intractable

$$p(z_p | \theta)$$

Known from theory

Step 2

$$\left\langle \frac{f(z_p)}{g_\omega(z_p)} \right\rangle$$

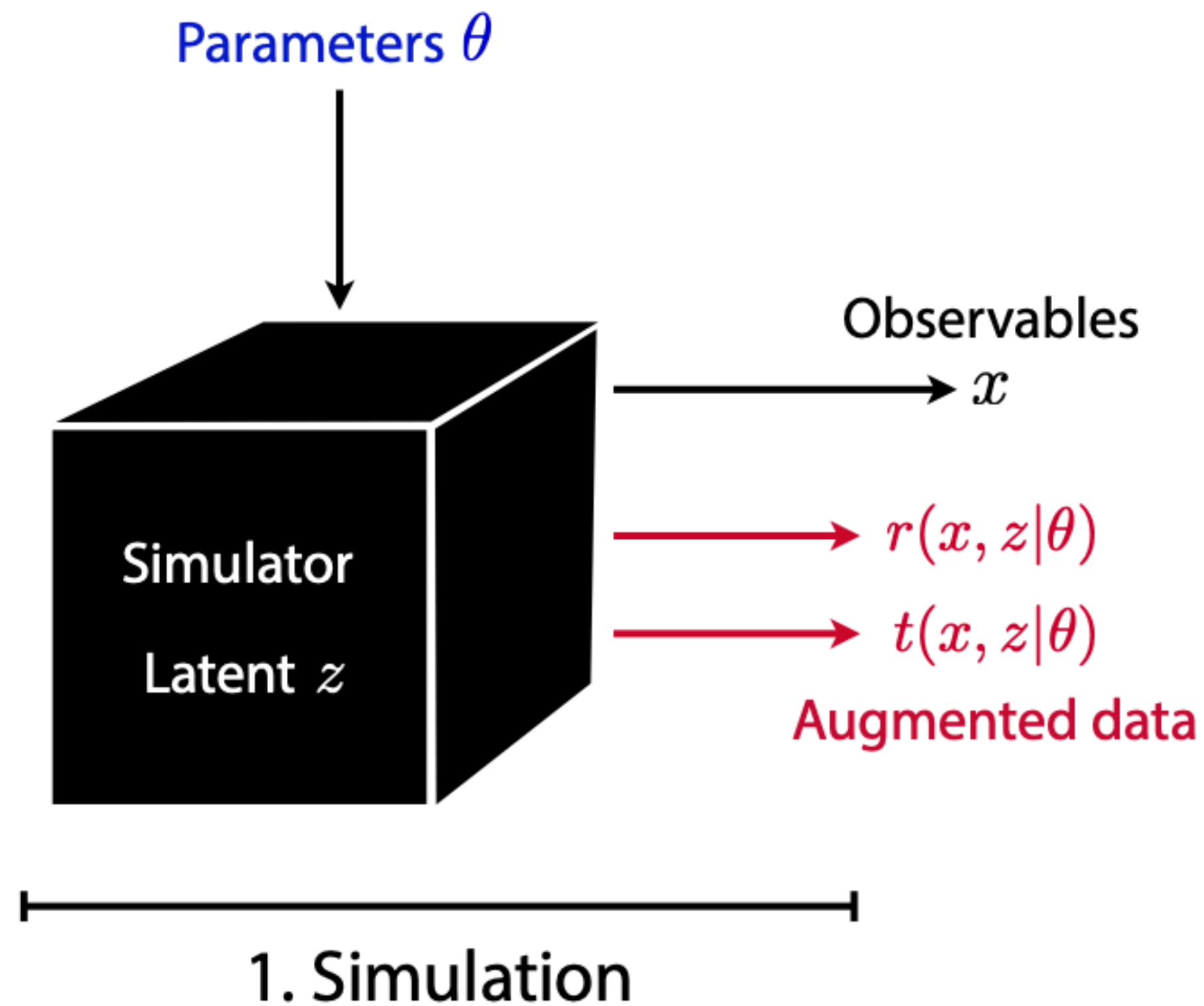
Use ML to speed-up integration

$$\hat{p}_\omega(x | z_p)$$

Use ML to parametrize transfer function

Solving it with likelihood-free inference

The MadMiner approach

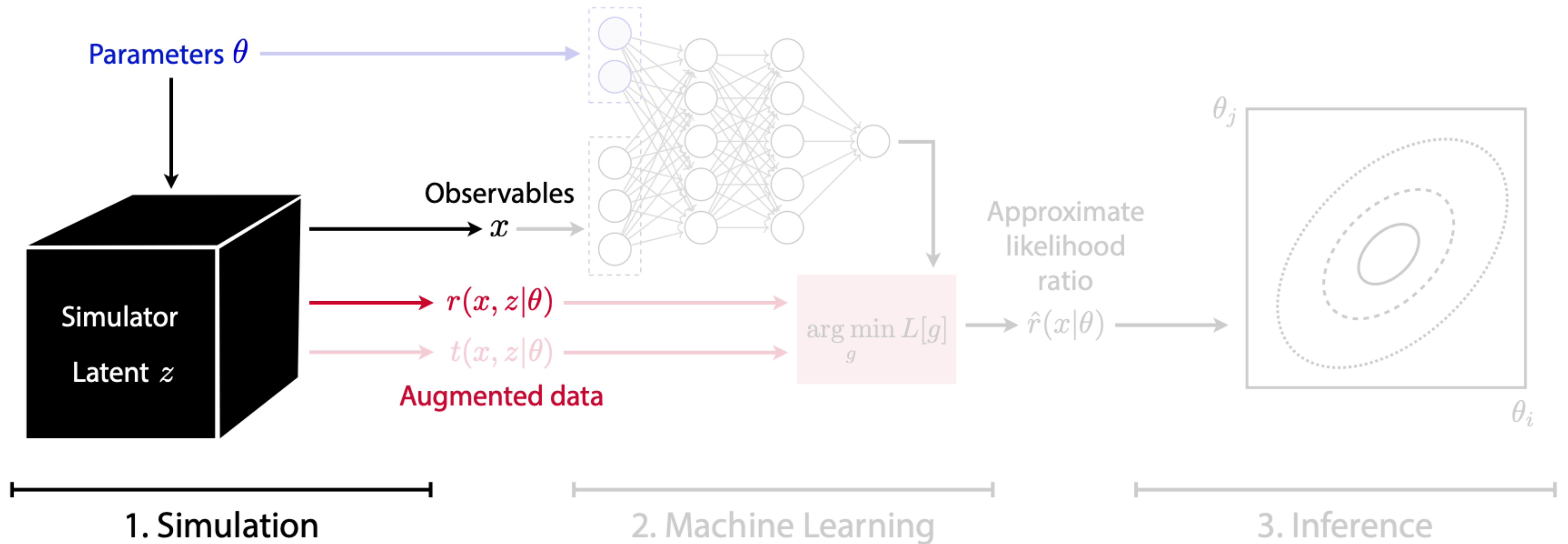


“Mining gold”: Extract additional information from simulator

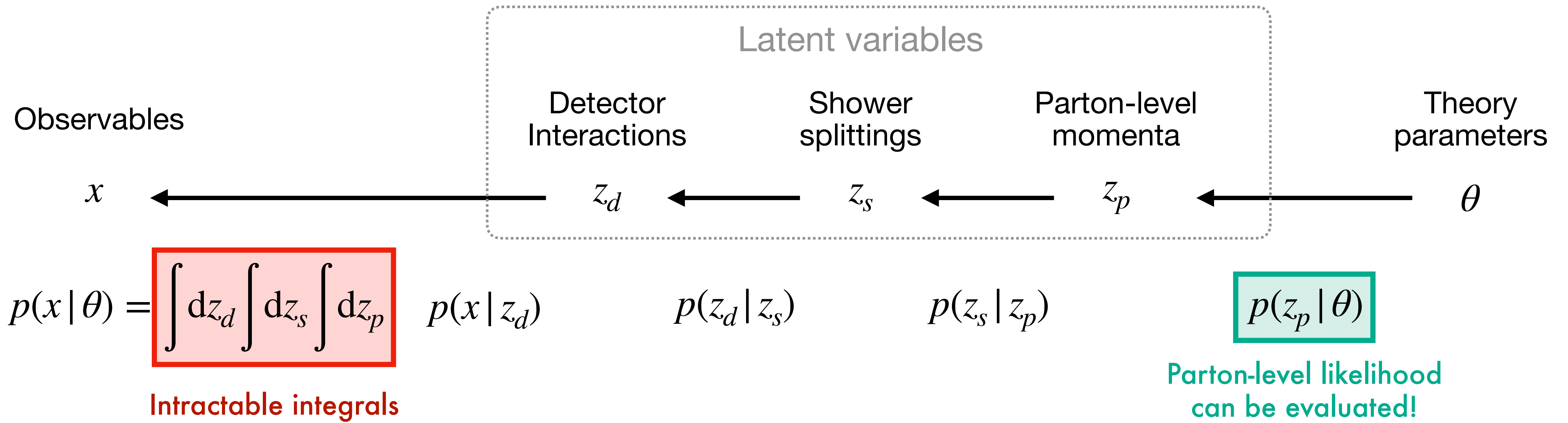
Use this information to train estimator for likelihood ratio

Set new constraints with standard hypothesis tests

The MadMiner approach



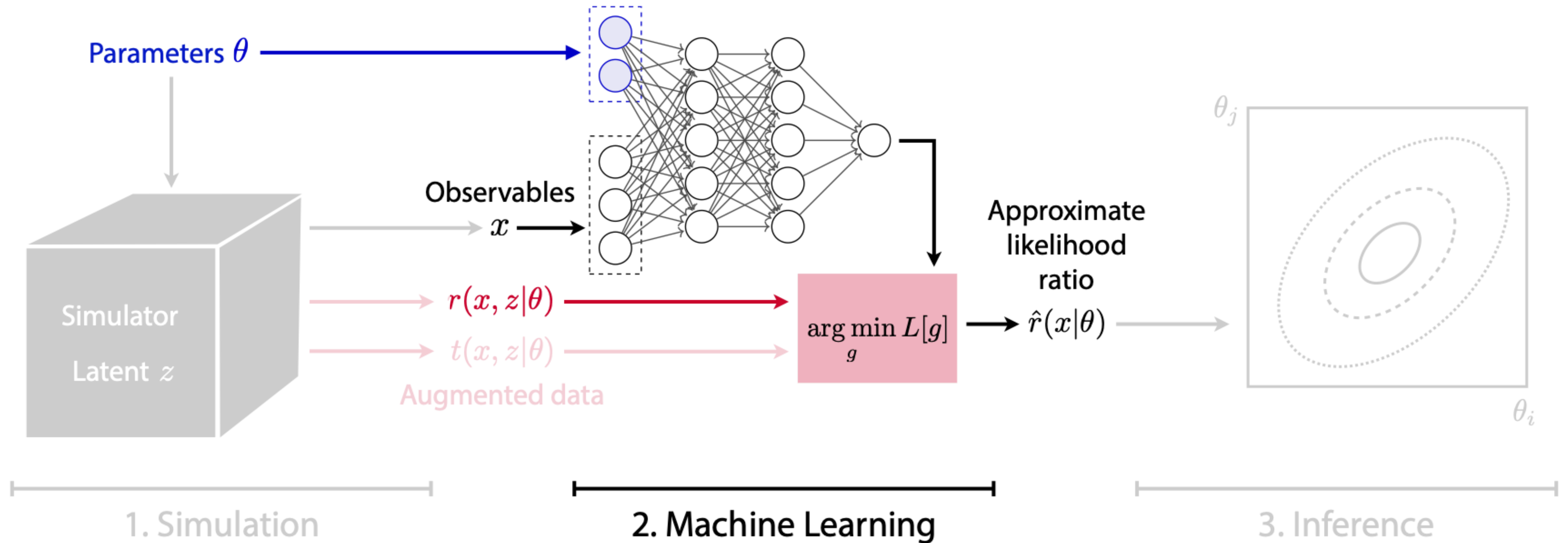
Mining gold from the simulator



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

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

The MadMiner approach



The value of gold

We can calculate the **joint likelihood ratio**

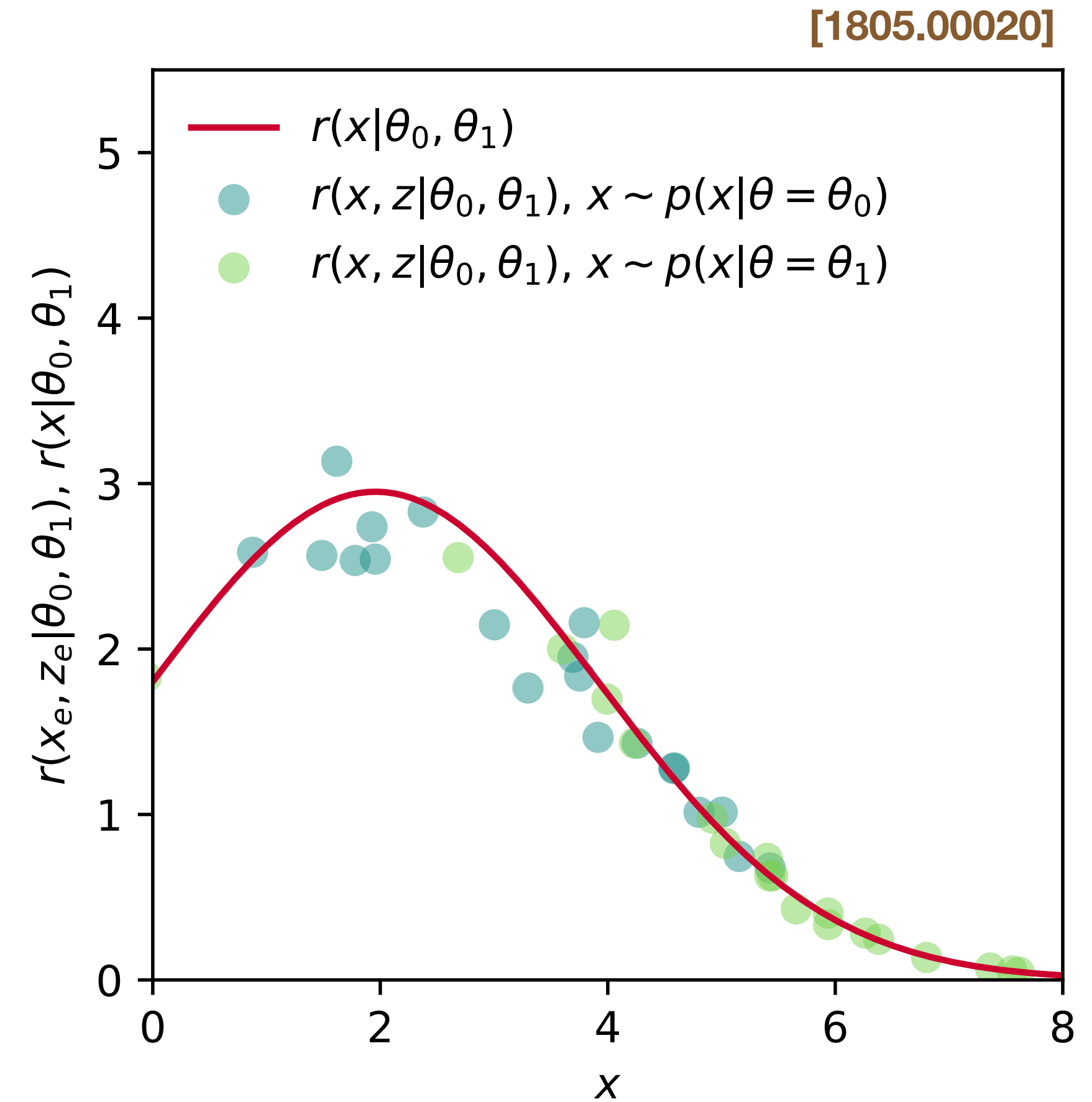
$$r(x, z | \theta_0, \theta_1) \equiv \frac{p(x, z_d, z_s, z_p | \theta_0)}{p(x, z_d, z_s, z_p | \theta_1)}$$



We want the **likelihood ratio function**

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

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



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**

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

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

$$L_r[\hat{r}(x | \theta_0, \theta_1)] = \int dx \int dz p(x, z | \theta_1) [\hat{r}(x | \theta_0, \theta_1) - r(x, z | \theta_0, \theta_1)]^2$$

One can show it is minimized by

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

Machine learning = applied calculus

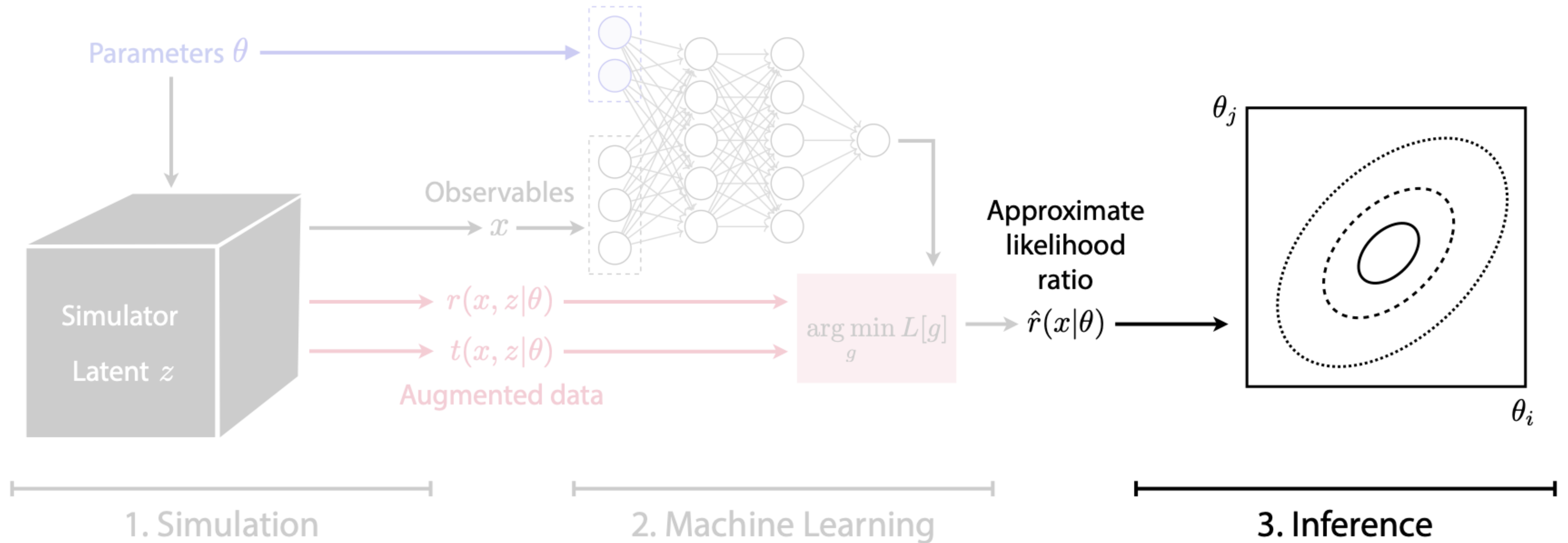
We can get a precise estimator by numerically minimizing a functional:

$$\hat{r}(x | \theta_0, \theta_1) = \arg \min_{\hat{r}(x | \theta_0, \theta_1)} \underbrace{\int dx \int dz p(x, z | \alpha_1) [\hat{r}(x | \theta_0, \theta_1) - r(x, z | \theta_0, \theta_1)]^2}_{L_r[\hat{r}(x | \theta_0, \theta_1)]}$$

We do this via **Machine Learning**:

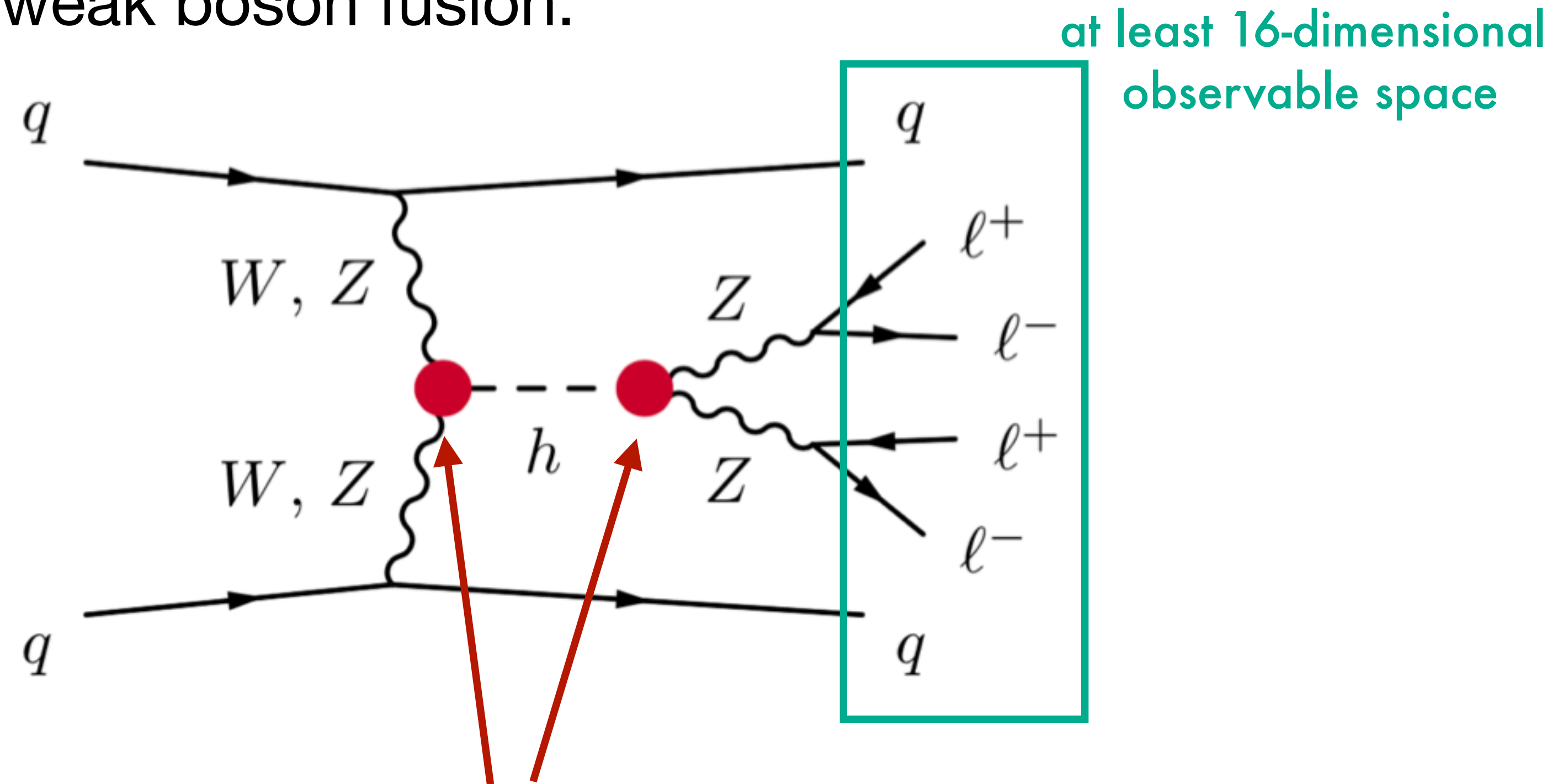
- Functional L_r \longrightarrow Loss function
$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_i [\hat{r}_\omega(x_i | \theta_0, \theta_1) - r(x_i, z_i | \theta_0, \theta_1)]^2$$
- Variational family $\hat{r}_\omega(x | \theta_0, \theta_1)$ \longrightarrow Flexible parametric function (e.g. neural network)
- Exact minimization \longrightarrow Numerical optimization algorithm (eg. Stochastic gradient descent)

The MadMiner approach



Constraining EFT parameters with ML

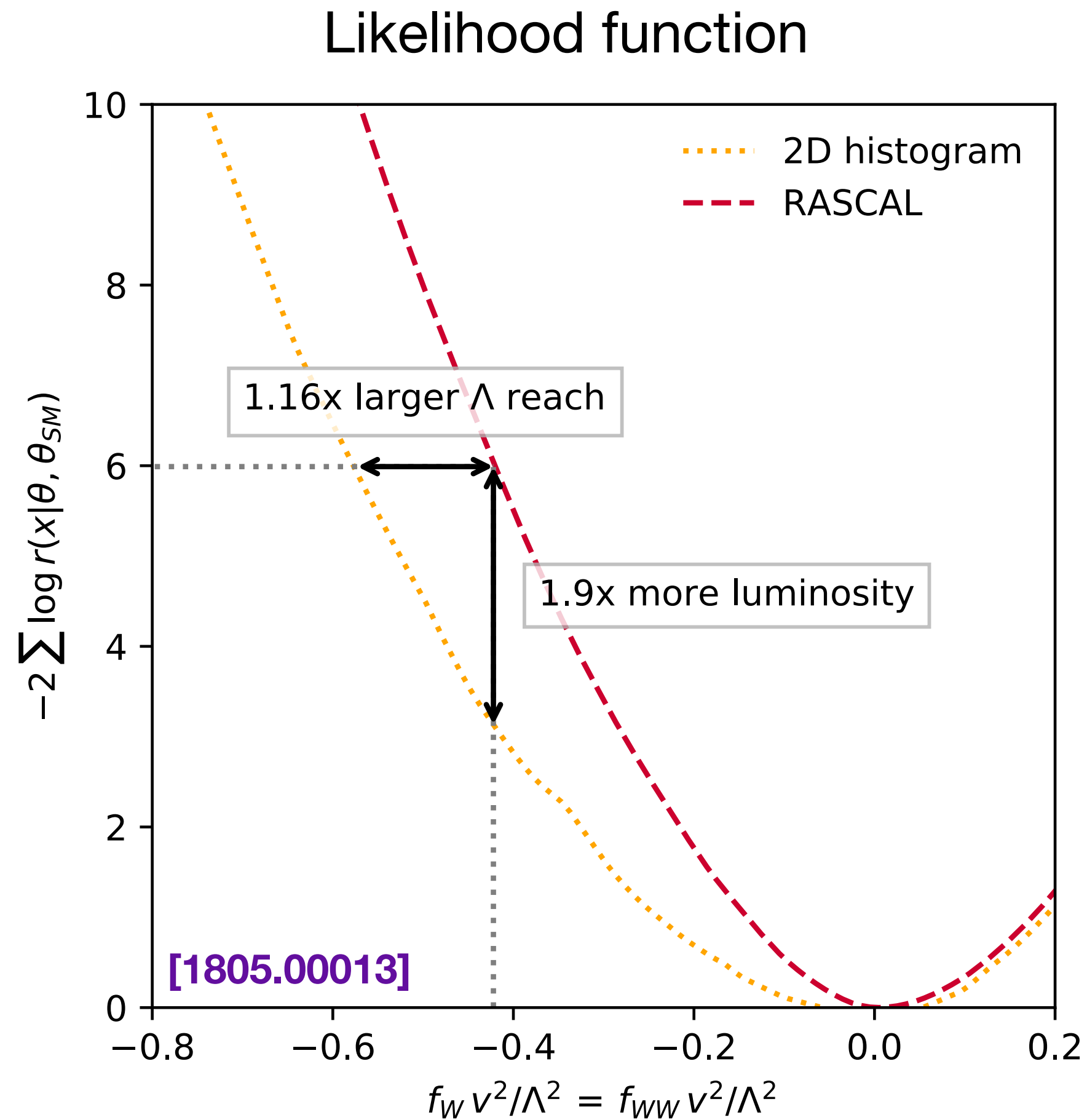
Higgs production in weak boson fusion:



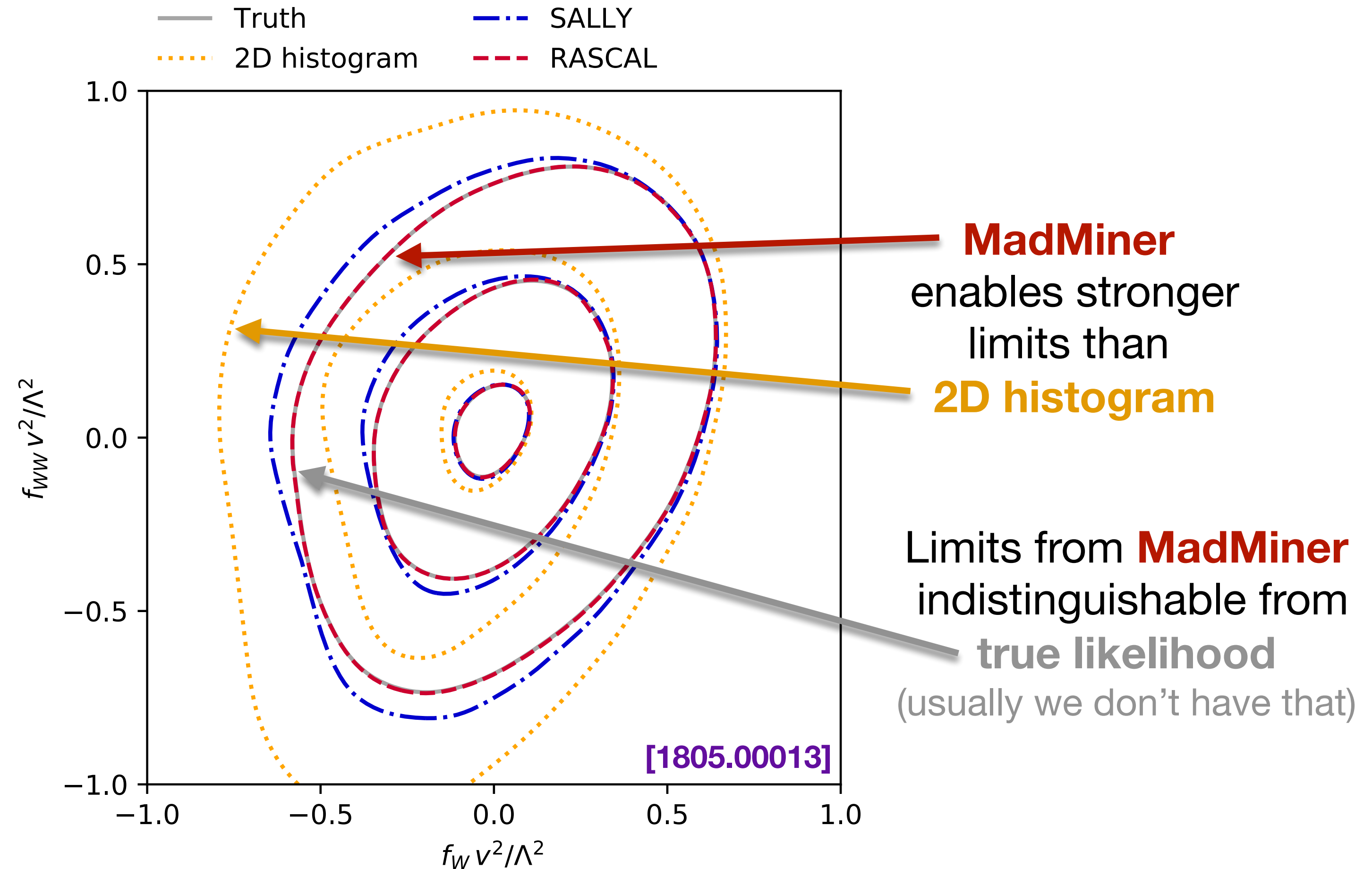
Exciting new physics might hide here!
We parameterize it with two EFT coefficients

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

Better sensitivity to new physics



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



Results are based on 36 observed events, assuming SM

Appendix

Bonus material

Variational calculus

$$\begin{aligned} L[\hat{g}(x)] &= \int dx \int dz p(x, z | \theta) [g(x, z) - \hat{g}(x)]^2 \\ &= \int dx \left[\hat{g}^2(x) \int dz p(x, z | \theta) - 2\hat{g}(x) \int dz p(x, z | \theta) g(x, z) + \int dz p(x, z | \theta) g^2(x, z) \right] \\ &\qquad\qquad\qquad = F(x) \end{aligned}$$

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

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

Choose

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