26TH International Conference on Computing in High Energy & Nuclear Physics (CHEP2023)
May, 8-12 2023

Unbiased detection of data departures from expectations with machine learning

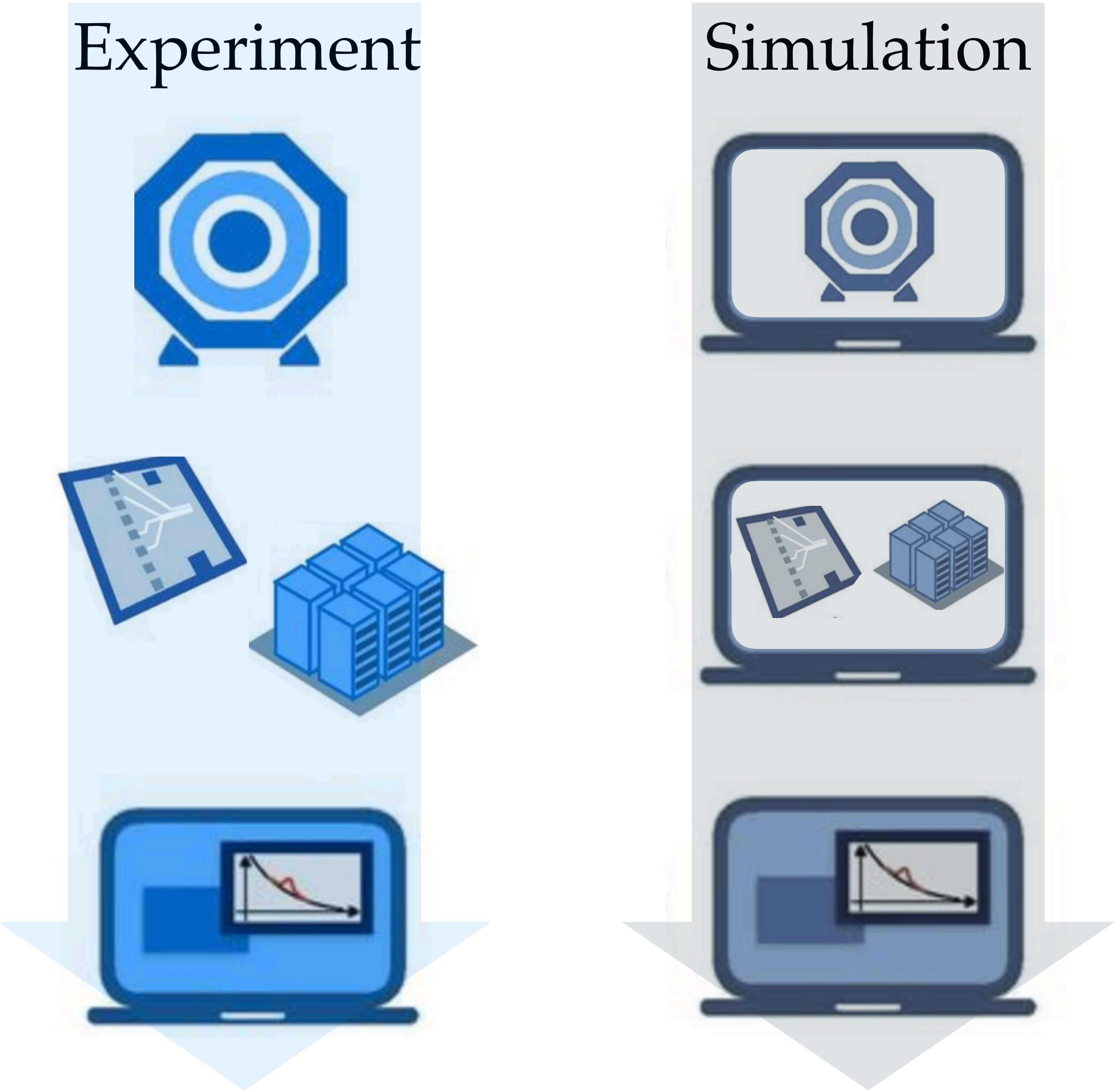
G. Grosso[1], M. Letizia, A. Wulzer, M. Zanetti
[1]University and INFN of Padova

Goodness of fit in High Energy Physics

Detector malfunctioning ?

Reconstruction Bugs ?

Unexpected Physics Beyond the Standard Model ?

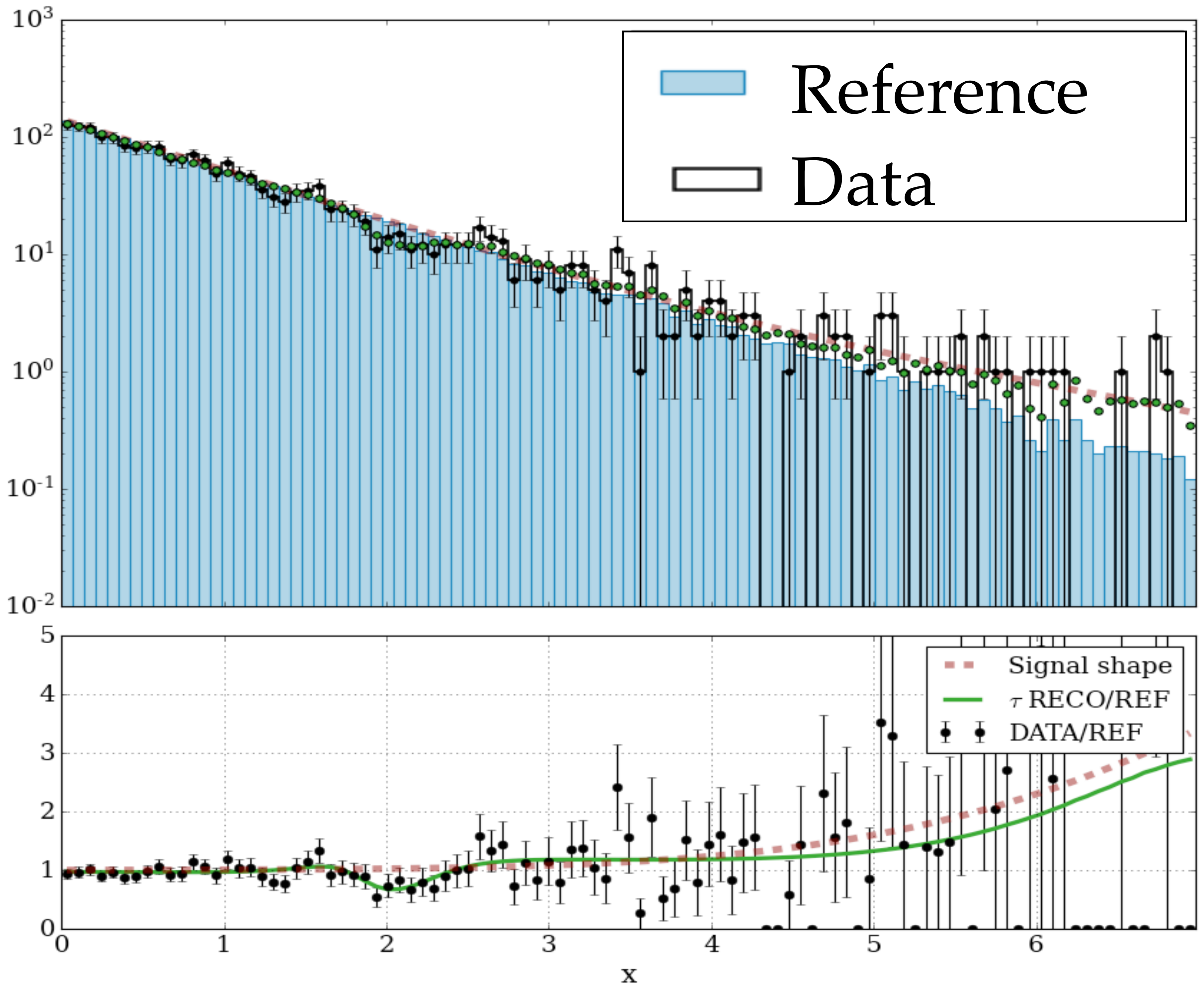


Can we trust our simulations ?

Are two simulators equivalent ?

Goodness of fit in High Energy Physics

These are problems of **Goodness of Fit**:



- **Data**: experimental measurements of the natural process
- **Reference model**: expected nominal behaviour of the data

(Standard Model, normal operating condition of a detector…).

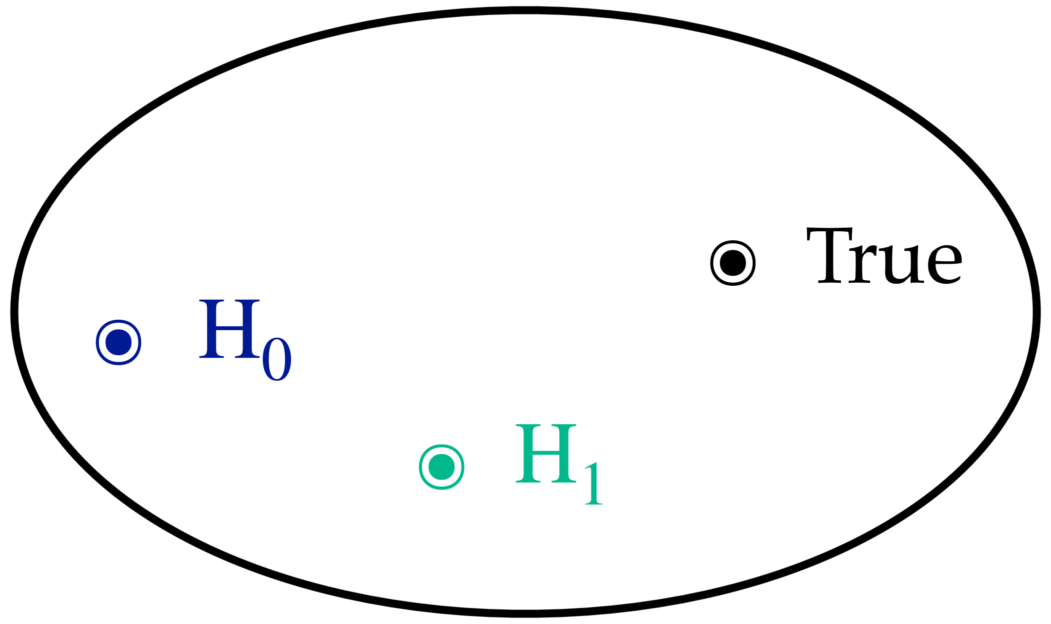
Most of the times not known in close form: Reference sample → two-sample test

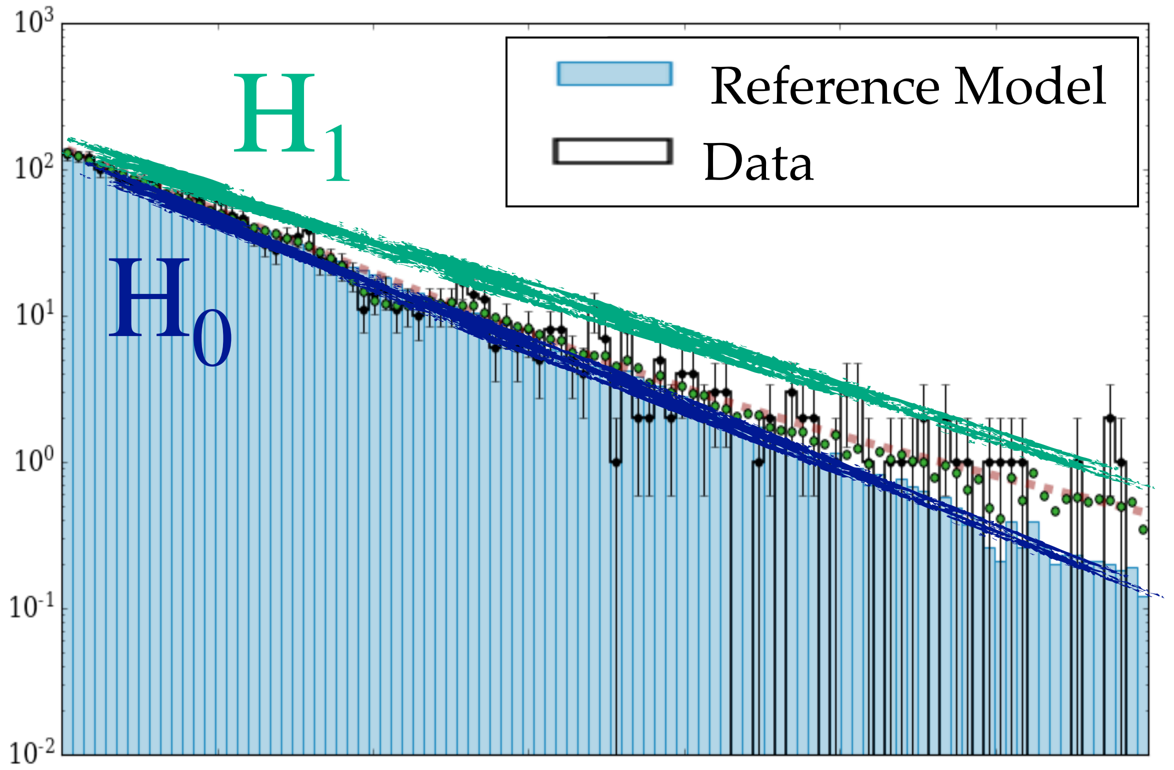
How well does the model fit the data?

Likelihood ratio test in High Energy Physics

…traditionally solved in HEP as a hypothesis test based on likelihoods ratio

→ Model-dependent approaches



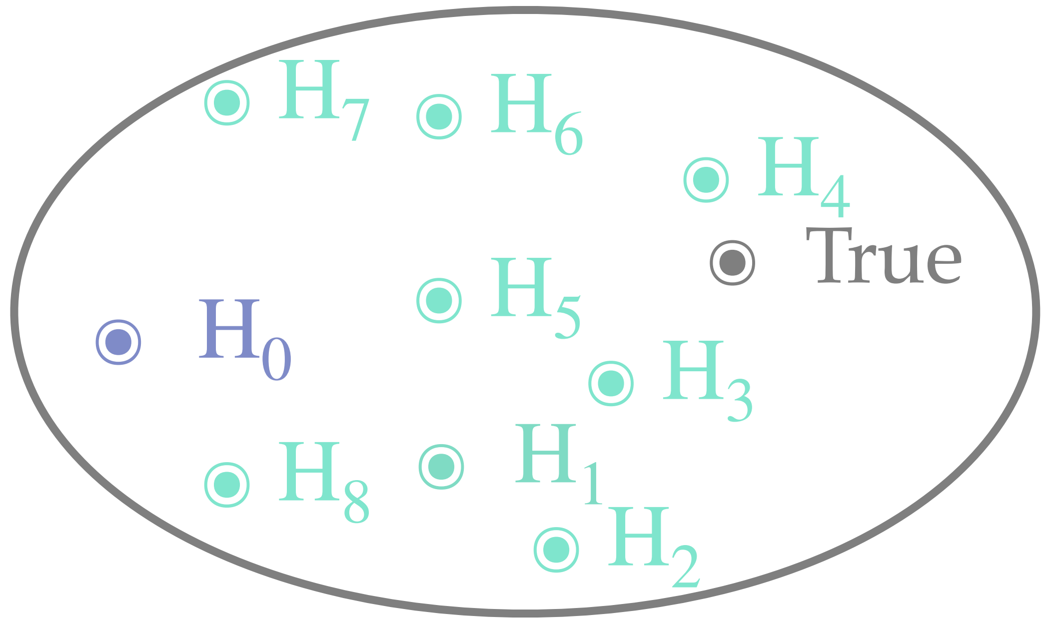


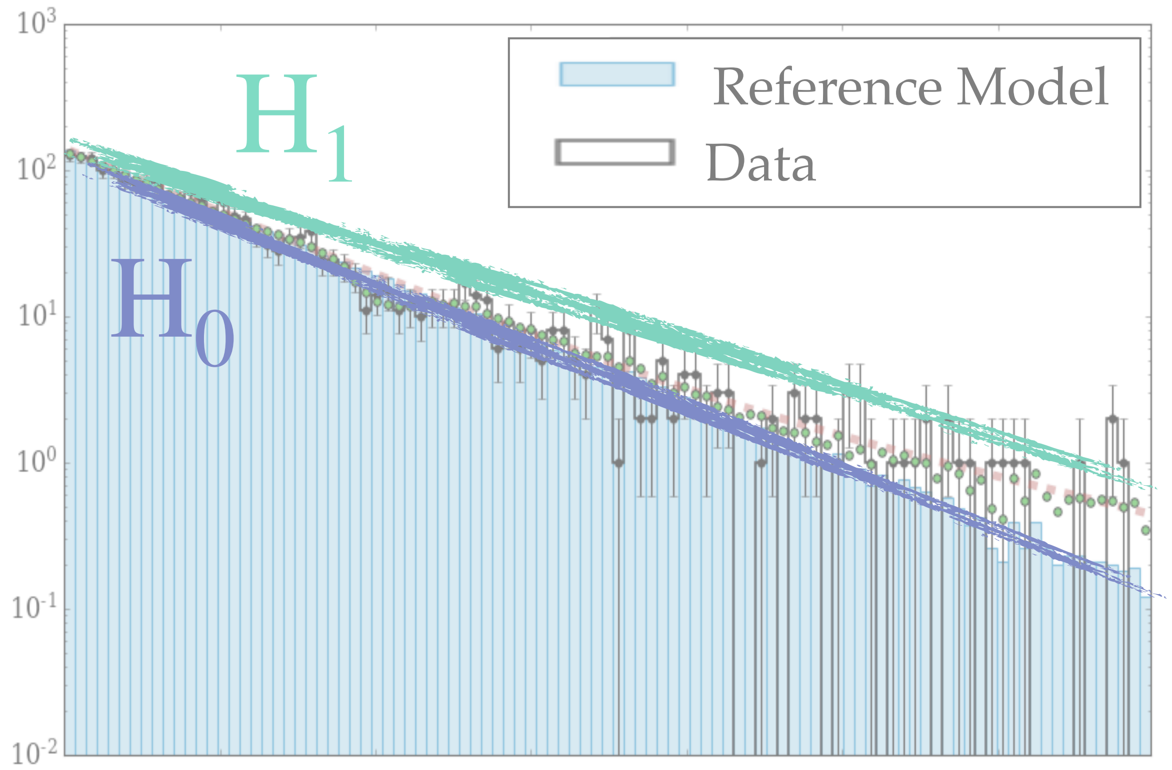
$$t(\mathcal{D}) = 2 \log \frac{\mathcal{L}(\mathcal{D} \,|\mathrm{H}_1)}{\mathcal{L}(\mathcal{D} \,|\mathrm{H}_0)}$$

Sensitivity (and optimality) guaranteed (according to Neyman and Pearson) only <u>if</u> the the data do follow the chosen alternative

Likelihood ratio test in High Energy Physics

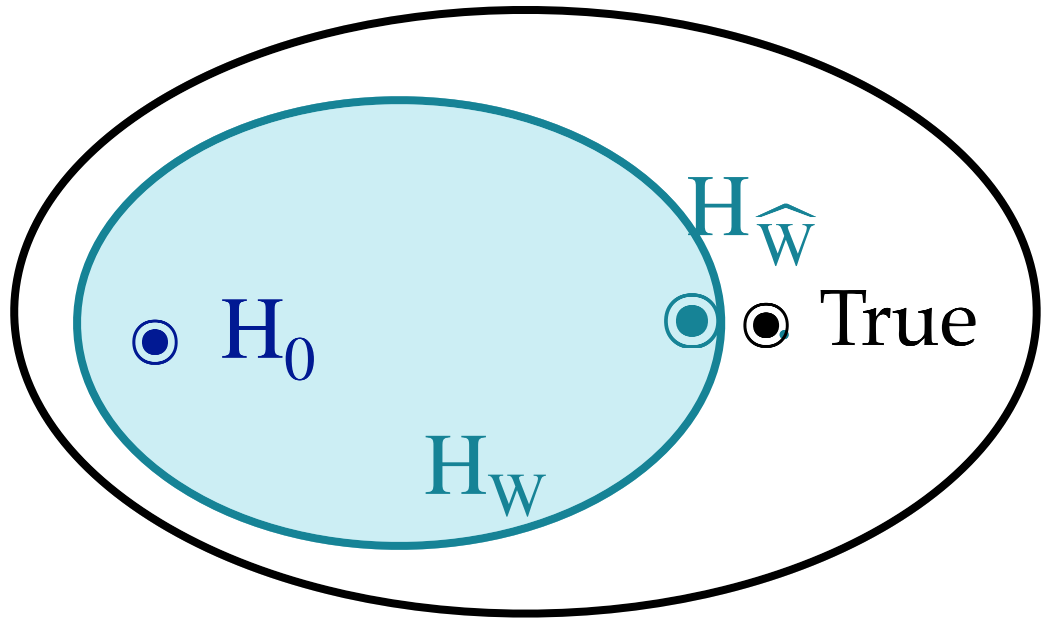
→ Model-dependent approaches

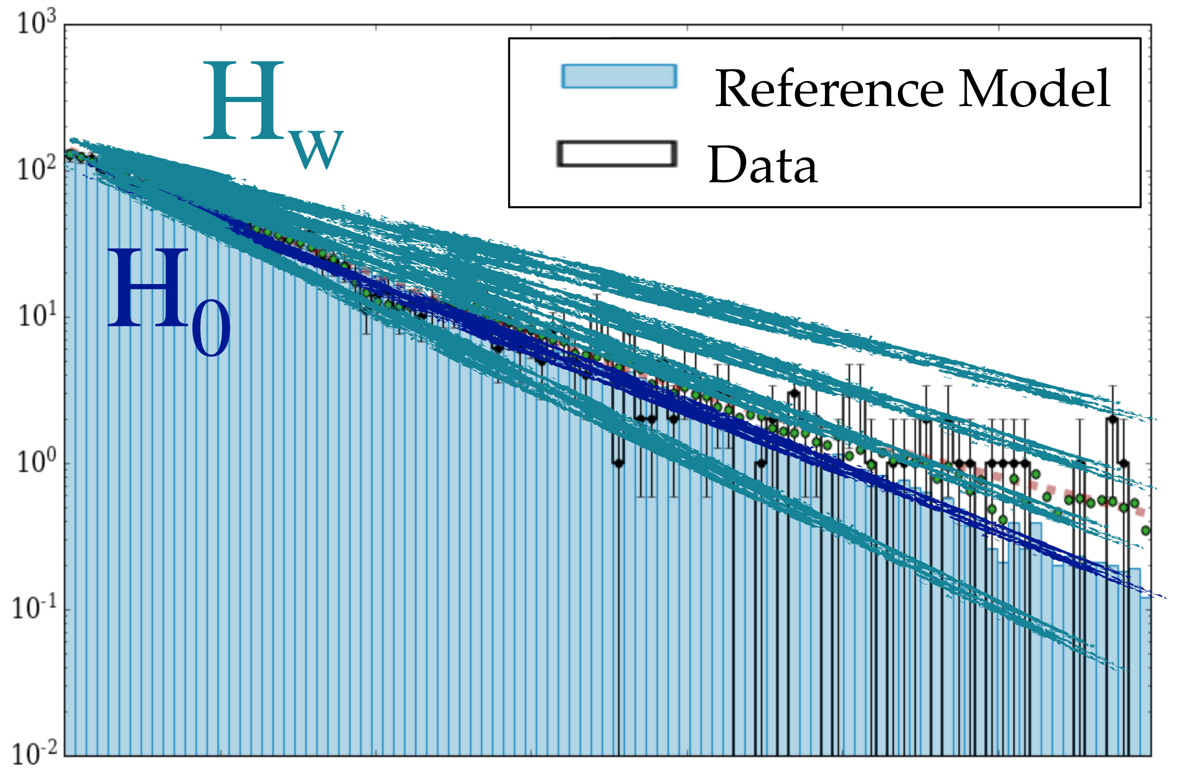




$$t(\mathcal{D}) = 2\,\log \frac{\mathcal{L}(\mathcal{D}\,|\mathrm{H}_1)}{\mathcal{L}(\mathcal{D}\,|\mathrm{H}_0)}$$

→ Model-*in*dependent approaches





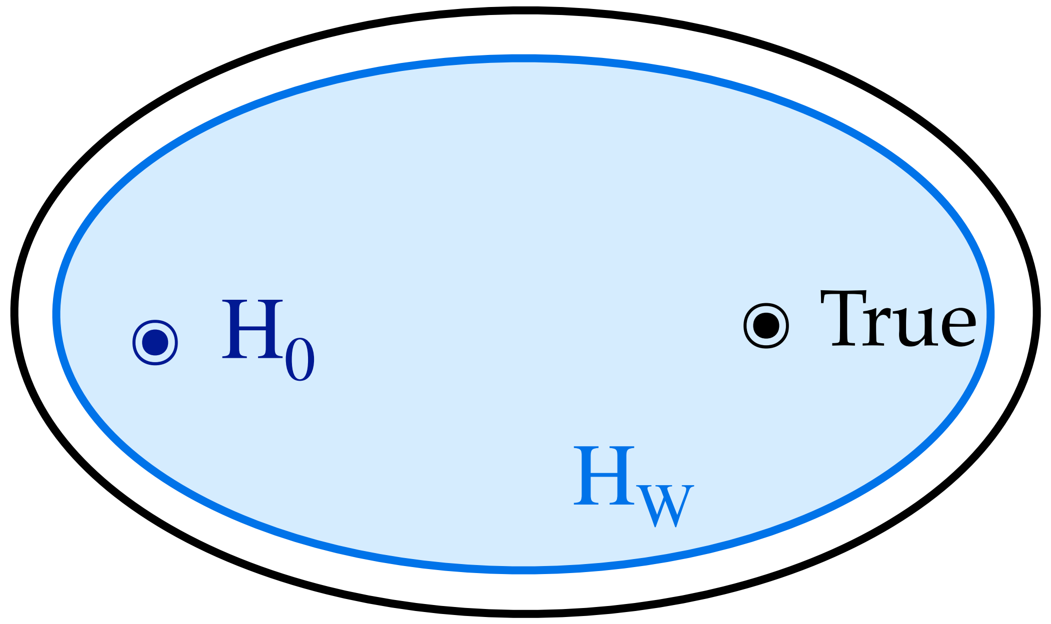
$$t(\mathcal{D}) = \max_{\mathbf{w}} \left[2\,\log \frac{\mathcal{L}(\mathcal{D}\,|\mathrm{H}_{\mathbf{w}})}{\mathcal{L}(\mathcal{D}\,|\mathrm{H}_0)} \right]$$

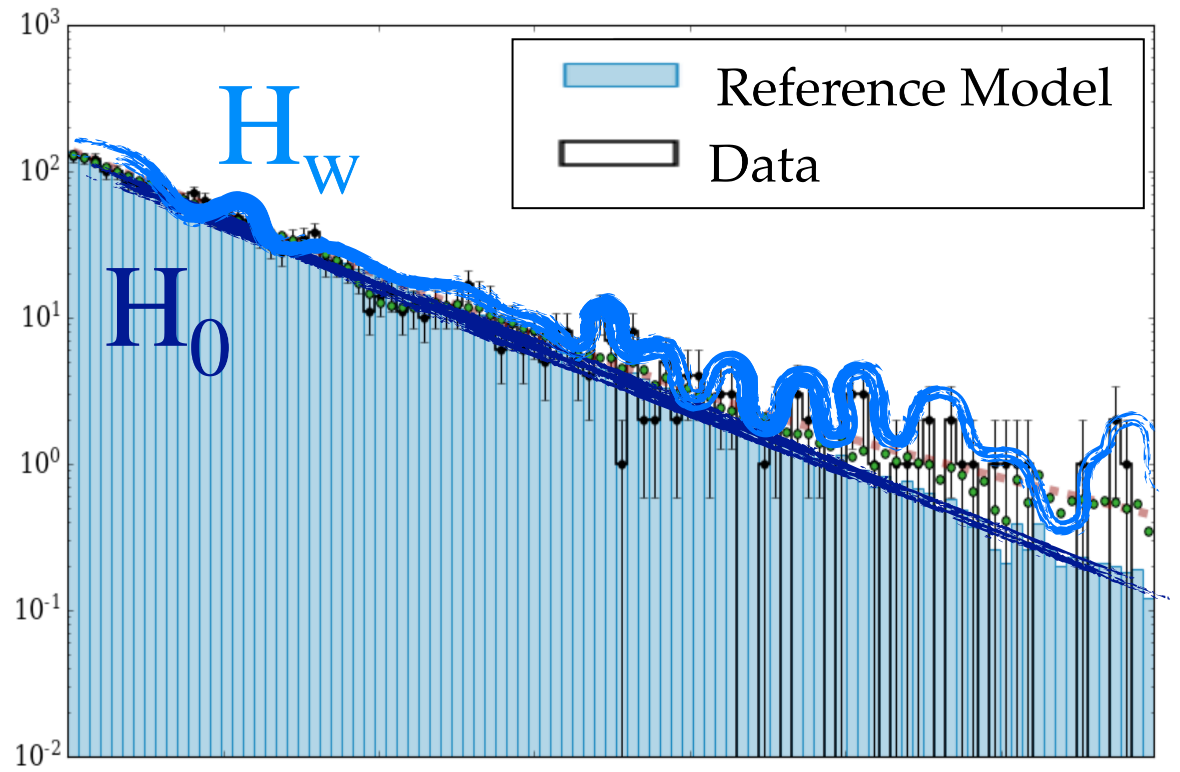
Likelihood ratio as a Goodness of fit test

Machine learning based

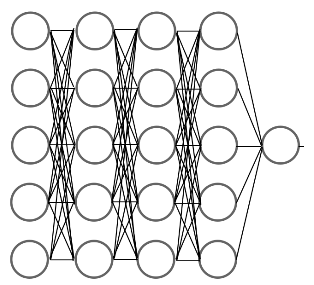
Expand the family of alternatives to increase the chance of containing the True data distribution

→ Model-*in*dependent approaches





New Physics Learning Machine



Universal approximator (NN, kernel methods, …)

$$n(x|H_{\mathbf{w}}) = e^{f(x;\mathbf{w})}\, n(x|R_0)$$

$$t(\mathcal{D}) = \max_{\mathbf{w}} \left[2 \log \frac{\mathcal{L}(\mathcal{D}\,|H_{\mathbf{w}})}{\mathcal{L}(\mathcal{D}\,|H_0)} \right]$$

New Physics Learning Machine (NPLM)

Main Idea: Maximum Likelihood from Minimal Loss

Test statistic

$$\bar{t}(\mathcal{D}) = 2\max_{\mathbf{w}} \log\left[\frac{\mathcal{L}(\mathcal{D}\,|\,\mathrm{H}_{\mathbf{w}})}{\mathcal{L}(\mathcal{D}\,|\,\mathrm{R}_0)}\right] = 2\max_{\mathbf{w}}\left\{\log\left[\frac{e^{-N(\mathbf{w})}}{e^{-N(\mathrm{R})}}\prod_{i=1}^{N_{\mathcal{D}}}\frac{n(x_i|\mathbf{w})}{n(x_i|\mathrm{R})}\right]\right\}$$

$$= -2\min_{\mathbf{w}}\{\bar{L}[f(\cdot;\,\mathbf{w})]\}$$

Loss function

$$\bar{L}\,[f(x;\,\mathbf{w})] = -\sum_{x\in\mathcal{D}}[f(x;\,\mathbf{w})] + \sum_{x\in\mathcal{R}} w_x\left[e^{f(x;\,\mathbf{w})} - 1\right]$$

w: trainable parameters on the NN model

D: data sample

R: reference sample (built according to the R_0 hypothesis); could be weighted (w)

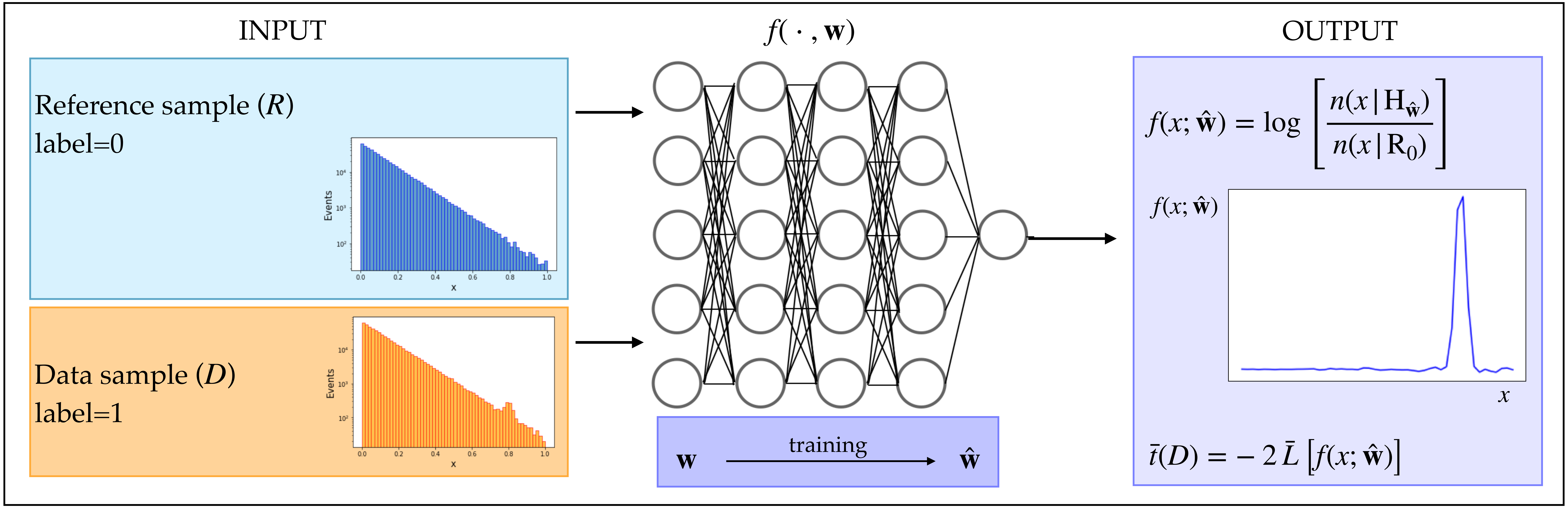
Assumptions:

- $N_R \gg N_D$ the statistical fluctuations of the reference sample are negligible.
- the weights of the reference sample (w) are such that the reference sample is normalised to match the data sample luminosity $\sum_{x\in R} w_x = N(R_0)$

"Learning New Physics from a Machine" Phys. Rev. D

New Physics Learning Machine (NPLM)

Learning the alternative from the data:



Unbinned input data

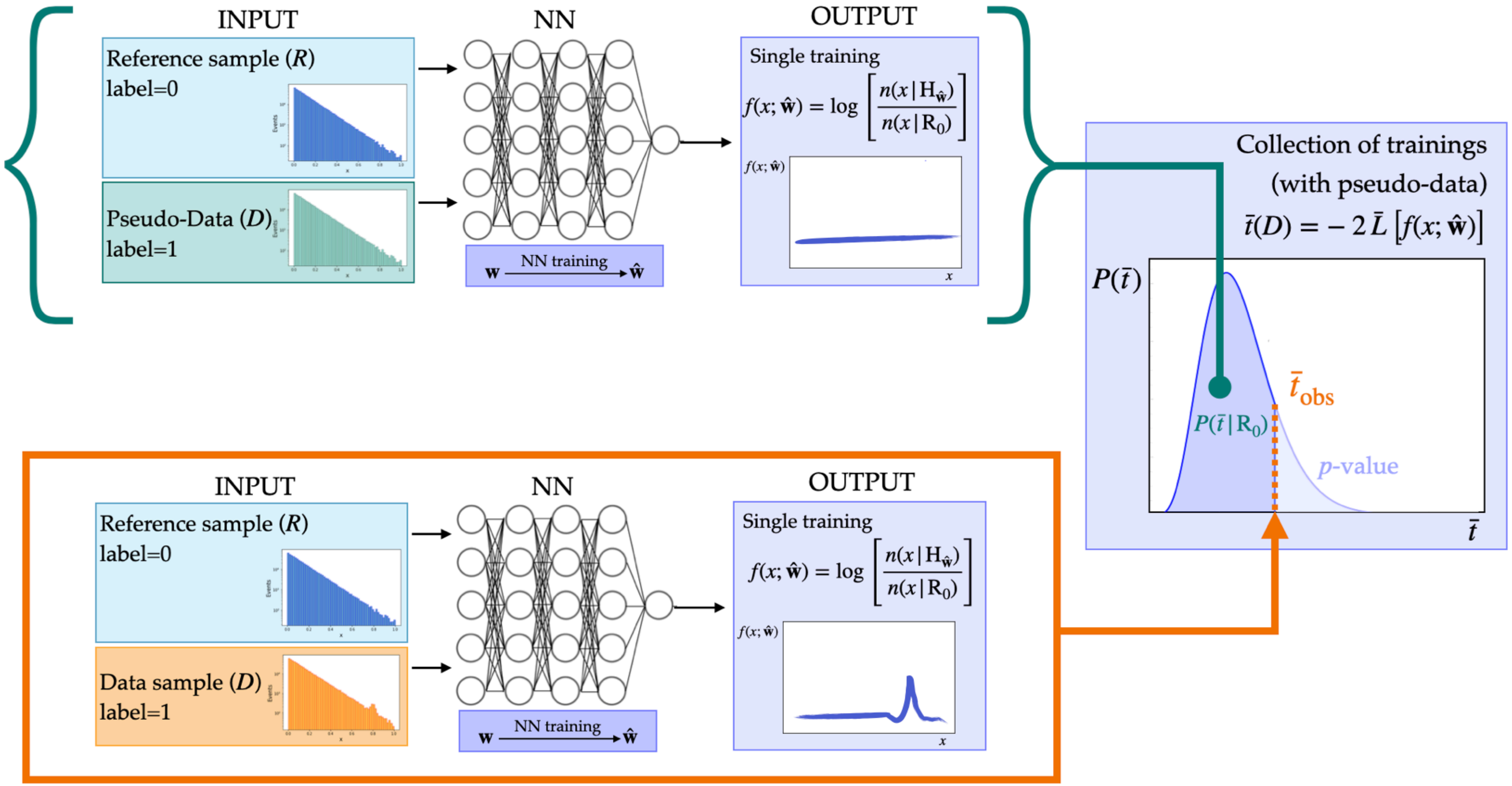
Interpretability
Where is the anomaly and how does it look like

New Physics Learning Machine (NPLM)

Frequentist *p*-value (aka calibration):

1. Run NPLM on toy experiments to simulated the response under the null hypothesis

2. Run NPLM the data of interest and check where the test outcome falls to compute an exclusion *p*-value

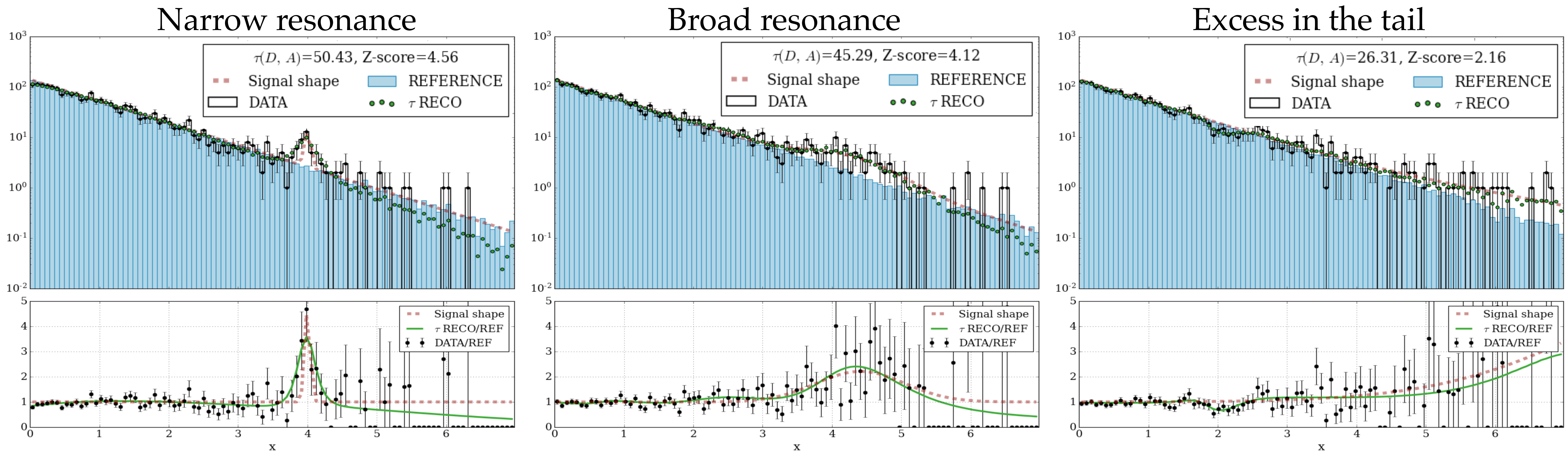


"Learning New Physics from a Machine" Phys. Rev. D

New Physics Learning Machine (NPLM)

Sensitivity to multiple discrepancy sources at once (**global** p-value)

Example: 1D exponential distribution



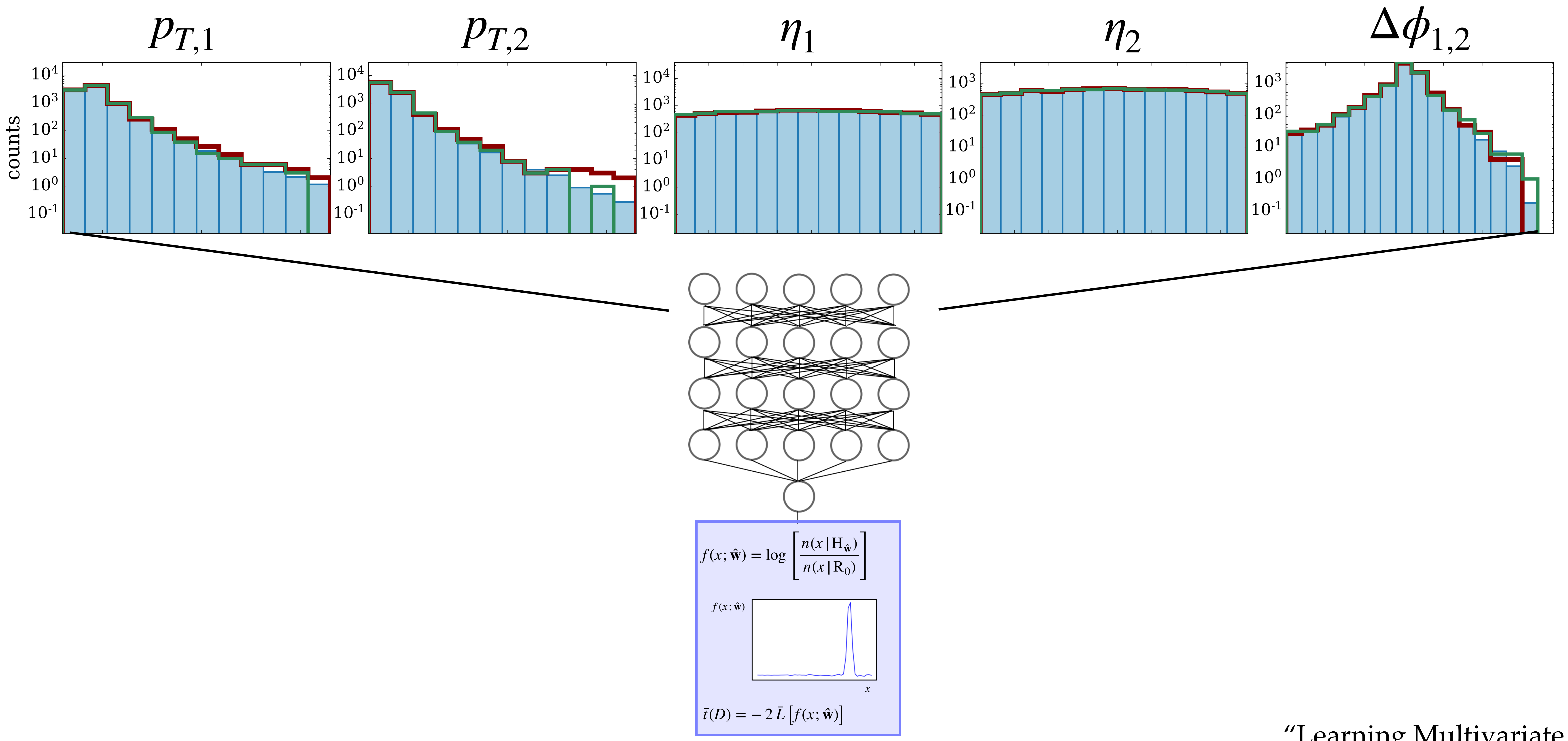
Signal reconstruction with the NN: $n(x\,|\,\mathrm{H}_{\hat{\mathrm{w}}}) = n(x\,|\,\mathrm{R}_0)\; e^{f(x;\,\hat{\mathrm{w}})}$

Architecture: [1-4-1] (13 dof), weigh clipping 9, $\mathrm{N}_{\mathrm{bkg}}$=2000

New Physics Learning Machine (NPLM)

Multivariate analysis

Example: 5D analysis of a dimuon final state at the LHC



"Learning Multivariate New Physics" Eur. Phys. J. C

New Physics Learning Machine (NPLM)

Dealing with **imperfect** Reference models

Maximum-Likelihood-ratio test statistic:

$$t(\mathcal{D},\,\mathcal{A}) = 2\log\left[\frac{\max\limits_{\mathbf{w},\,\boldsymbol{\nu}}\mathcal{L}(\mathrm{H}_{\mathbf{w},\,\boldsymbol{\nu}}|\mathcal{D},\,\mathcal{A})}{\max\limits_{\boldsymbol{\nu}}\mathcal{L}(\mathrm{R}_{\boldsymbol{\nu}}|\mathcal{D},\,\mathcal{A})}\right] = 2\log\left[\frac{\max\limits_{\mathbf{w},\,\nu}\mathcal{L}(\mathrm{H}_{\mathbf{w},\,\boldsymbol{\nu}}|\mathcal{D})\mathcal{L}(\boldsymbol{\nu}|\mathcal{A})}{\max\limits_{\boldsymbol{\nu}}\mathcal{L}(\mathrm{R}_{\boldsymbol{\nu}}|\mathcal{D})\mathcal{L}(\boldsymbol{\nu}|\mathcal{A})}\right]$$

$$= \tau(\mathcal{D},\,\mathcal{A}) - \Delta(\mathcal{D},\,\mathcal{A})$$

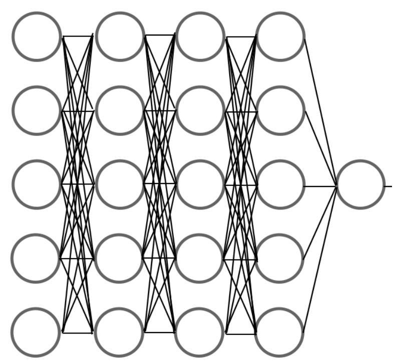
- $\mathrm{R}_{\boldsymbol{\nu}}$: reference hypothesis (null)
- $\mathrm{H}_{\mathbf{w},\,\boldsymbol{\nu}}$: alternative hypothesis
- $\mathbf{w}$: trainable parameters on the NN model
- $\boldsymbol{\nu}$: set of nuisance parameters modelling the uncertainties effects
- $\mathcal{D}$: data sample
- $\mathcal{A}$: auxiliary sample (used to constrain $\boldsymbol{\nu}$)

Parametrisation of the alternative hypothesis:

$$n(x\,|\,\mathrm{H}_{\hat{\mathbf{w}},\,\hat{\nu}}) = n(x\,|\,\mathrm{R}_0)\,\frac{n(x\,|\,\mathrm{R}_{\hat{\nu}})}{n(x\,|\,\mathrm{R}_0)}\,e^{f(x;\,\hat{\mathbf{w}})}$$

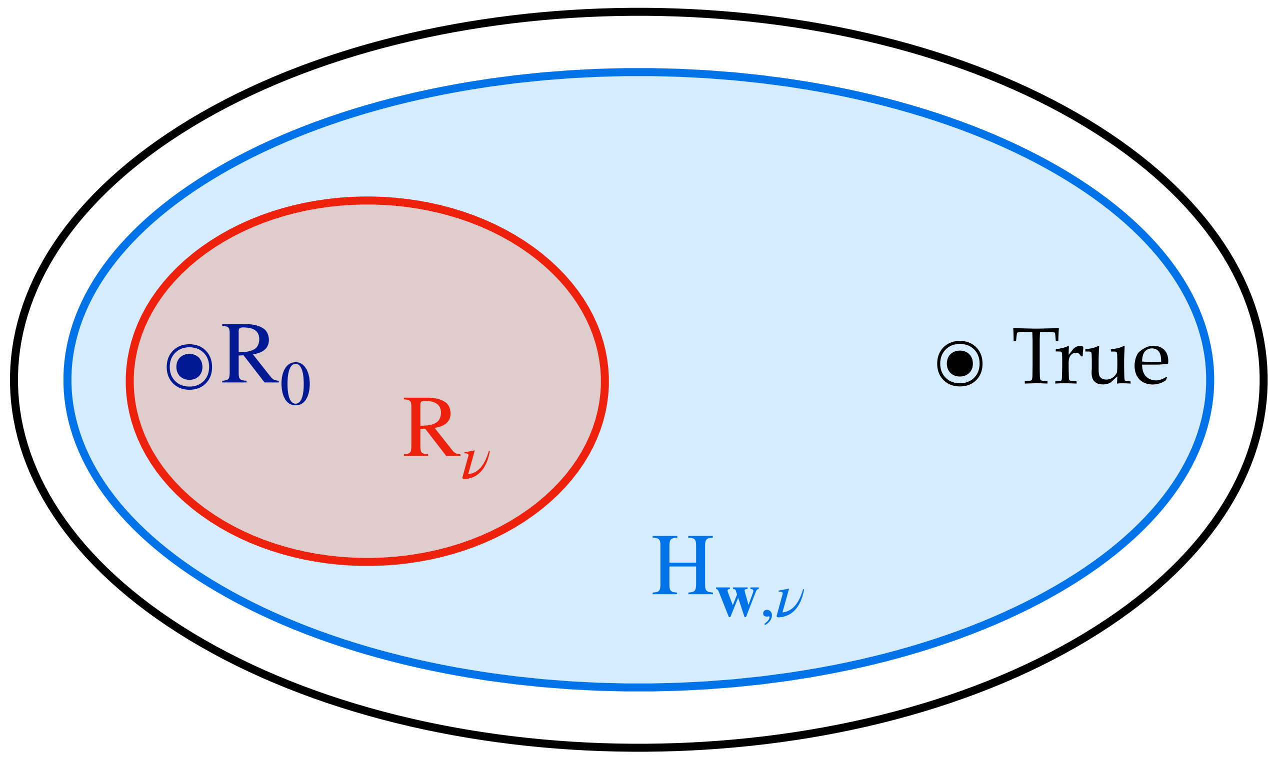
central value Reference (ν=0)

NN model



Model of systematic uncertainties effects

$$\hat{r}(x;\nu) = \frac{n(x\,|\,\mathrm{R}_\nu)}{n(x\,|\,\mathrm{R}_0)} = \exp\left[\hat{\delta}_1(x)\,\nu + \hat{\delta}_2(x)\,\nu^2 + \dots\right]$$

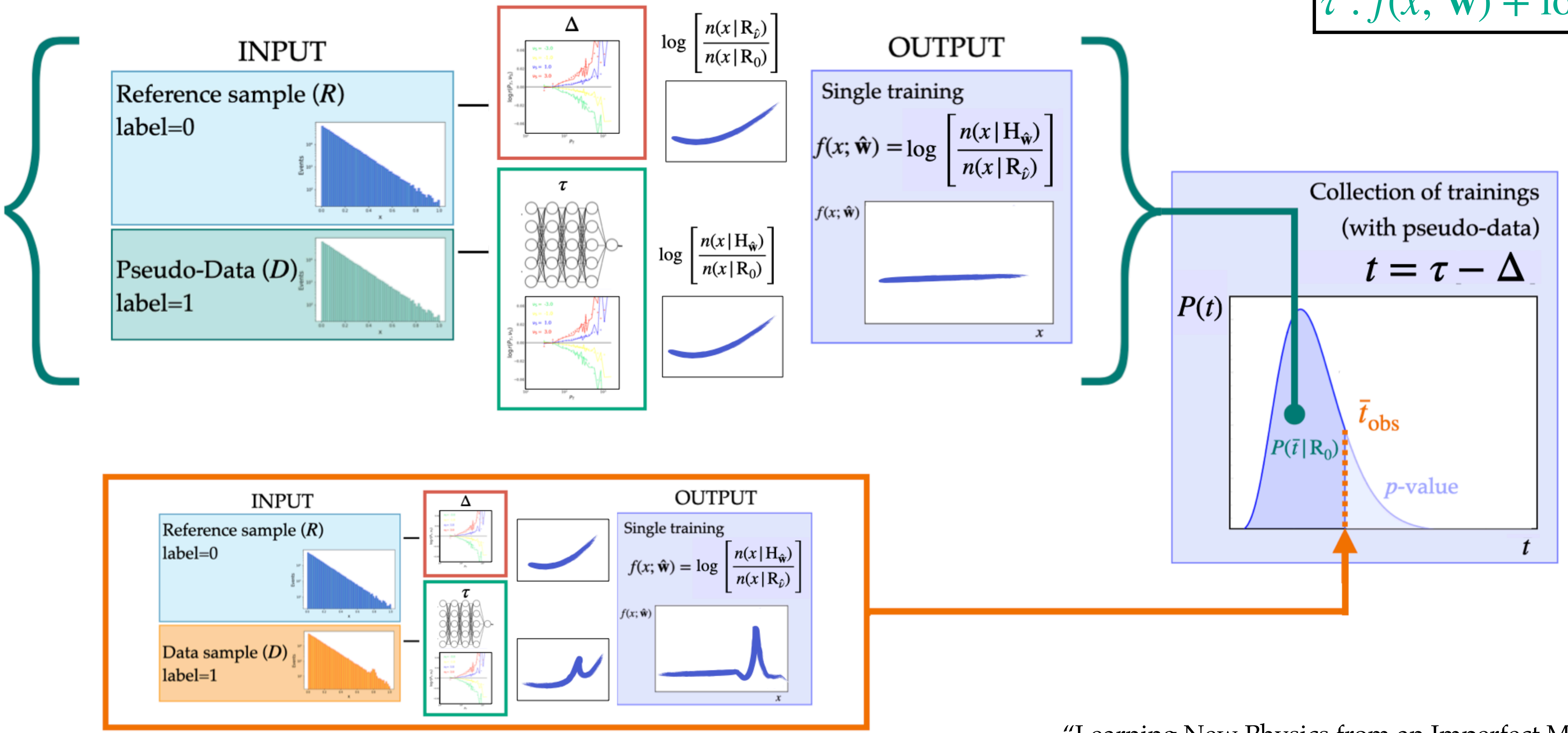


"Learning New Physics from an Imperfect Machine" Eur. Phys. J. C

New Physics Learning Machine (NPLM)

Dealing with **imperfect** Reference models

Δ : $\log r(x;\nu)$
τ : $f(x;\mathbf{w}) + \log r(x;\nu)$



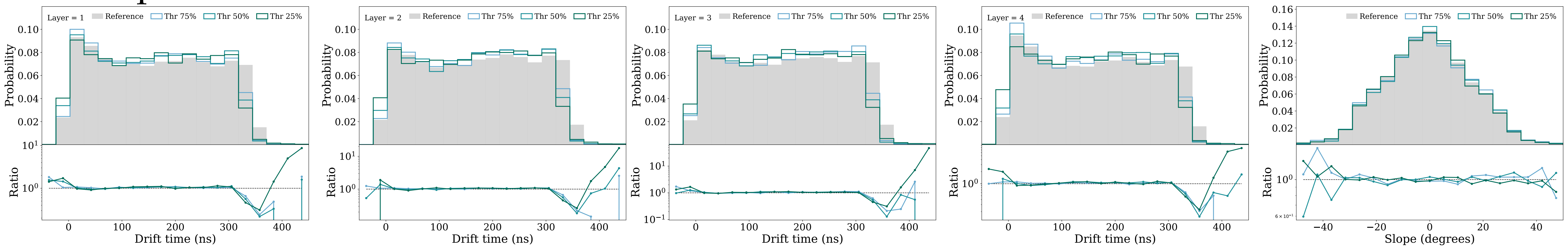
"Learning New Physics from an Imperfect Machine" Eur. Phys. J. C

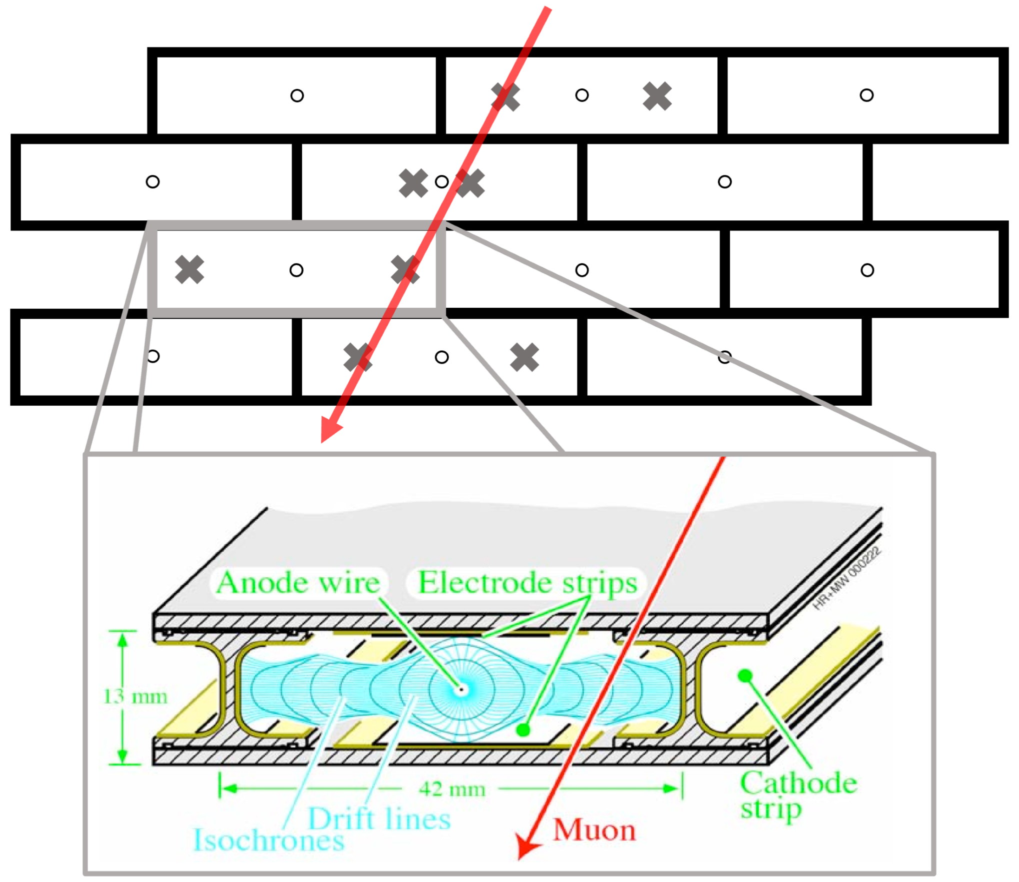
New Physics Learning Machine (NPLM)

Efficient computation: kernel methods on GPUs (Eur. Phys. J. C, 82(10))

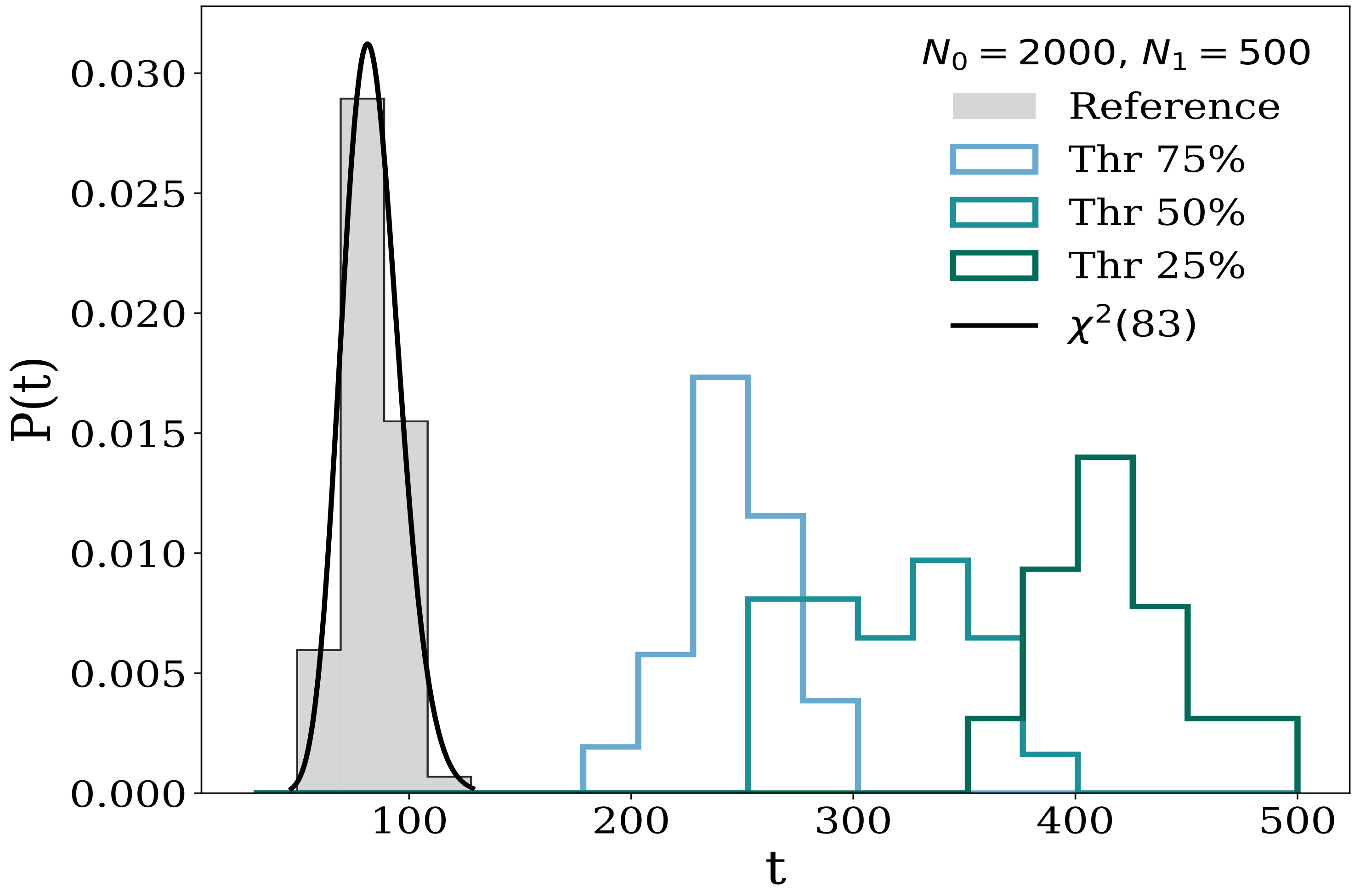
Example: Online Data Quality monitoring of a Drift Tube Chamber

5D input data





- **Reference sample:** long run in optimal conditions
- **Anomalous samples**: short runs acquired in presence of a controlled anomaly in the value of the **threshold tension** of the DT chamber



More about this in our recent preprint arXiv:2303.05413, and Matteo's talk this afternoon!

Summary

- End-to-end strategy to test a model Goodness of Fit to the data.
- Interpretability: a posteriori signal characterisation
- Global p-value: one test to detect them all
- Multivariate
- Systematic uncertainties for imperfect Reference models
- Fast execution: kernel methods with GPU

Outlook

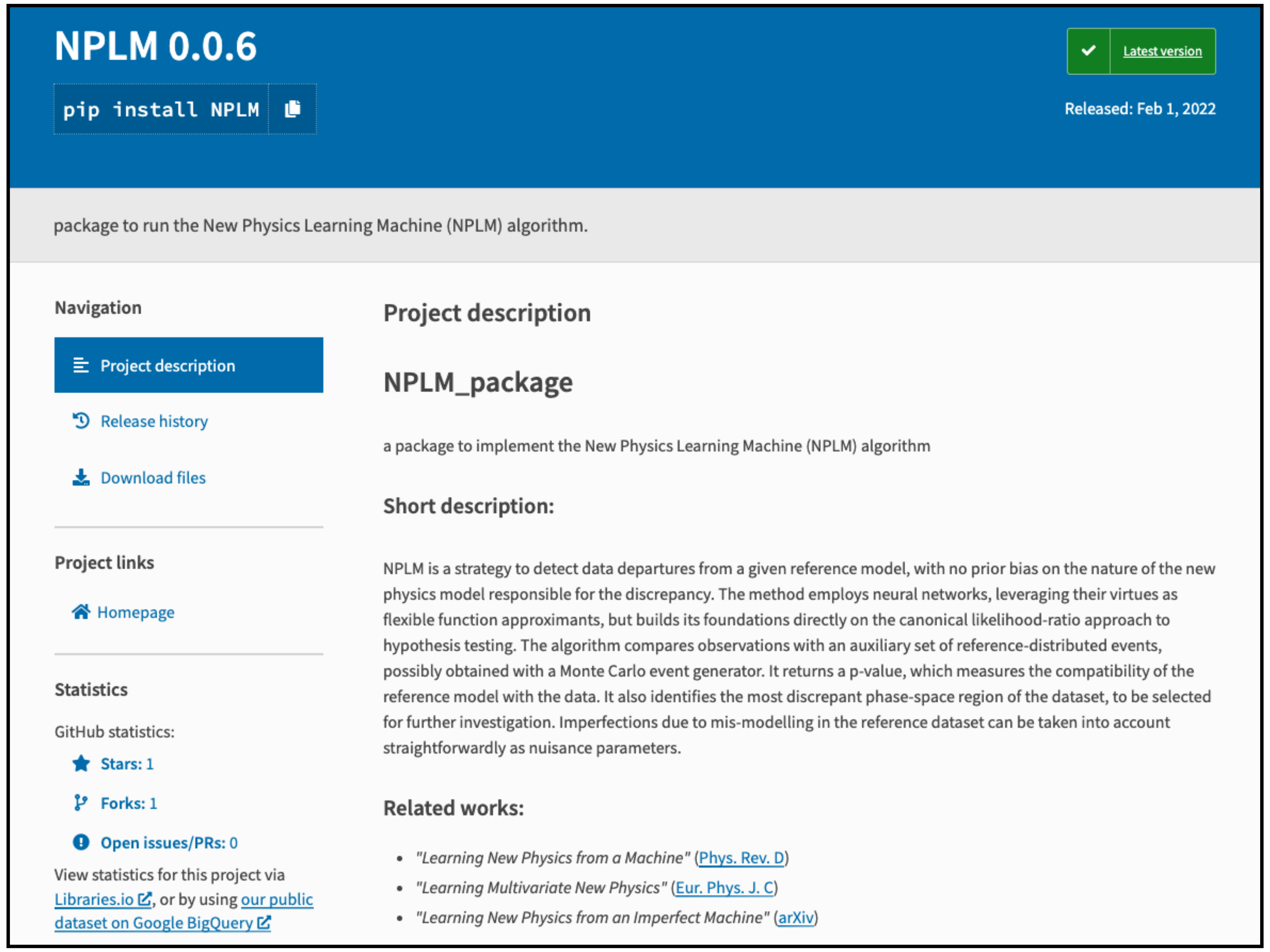
- Fast execution + systematics uncertainties
- Model and model hyper-parameters selection: regularisation vs. flexibility
- More applications in (and out) the HEP scope:
  - Model-independent BSM searches at the LHC
  - (Quasi) online data quality monitoring
  - Validation of data generators
  - ???

References

- "Learning new physics from a machine" Phys. Rev. D 99, 015014 (d'Agnolo, Wulzer)
- "Learning multivariate new physics" Eur. Phys. J. C 81, 89 (2021) (d'Agnolo, Grosso, Pierini, Wulzer, Zanetti)
- "Learning new physics from an imperfect machine" Eur. Phys. J. C 82, 275 (2022) (d'Agnolo, Grosso, Pierini, Wulzer, Zanetti)
- "Learning new physics efficiently with nonparametric methods" Eur. Phys. J. C, 82(10) (Letizia, Grosso et al.)
- "Fast kernel methods for Data Quality Monitoring as a goodness-of-fit test" arXiv:2303.05413 (Lai, Letizia, Grosso et al.)

Getting started with NPLM

- NPLM package: python-based package to run the NPLM analysis strategy
- Tutorial on 1D toy model for getting started


NPLM 0.0.6

pip install NPLM

Latest version

Released: Feb 1, 2022

package to run the New Physics Learning Machine (NPLM) algorithm.

Navigation

Project description

Release history

Download files

Project links

Homepage

Statistics

GitHub statistics:

Stars: 1

Forks: 1

Open issues/PRs: 0

View statistics for this project via Libraries.io, or by using our public dataset on Google BigQuery

Project description

NPLM_package

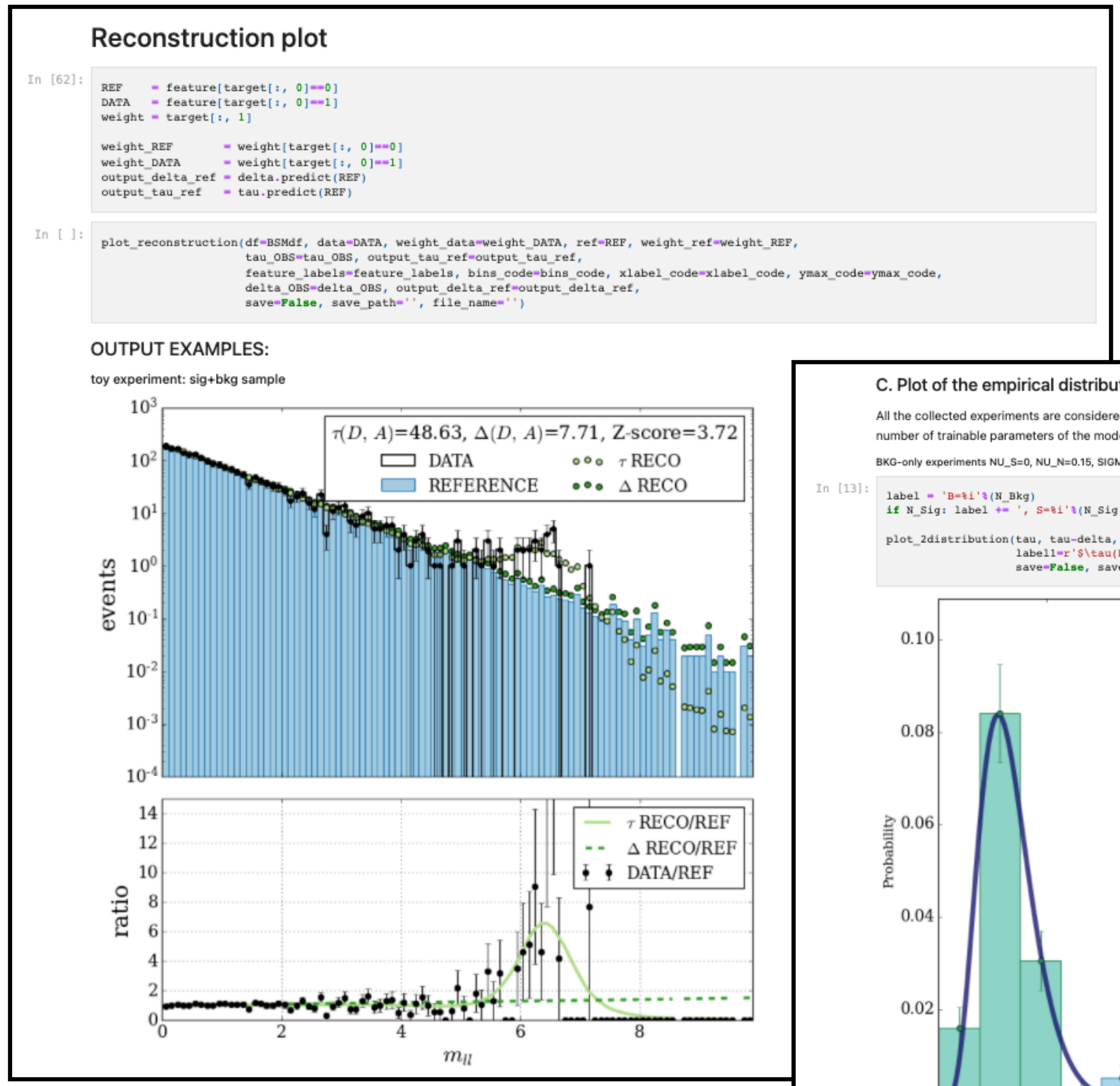
a package to implement the New Physics Learning Machine (NPLM) algorithm

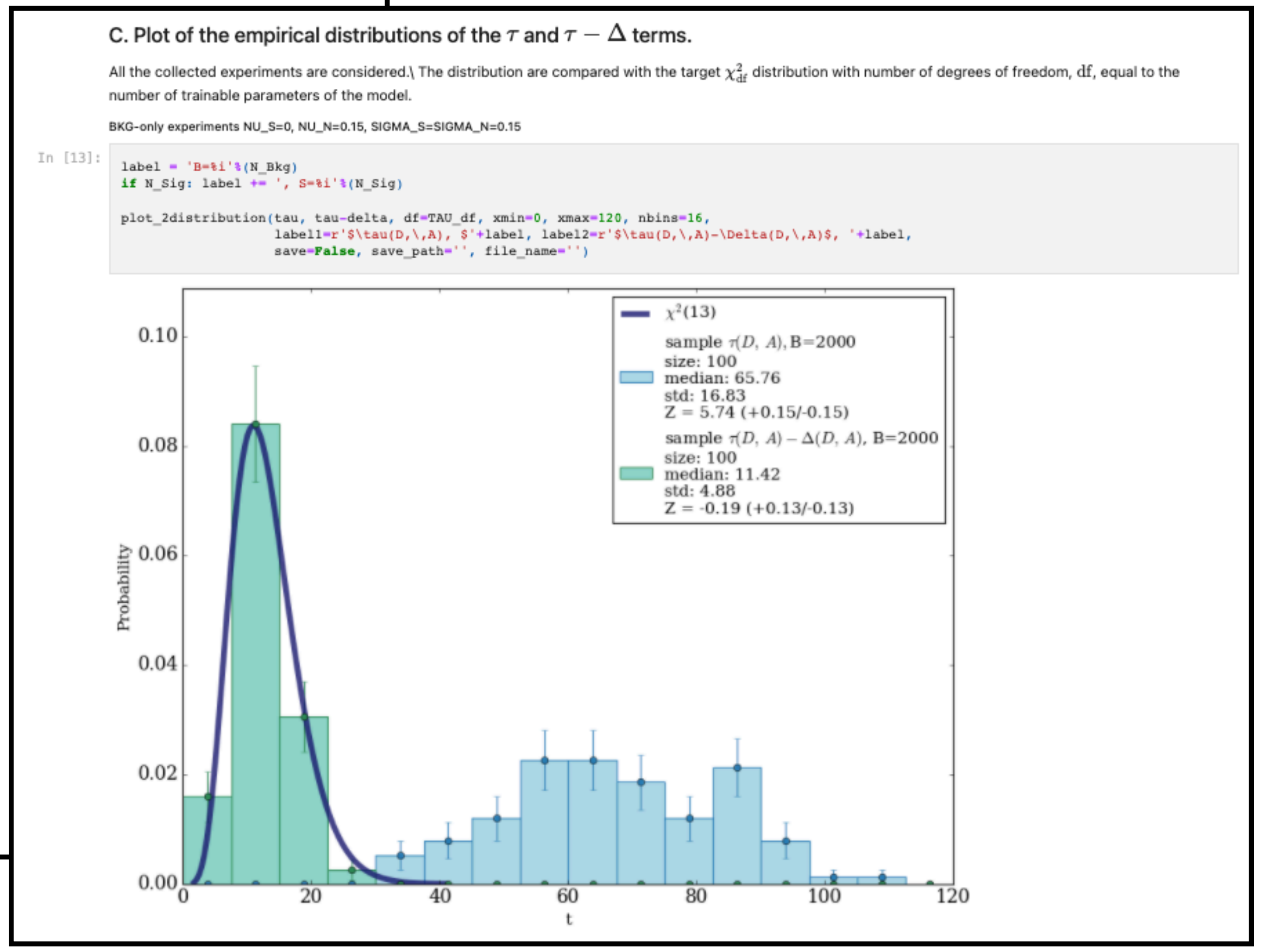
Short description:

NPLM is a strategy to detect data departures from a given reference model, with no prior bias on the nature of the new physics model responsible for the discrepancy. The method employs neural networks, leveraging their virtues as flexible function approximants, but builds its foundations directly on the canonical likelihood-ratio approach to hypothesis testing. The algorithm compares observations with an auxiliary set of reference-distributed events, possibly obtained with a Monte Carlo event generator. It returns a p-value, which measures the compatibility of the reference model with the data. It also identifies the most discrepant phase-space region of the dataset, to be selected for further investigation. Imperfections due to mis-modelling in the reference dataset can be taken into account straightforwardly as nuisance parameters.

Related works:

- *"Learning New Physics from a Machine"* (Phys. Rev. D)
- *"Learning Multivariate New Physics"* (Eur. Phys. J. C)
- *"Learning New Physics from an Imperfect Machine"* (arXiv)





Backup slides

New Physics Learning Machine (NPLM)

Controlling type I errors: NN model regularisation

Due to the **finite size** of the training samples, the **sparsity** of the data (especially in multivariate problems) and the **ill-definition of the loss** (unbounded from below), the distribution of $t(D)$ under R_0 doesn't converge to a stable configuration naturally (hope for the emergence of a χ^2).

→ a **(NN) MODEL REGULARIZATION** procedure can solve this problem!

The regularisation enforces a level of *smoothness* in the model, preventing from "overfitting" sparse data points in the sample.

New Physics Learning Machine (NPLM)

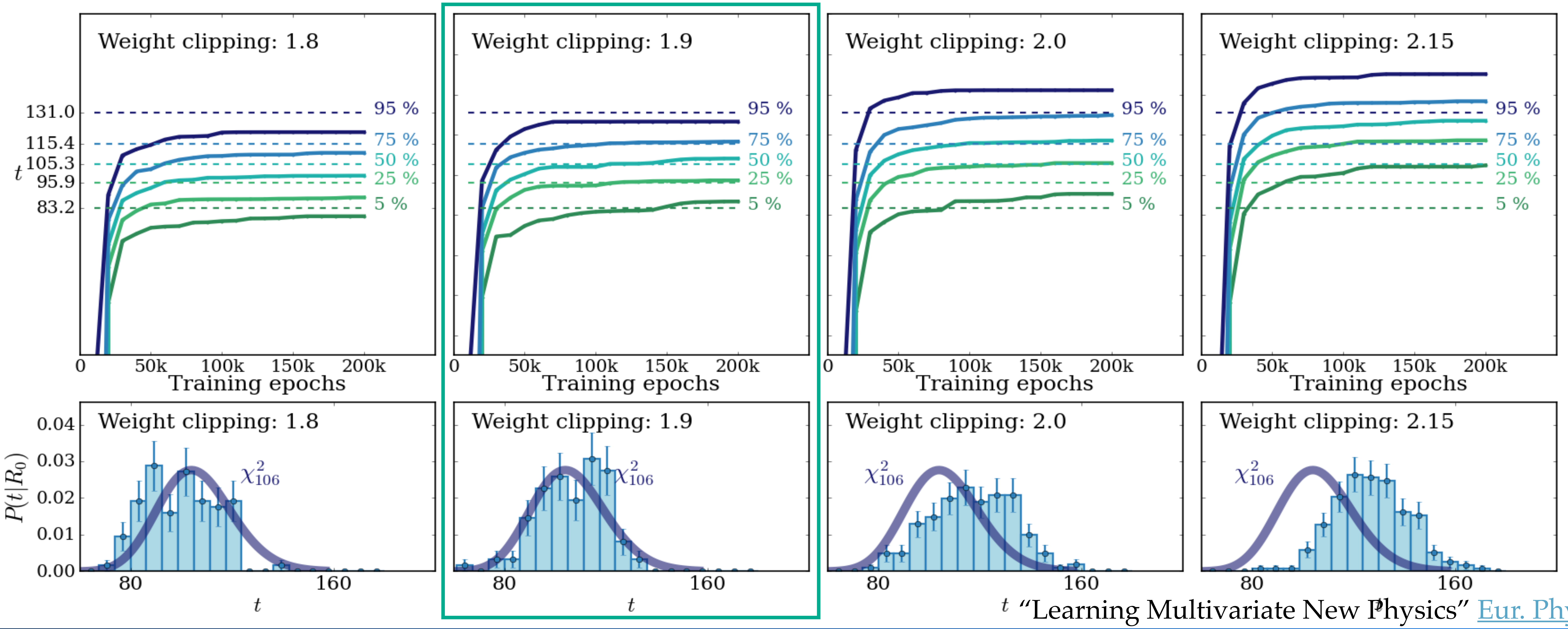
Controlling type I errors: NN model regularisation

Weight clipping parameter:
Upper boundary to the magnitude that each trainable parameter can assume during the training.

For a chosen NN architecture, tuning the weight clipping allows to recover a good agreement of the empirical distribution of t under R_0 with a **target** $\chi^2_{|\mathbf{w}|}$ distribution.

Example:
NN model: 5-7-7-1,
Number of parameters:106

Legend:
- Percentiles of the empirical $\bar{t}$ distribution under R_0
- Percentiles of the target $\chi^2_{|\mathbf{w}|}$
- Empirical $\bar{t}$ distribution under R_0
- Target $\chi^2_{|\mathbf{w}|}$



"Learning Multivariate New Physics" Eur. Phys. J. C

New Physics Learning Machine (NPLM)

Maximum-Likelihood-ratio test statistic:

$$t(\mathcal{D},\,\mathcal{A}) = 2\log\left[\frac{\max_{\mathbf{w},\,\boldsymbol{\nu}} \mathcal{L}(\mathrm{H}_{\mathbf{w},\,\boldsymbol{\nu}}|\mathcal{D})\mathcal{L}(\boldsymbol{\nu}|\mathcal{A})}{\max_{\boldsymbol{\nu}} \mathcal{L}(\mathrm{R}_{\boldsymbol{\nu}}|\mathcal{D})\mathcal{L}(\boldsymbol{\nu}|\mathcal{A})}\right]\cdot\frac{\mathcal{L}(\mathrm{R}_{\mathbf{0}}|\mathcal{D})\mathcal{L}(\mathbf{0}|\mathcal{A})}{\mathcal{L}(\mathrm{R}_{\mathbf{0}}|\mathcal{D})\mathcal{L}(\mathbf{0}|\mathcal{A})}$$

$$= \tau(\mathcal{D},\,\mathcal{A}) - \Delta(\mathcal{D},\,\mathcal{A})$$

- $\mathrm{R}_{\boldsymbol{\nu}}$: reference hypothesis (null)
- $\mathrm{H}_{\mathbf{w},\,\boldsymbol{\nu}}$: alternative hypothesis
- $\mathbf{w}$: trainable parameters on the NN model
- $\boldsymbol{\nu}$: set of nuisance parameters modelling the uncertainties effects
- $\mathcal{D}$: data sample
- $\mathcal{A}$: auxiliary sample (used to constrain $\boldsymbol{\nu}$)

Tau term:

$$\tau(\mathcal{D},\,\mathcal{A}) = 2\max_{\mathbf{w},\,\boldsymbol{\nu}}\log\left[\frac{\mathcal{L}(\mathrm{H}_{\mathbf{w},\,\boldsymbol{\nu}}|\mathcal{D})\,\mathcal{L}(\boldsymbol{\nu}|\mathcal{A})}{\mathcal{L}(\mathrm{R}_{\mathbf{0}}|\mathcal{D})\,\mathcal{L}(\mathbf{0}|\mathcal{A})}\right] = -2\min_{\mathbf{w},\boldsymbol{\nu}} L\left[f(x,\mathbf{w}),\boldsymbol{\nu};\widehat{\delta}(x)\right]$$

Depends on the NN model, sensitive to New Physics

Delta term:

$$\Delta(\mathcal{D},\,\mathcal{A}) = 2\max_{\boldsymbol{\nu}}\log\left[\frac{\mathcal{L}(\mathrm{R}_{\boldsymbol{\nu}}|\mathcal{D})\,\mathcal{L}(\boldsymbol{\nu}|\mathcal{A})}{\mathcal{L}(\mathrm{R}_{\mathbf{0}}|\mathcal{D})\,\mathcal{L}(\mathbf{0}|\mathcal{A})}\right] = -2\min_{\boldsymbol{\nu}} L\left[\boldsymbol{\nu};\widehat{\delta}(x)\right]$$

Purely SM term, sensitive only to uncertainties related discrepancies

"Learning New Physics from an Imperfect Machine" Eur. Phys. J. C

New Physics Learning Machine (NPLM)

τ term

INPUT

- Reference sample (R) label=0
- Data sample (D) label=1
- Pre-trained modules: $\hat{\delta}_1$, $\hat{\delta}_2$ … $\hat{\delta}_n$
- Auxiliary measurements $\hat{\nu}(A)$

Model

BSM module

$$f(\,\cdot\,;\mathbf{w})$$

δ module: $\hat{\delta}_1$, $\hat{\delta}_2$ … $\hat{\delta}_n$

$$\hat{r}(x;\nu) = \exp\left[\hat{\delta}_1(\,\cdot\,)\,\nu + \hat{\delta}_2(\,\cdot\,)\,\nu^2 + \ldots + \hat{\delta}_n(\,\cdot\,)\,\nu^n\right]$$

Loss function

$$L[f(\,\cdot\,;\mathbf{w}),\nu,\hat{\delta}] = \sum_{e\in R} w_e\left[e^{f(x_e,\mathbf{w})+\log\hat{r}(x_e;\nu)} - 1\right] - \sum_{x\in D}\left[f(x,\mathbf{w}) + \log\hat{r}(x;\nu)\right] - \log\left[\frac{\mathscr{L}(\nu\,|\,A)}{\mathscr{L}(0\,|\,A)}\right]$$

Trainable parameters: $\mathbf{w}, \nu$

Constant terms: $\hat{\delta}$

OUTPUT

$$\tau(D,A) = -2\min_{\mathbf{w},\nu} L\left[f(\,\cdot\,,\mathbf{w}),\nu;\hat{\delta}(\,\cdot\,)\right]$$

Δ term

INPUT

- Data sample (D label=1
- Pre-trained modules: $\hat{\delta}_1$, $\hat{\delta}_2$ … $\hat{\delta}_n$
- Auxiliary measurements $\hat{\nu}(A)$

Model

δ module: $\hat{\delta}_1$, $\hat{\delta}_2$ … $\hat{\delta}_n$

$$\hat{r}(x;\nu) = \exp\left[\hat{\delta}_1(\,\cdot\,)\,\nu + \hat{\delta}_2(\,\cdot\,)\,\nu^2 + \ldots + \hat{\delta}_n(\,\cdot\,)\,\nu^n\right]$$

Loss function

$$L[f(\,\cdot\,;\mathbf{w}),\nu,\hat{\delta}] = \mathrm{N}(\mathrm{R}_\nu) - \mathrm{N}(\mathrm{R}_0) - \sum_{x\in D}\left[\log\hat{r}(x;\nu)\right] - \log\left[\frac{\mathscr{L}(\nu\,|\,A)}{\mathscr{L}(0\,|\,A)}\right]$$

Trainable parameters: ν

Constant terms: $\hat{\delta}$

OUTPUT

$$\Delta(D,A) = -2\min_{\mathbf{w},\nu} L\left[\nu;\hat{\delta}(\,\cdot\,)\right]$$

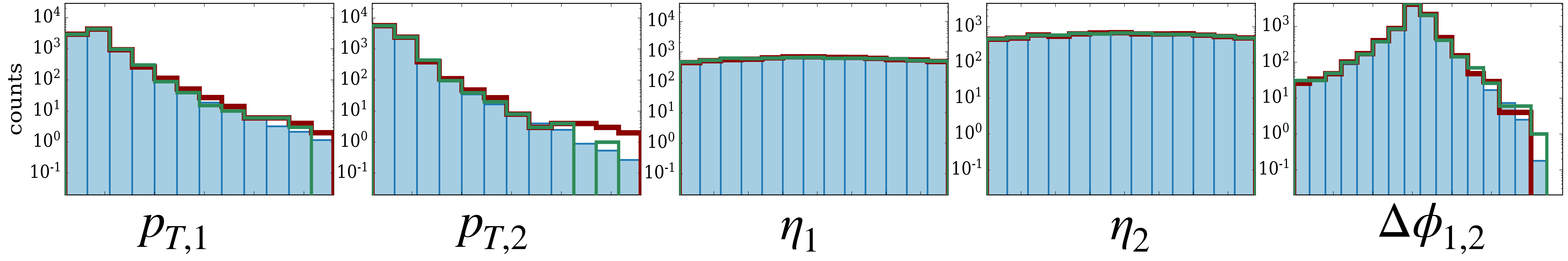
New Physics Learning Machine (NPLM)

Modelling the family of Reference hypotheses: shape effects

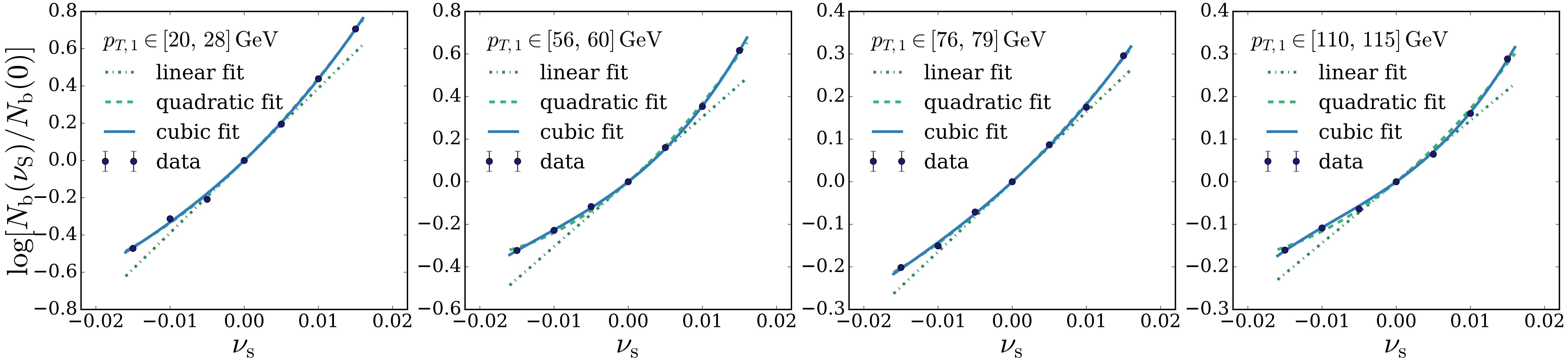
1) Preliminary study: Binned analysis to determine the proper order for the Taylor's expansion

Example:
Two-body final state
(5D analysis)

Input: set of variables describing the kinematic of the final state



Bin-wise scale effects on the electrons p_T distribution



- Right order of approximation
- Where and how many instances are needed

New Physics Learning Machine (NPLM)

Modelling the family of Reference hypotheses: shape effects

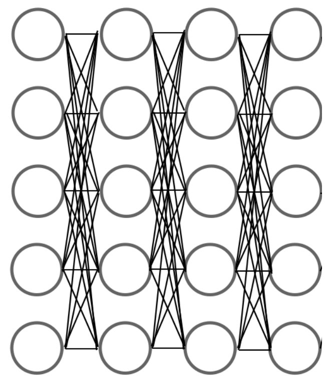
2) Taylor's expansion learning: Training a neural network model to learn each coefficient of the Taylor's expansion of

$$r(x;\nu) = \frac{n(x\,|\,\mathrm{R}_\nu)}{n(x\,|\,\mathrm{R}_0)}$$

Parametrised classifier

Input samples

ν_1	ν_2	…	ν_n
$S_0(\nu_1) \sim \mathrm{R}_0$	$S_0(\nu_2) \sim \mathrm{R}_0$	…	$S_0(\nu_n) \sim \mathrm{R}_0$
$S_1(\nu_1) \sim \mathrm{R}_{\nu_1}$	$S_1(\nu_2) \sim \mathrm{R}_{\nu_2}$		$S_1(\nu_n) \sim \mathrm{R}_{\nu_n}$



$\hat{\delta}_1$

$\hat{\delta}_2$

$$\hat{r}(x;\nu) = \exp\left[\hat{\delta}_1(x)\,\nu + \hat{\delta}_2(x)\,\nu^2\right]$$

Loss function*

$$L[\hat{\delta}(\,\cdot\,)] = \sum_{\nu_i}\left[\sum_{e\in S_0(\nu_i)} w_e c(x_e)^2 + \sum_{e\in S_1(\nu_i)} w_e\left[1 - c(x_e)\right]^2\right], \qquad c(x) = \frac{1}{1+\hat{r}(x;\nu)}$$

* *Parametrized classifiers for optimal EFT sensitivity* arXiv:2007.10356

New Physics Learning Machine (NPLM)

Controlling type I errors: validation of the $(\tau - \Delta)$ procedure

"Toy Data" : test the procedure on simulated toys following the Reference (SM) hypothesis with generation value for the nuisance parameters $\nu^* = \pm\sigma_\nu$:

$$\mathcal{D} \sim \mathrm{R}_{\boldsymbol{\nu}^*}, \quad \boldsymbol{\nu}^* = \pm\sigma_{\boldsymbol{\nu}}$$

The $t(D)$ distribution under the reference hypothesis $\mathrm{R}_{\boldsymbol{\nu}^*}$ is **compatible with the** $\chi^2_{|\mathbf{w}|}$ (found by regularizing) for values of the true nuisance parameters within the uncertainty ($\boldsymbol{\nu}^* = \pm\sigma_{\boldsymbol{\nu}}$).

t does **not depend** on the true value of the nuisance parameters!

We can build a *frequentist* test statistic targeting the $\chi^2_{|\mathbf{w}|}$.

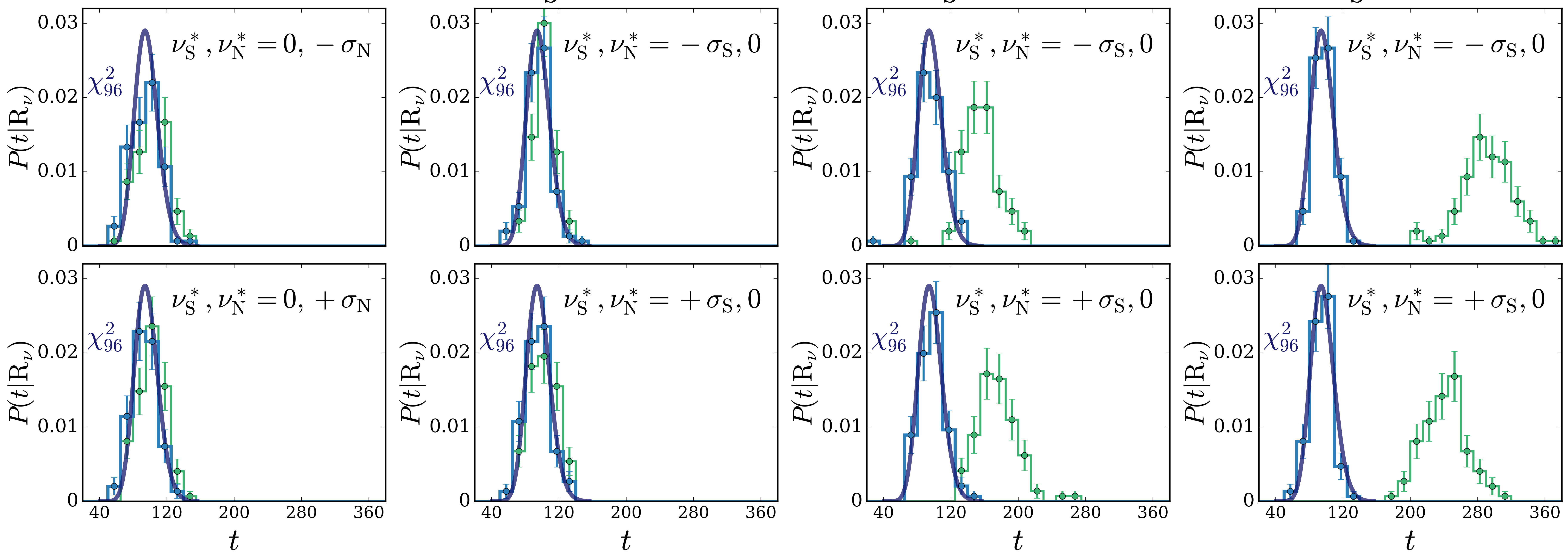
New Physics Learning Machine (NPLM)

Controlling type I errors: validation of the $(\tau - \Delta)$ procedure

Reference sample: $R \sim \mathrm{R}_0$
Data sample: $D \sim \mathrm{R}_{\nu^*}$

Normalization uncertainty $\sigma_N = 2.5\,\%$	Scale uncertainty Muon-like regime $\sigma_S = 0.05\,\%$	Scale uncertainty Electron-like regime $\sigma_S = 0.3\,\%$	Scale uncertainty Tau-like regime $\sigma_S = 3\,\%$

Example:
NN [5-5-5-5-1]
trainable parameters = 96
weight clipping = 2.16



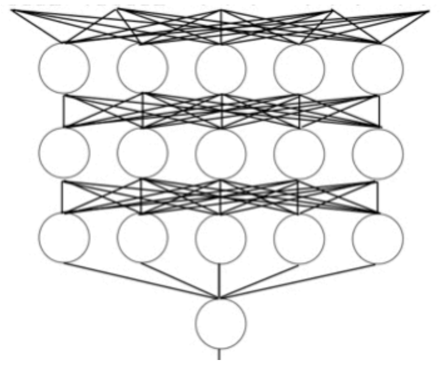
$\tau(D,\ A)$
$t(D,\ A) = \tau(D,\ A) - \Delta(D,\ A)$

Training speed up: **kernel methods**

- NN model replaced with kernels:

Neural Network model

$f(x, w) = \mathrm{NN}(x, w)$



Kernel-based model (FALKON)

$$f_w(x) = \sum_{i=1}^{M} w_i k_\sigma(x, \tilde{x}_i),$$

$$k_\sigma(x, x') = e^{-\|x - x'\|^2 / 2\sigma^2}$$

- NPLM loss replaced with a weighted cross entropy loss

$$\hat{L}(f_w) + \lambda R(f_w)$$

Weighted cross entropy:

$$\hat{L}(f_w) = \frac{1}{N} \sum_{i=1}^{N} a_0 (1 - y) \log\left(1 + e^{f(x)}\right) + a_1 y \log\left(1 + e^{-f(x)}\right)$$

Regularization term:

$$R(f_w) = \sum_{ij} w_i w_j k_\sigma(x_i, x_j)$$

This is also a smoothness requirement!

- Different heuristic to select the model hyperparameters (M, σ, λ)
- Online training on **GPU**: drastic drop of training time!
- (Systematic uncertainties not yet implemented)

UniGe | MaLGa

Learning new physics efficiently with nonparametric methods. [Eur. Phys. J. C, 82(10)]