Βελτιστοποίηση Υπερπαραμέτρων για Σύνολα PDF σε Πολλαπλές GPUs

Tanjona R. Rabemananjara in Collab. Juan M. Cruz-Martinez, Tommaso Giani, NLeSC, NNPDF Collaboration

Workshop on Future Accelerators

April 2026, Corfu, Greece

Hyperparameter Optimisation of PDF Ensemble On multiple GPUs

Tanjona R. Rabemananjara in Collab. Juan M. Cruz-Martinez, Tommaso Giani, NLeSC, NNPDF Collaboration

Workshop on Future Accelerators

April 2026, Corfu, Greece

cnrs

IJCLab Irène Joliot-Curie Laboratoire de Physique des 2 Infinis

université PARIS-SACLAY

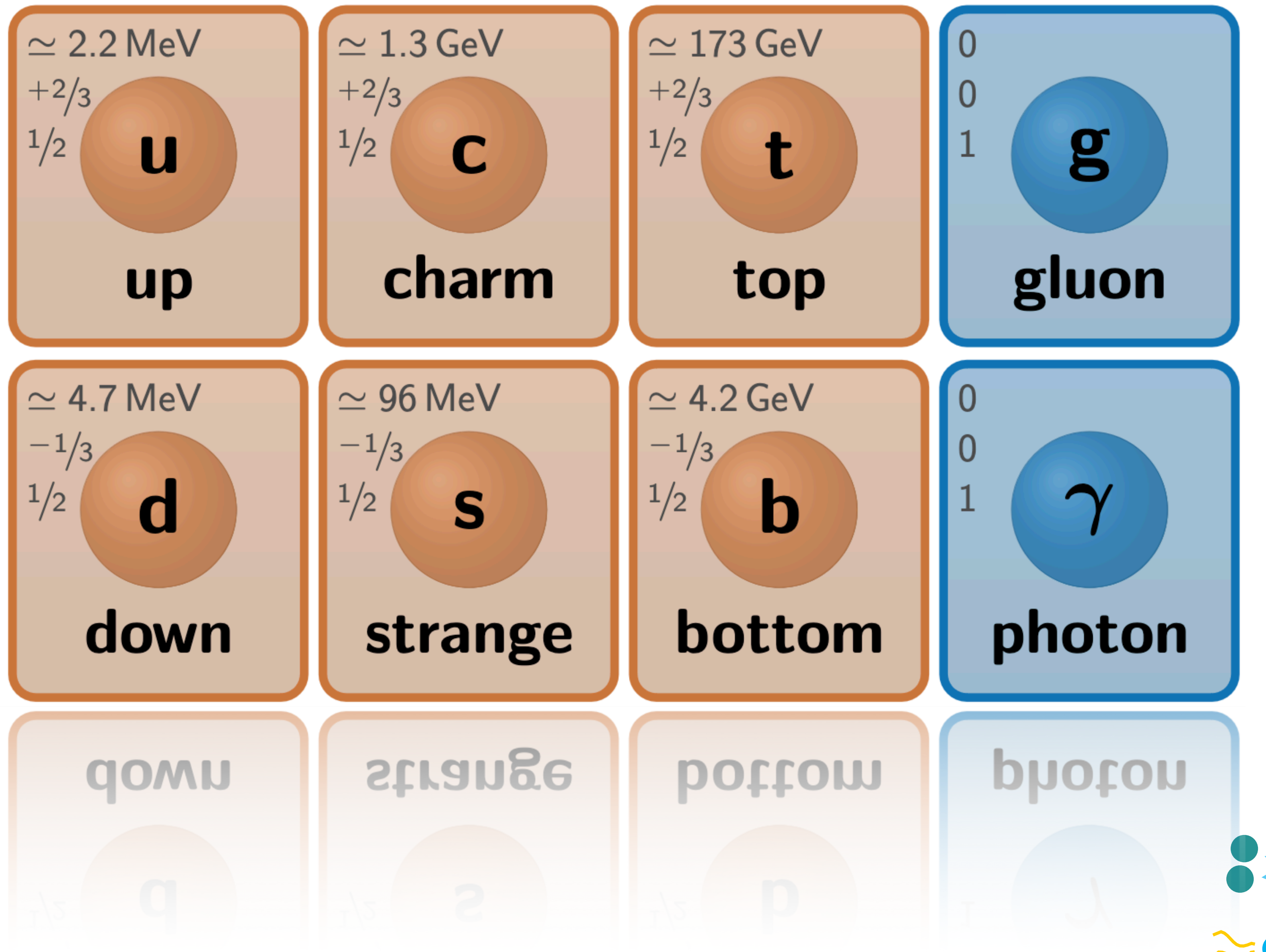
anr agence nationale de la recherche

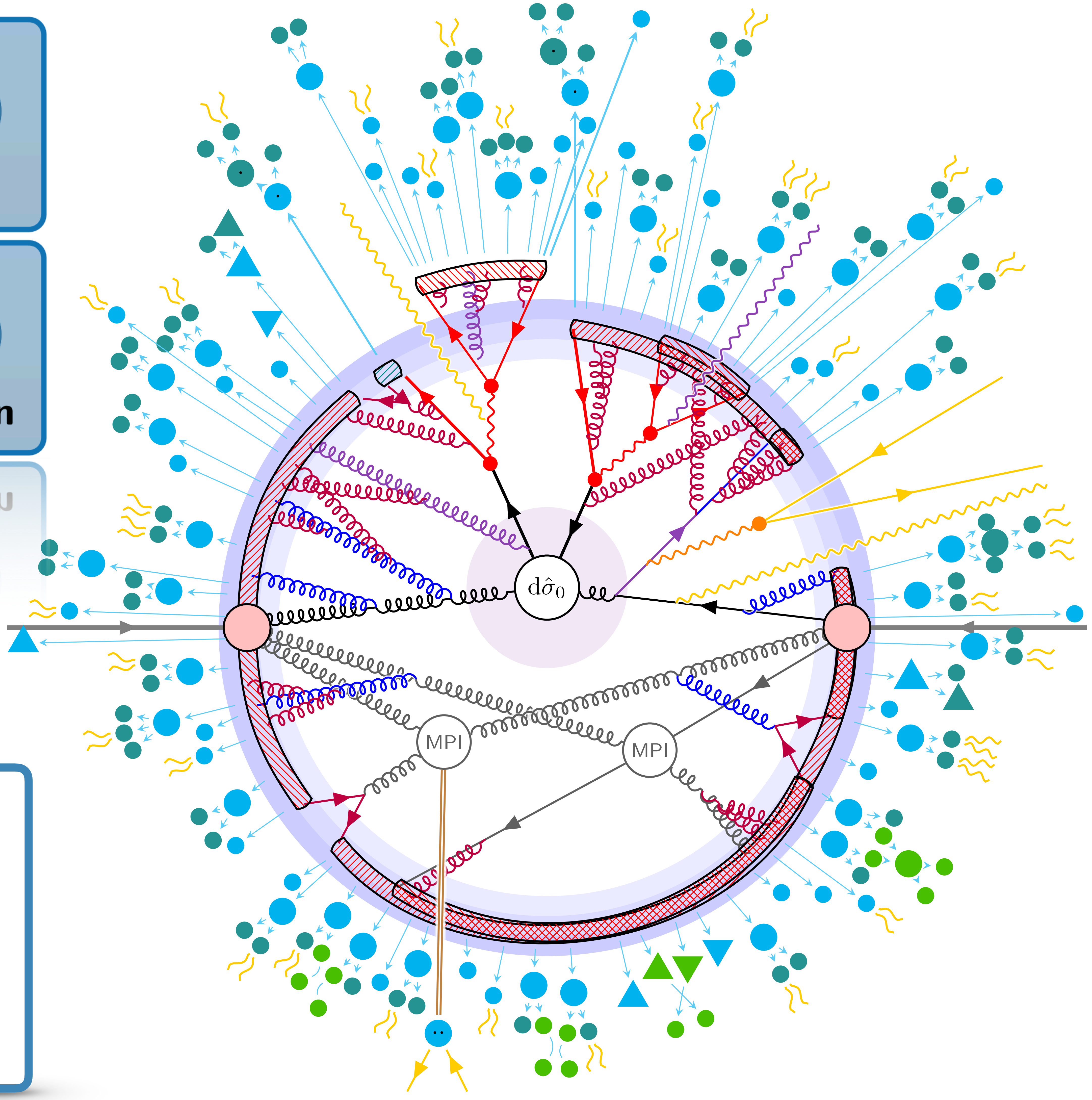
NNPDF

1. Global PDF Extraction in a Nutshell
2. Hyperparameter Optimisation (HPO)& Distributed Training on High-Performance Computers (HPCs)
3. HPO for PDF Ensemble
4. Uncertainty Validations

netherlands eScience center

Predictions in High-Energy Physics

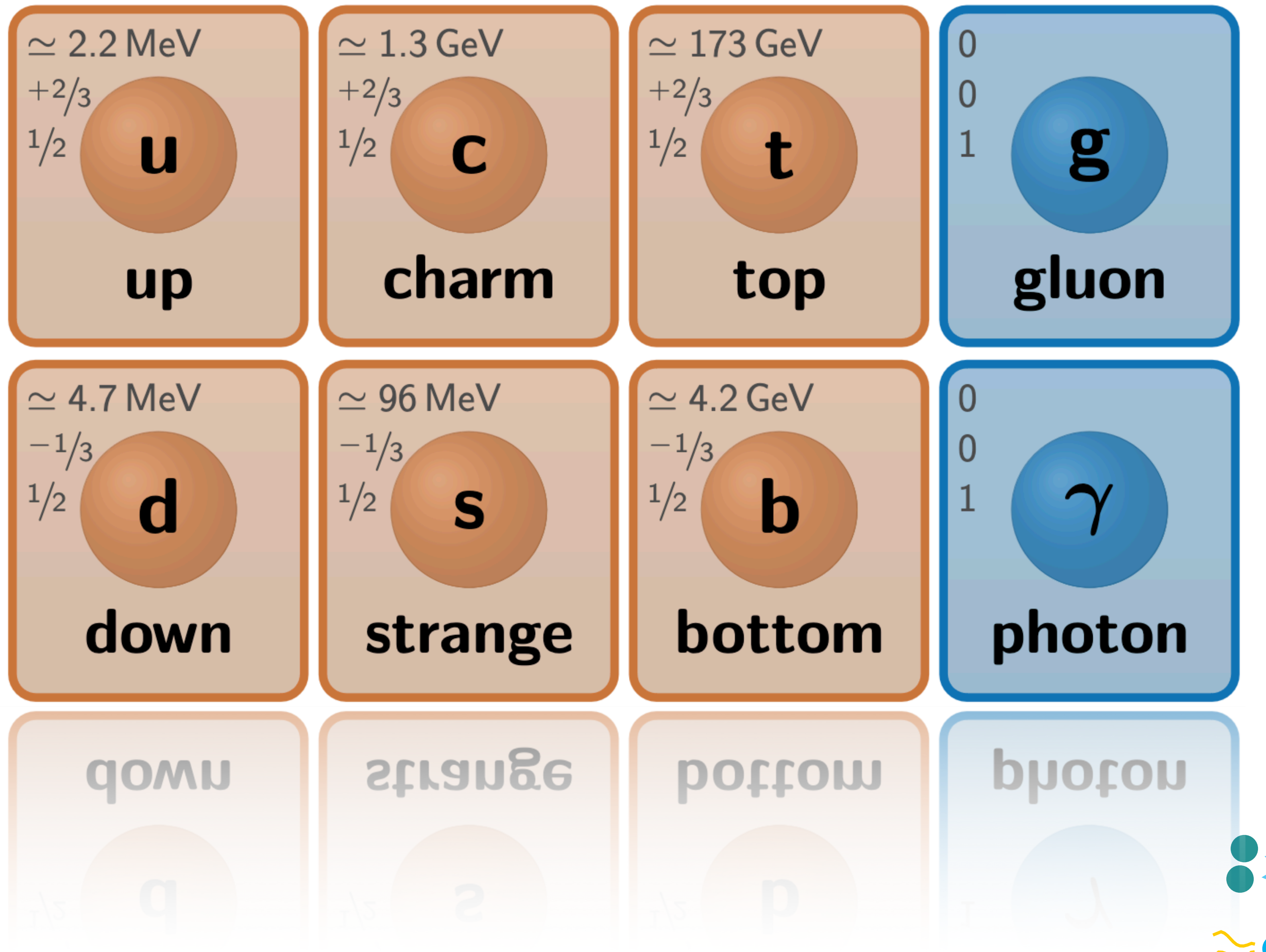


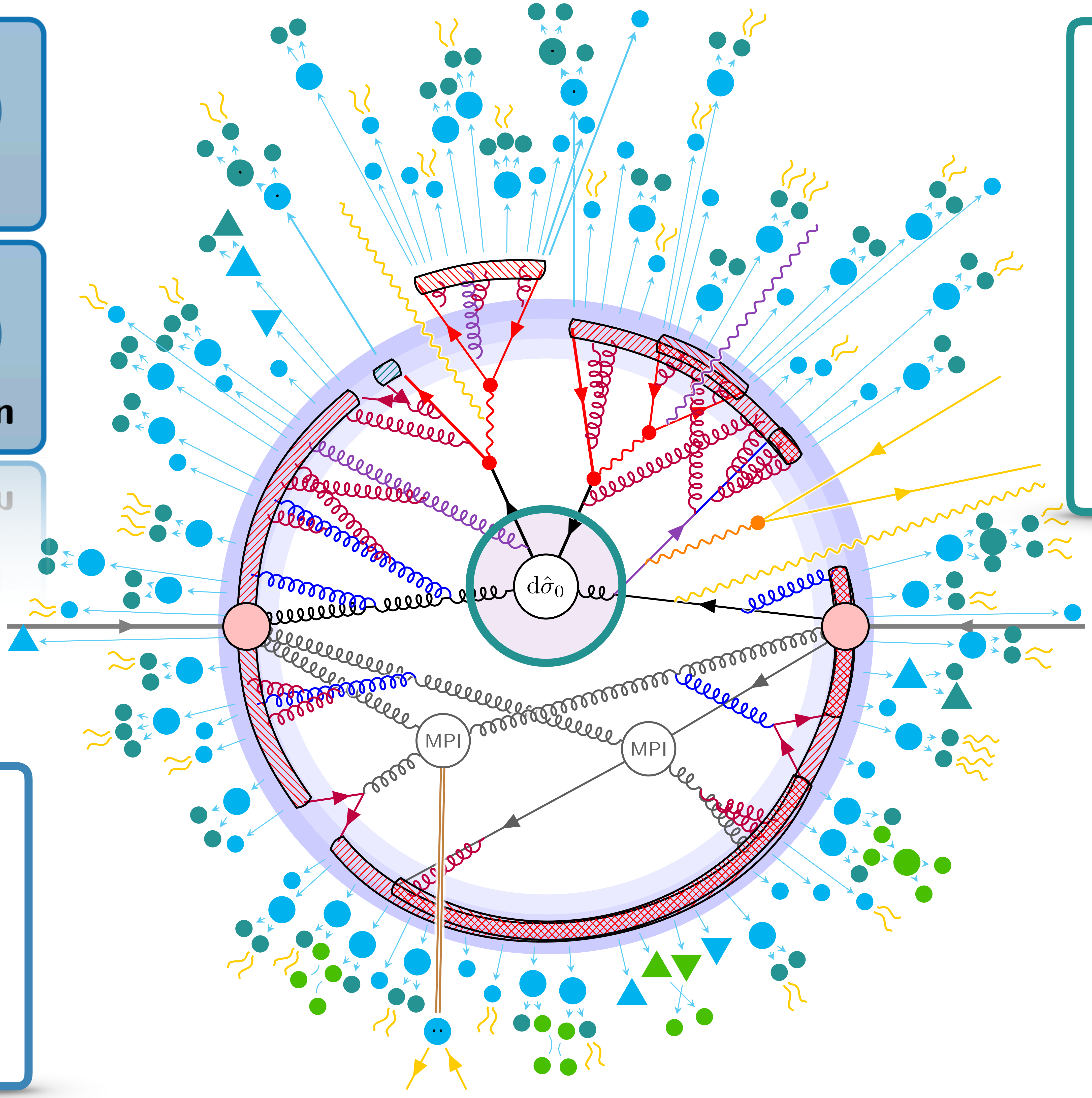


Experiments measure an Observable $\mathcal{O}$ (cross-section, decay rates, etc.):

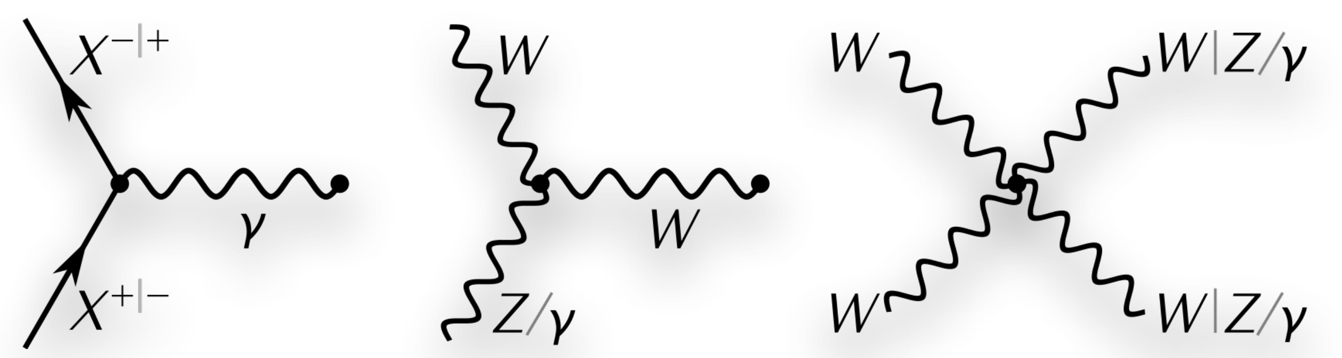
$$\mathcal{O} = \sum \hat{\sigma}_0 \otimes \text{PDF}$$

Predictions in High-Energy Physics





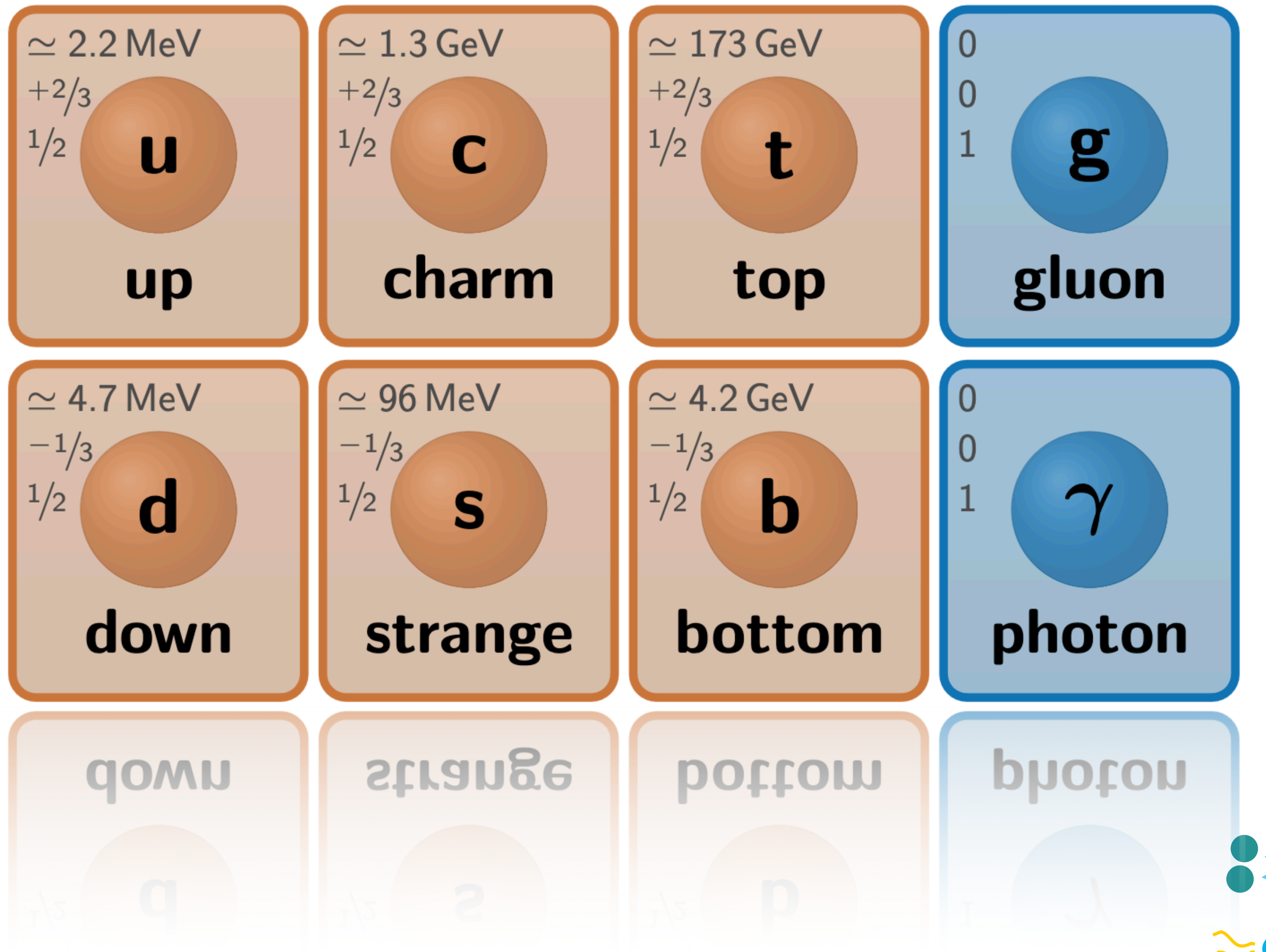
Hard scattering $\hat{\sigma}_0$: encodes short-range interactions; computed from first principles.



Experiments measure an Observable $\mathcal{O}$ (cross-section, decay rates, etc.):

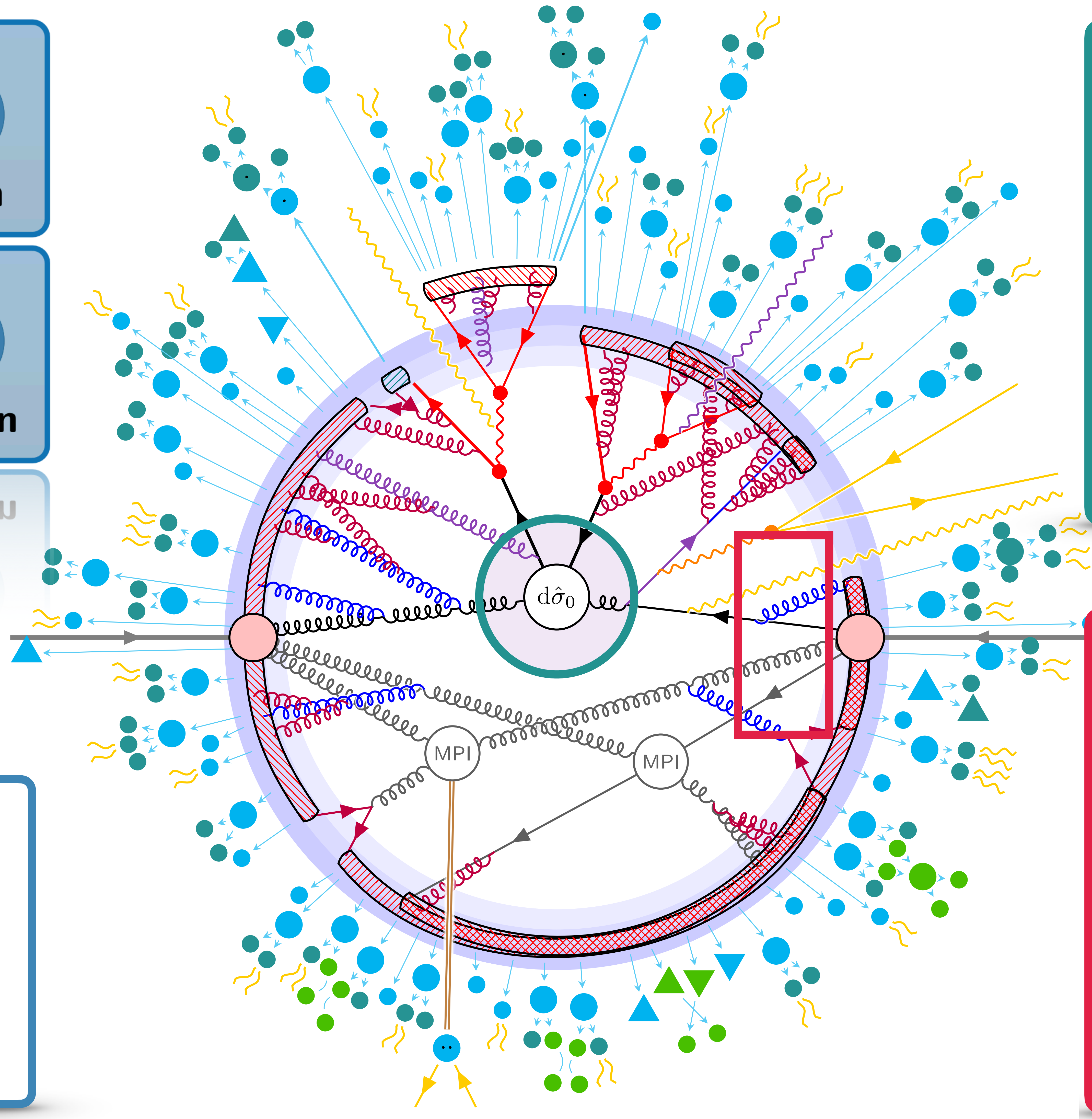
$$\mathcal{O} = \sum \hat{\sigma}_0 \otimes \text{PDF}$$

Predictions in High-Energy Physics

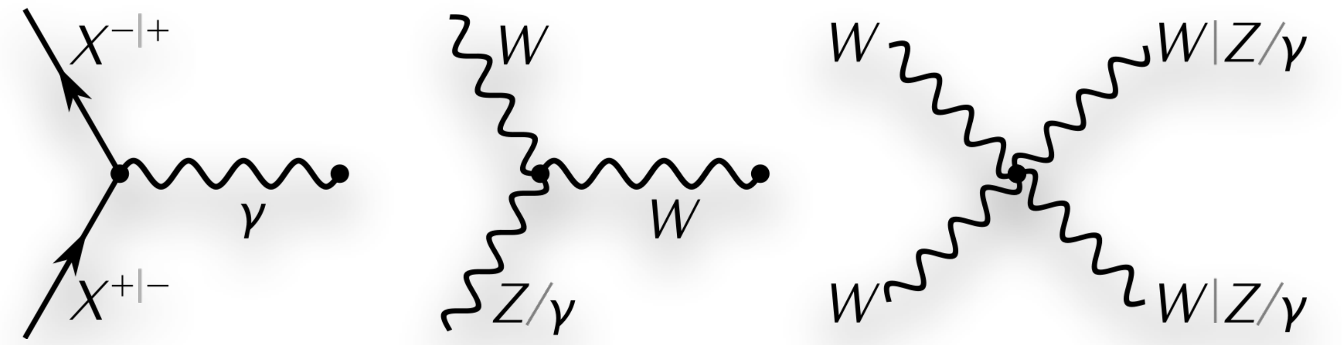


Experiments measure an Observable $\mathcal{O}$ (cross-section, decay rates, etc.):

$$\mathcal{O} = \sum \hat{\sigma}_0 \otimes \text{PDF}$$



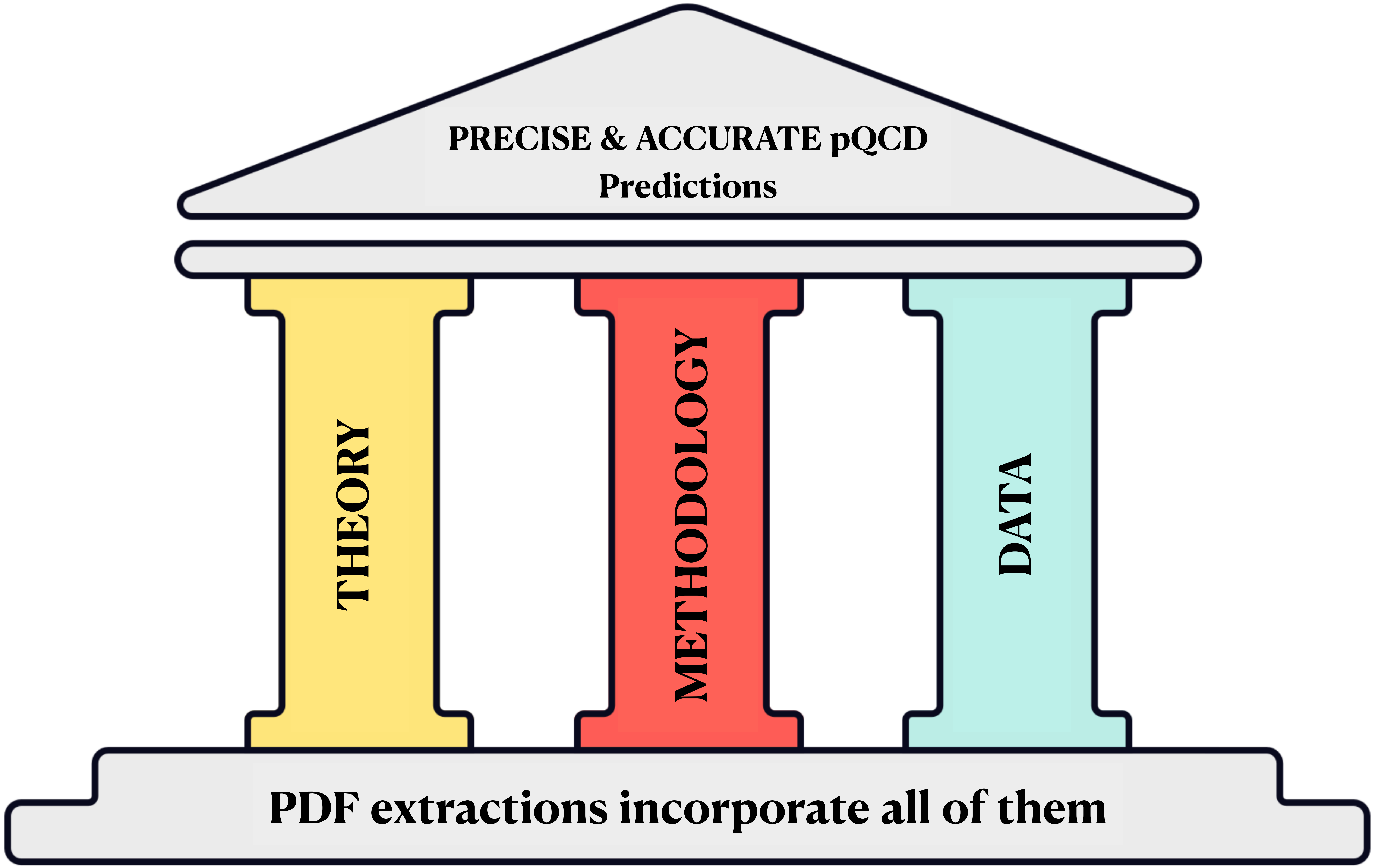
Hard scattering $\hat{\sigma}_0$: encodes short-range interactions; computed from first principles.



Parton Distribution Functions (PDFs): encodes long-range non-perturbative interactions; cannot be computed from first principle and have to be determined from experimental Data.

PDFs are Universal

Three Pillars of pQCD Predictions

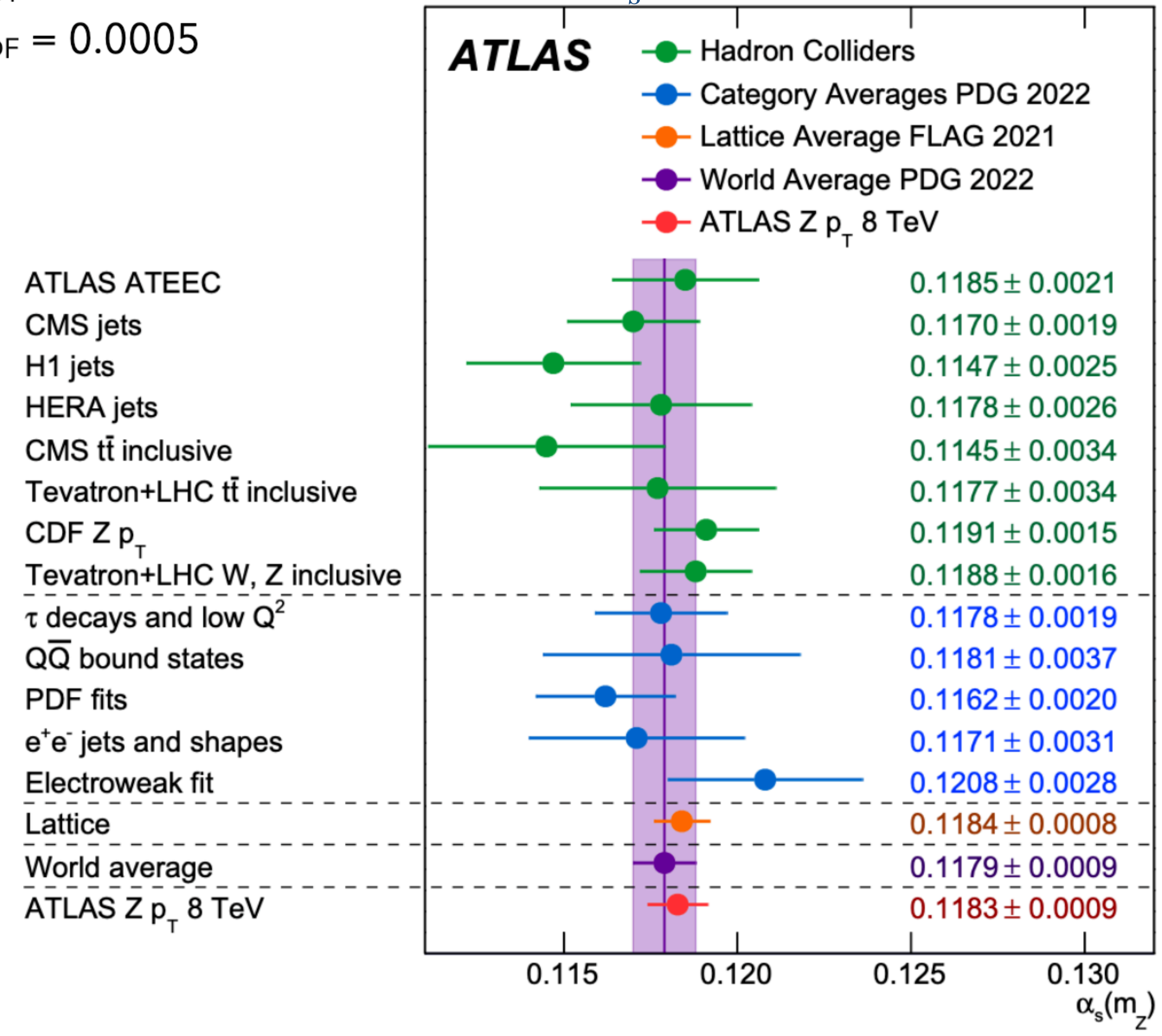


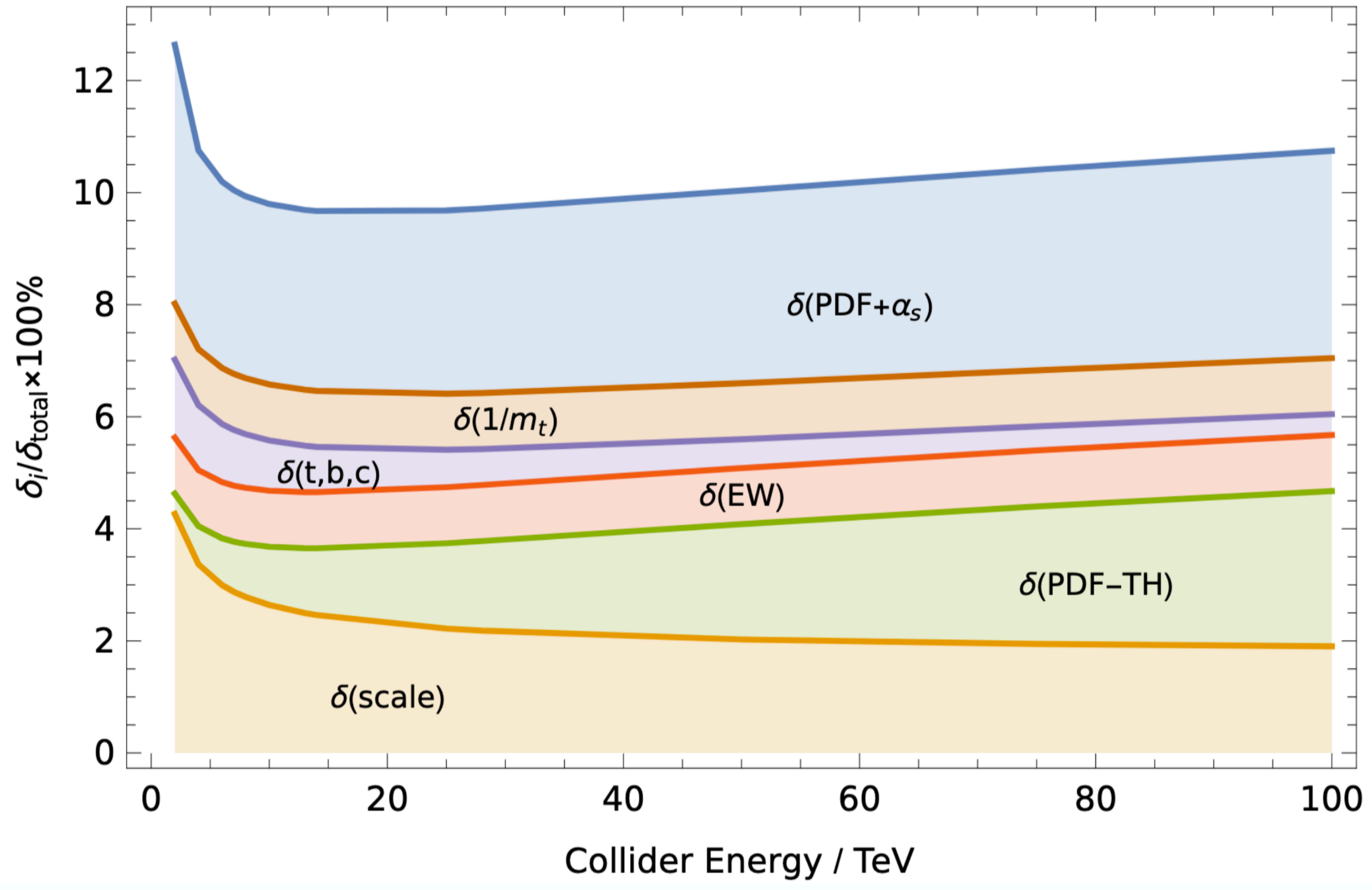
Why PDFs are still relevant?

PDFs are becoming a bottleneck for LHC precision calculations with the largest uncertainties ⟺ **Could be Hiding New Physics**

Δ_{TOT} = 0.0009
Δ_{PDF} = 0.0005

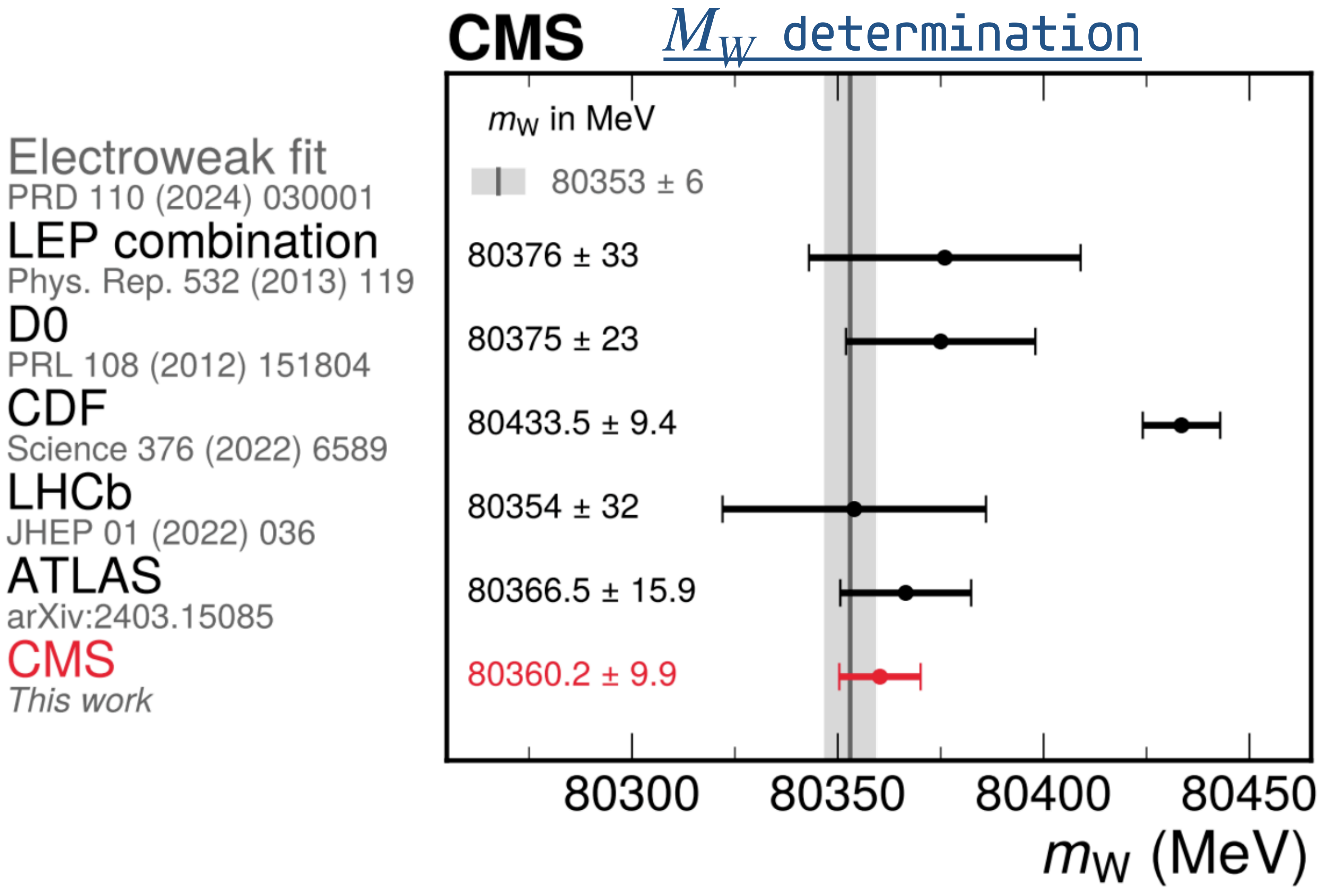
α_s determination





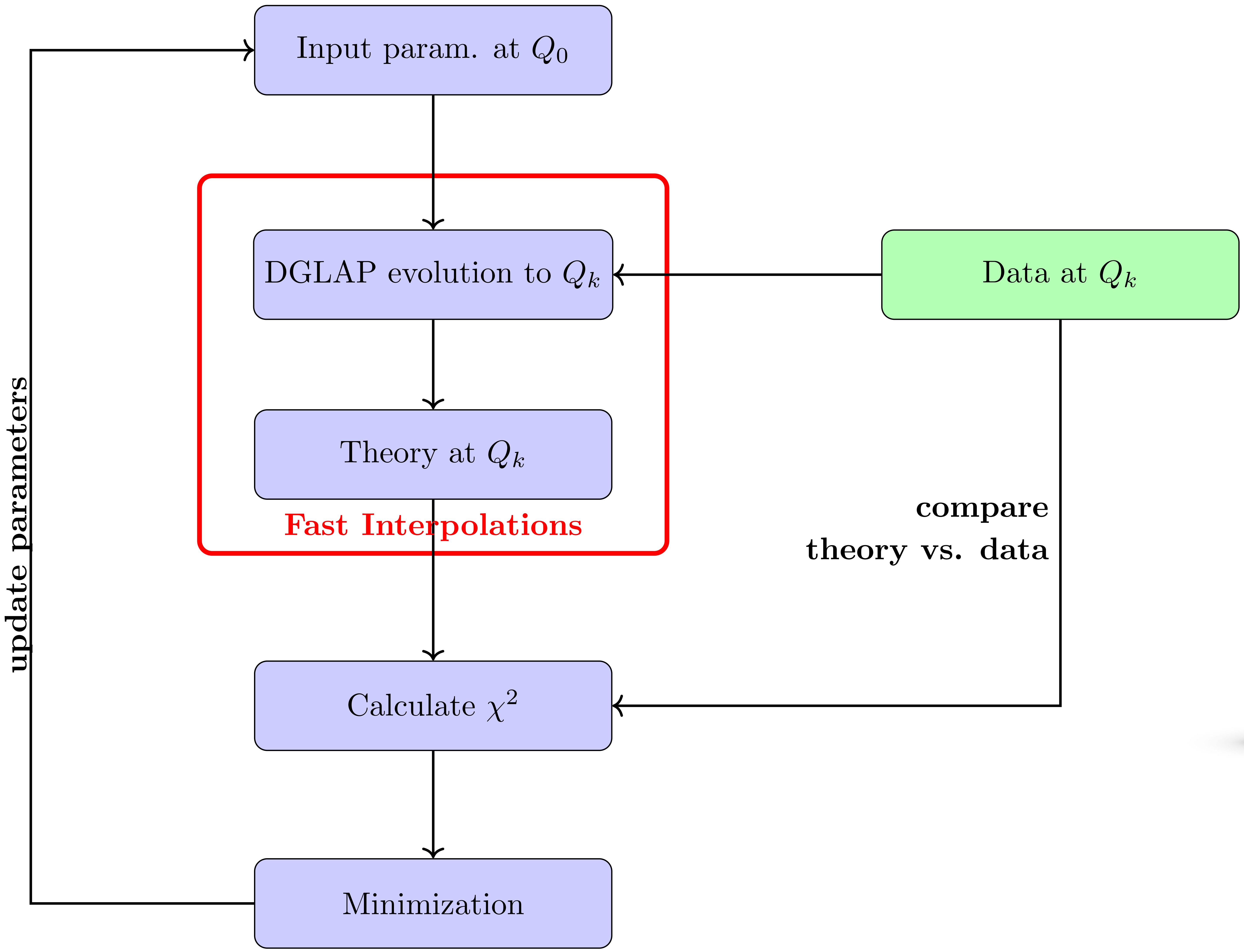
Δ_{TOT} = 9.9 MeV
Δ_{PDF} = 4.4 MeV

M_W determination



1. Global PDF Extractions in a Nutshell

Anatomy of PDF Extraction



Perturbative Part: MC Simulations & HPC

Theoretical predictions are computed analytically and numerically

The Q^2-dependence is captured via an Evolution Eq:

$$f_i\left(x_\alpha, Q_k\right) = E_{ij,\alpha,\beta}\left(Q_k \leftarrow Q_0\right) f_j\left(x_\beta, Q_0\right)$$

The Evolution Operators satisfy the **Renormalisation Group Equation (RGE)**:

$$\frac{d}{d\ln Q^2}\mathbf{F}(x, Q^2) = \mathbf{E}(x, Q^2)\mathbf{F}(x, Q^2)$$

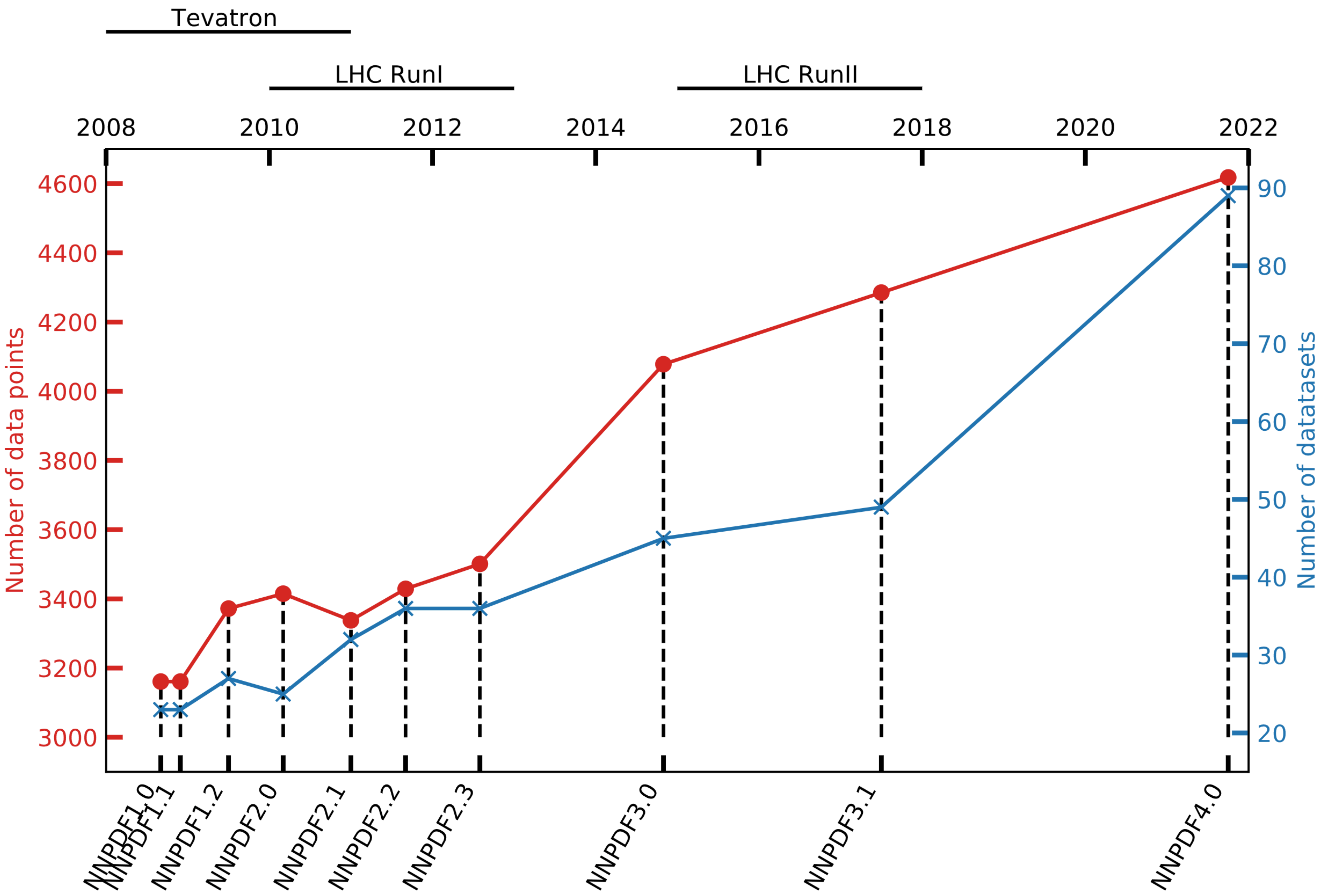
$$\mathbf{E}\left(Q_k \leftarrow Q_0\right) = \mathscr{P}\exp\left(-\int_{Q_k^2}^{Q_0^2}\frac{d\mu^2}{\mu^2}\mathbf{\Gamma}(\mu^2)\right)$$

Where $\mathbf{\Gamma}$ is a $N \times N$ matrix and $\mathscr{P}$ a normalisation.

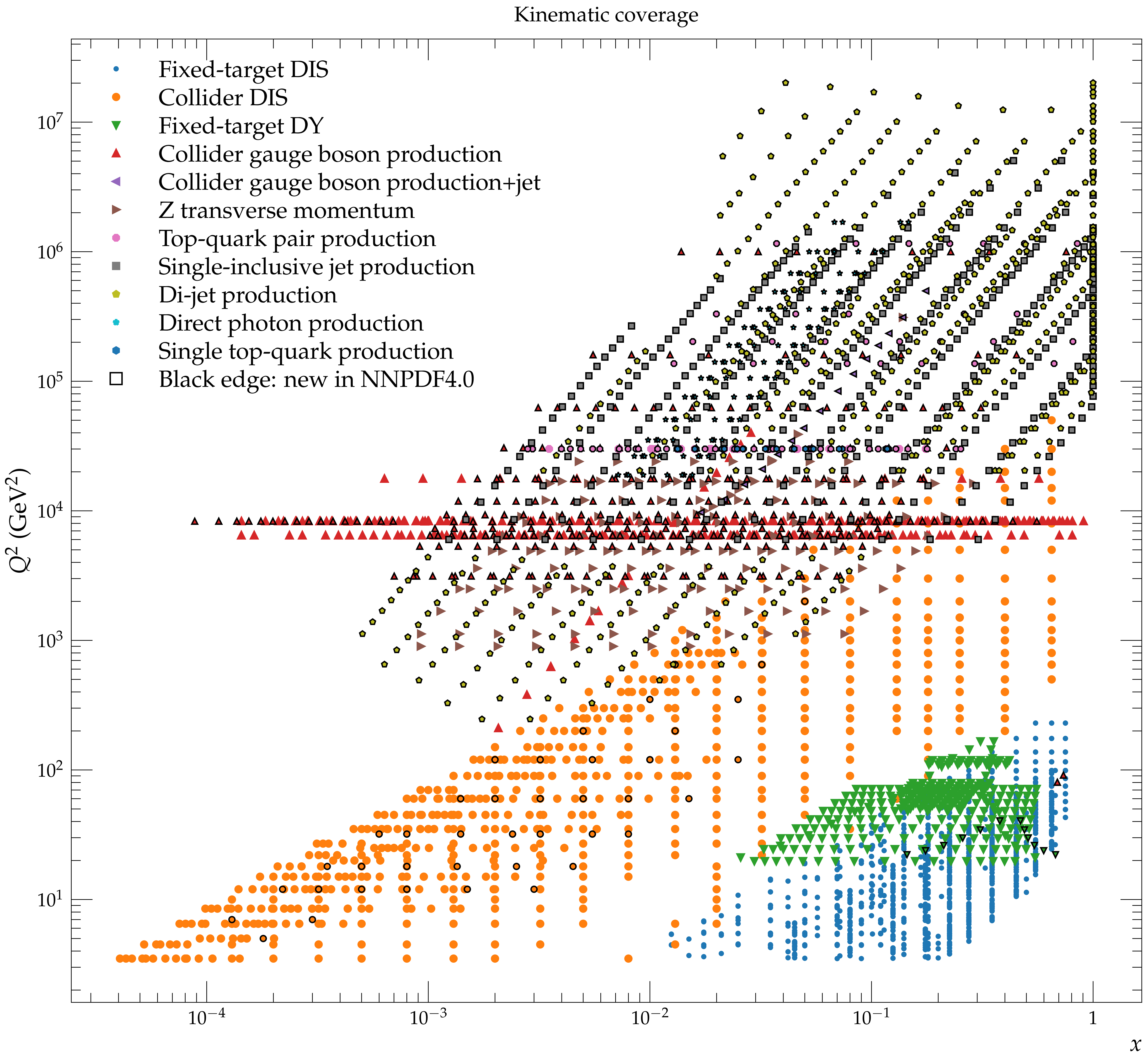
Non-Perturbative Part: Data-Driven

Uses DNNs for parametrisation and SGD for Minimisation

Experimental Measurements:From NNPDF1.0 to 4.0



Experimental Measurements



Snapshot of the Experimental Datasets

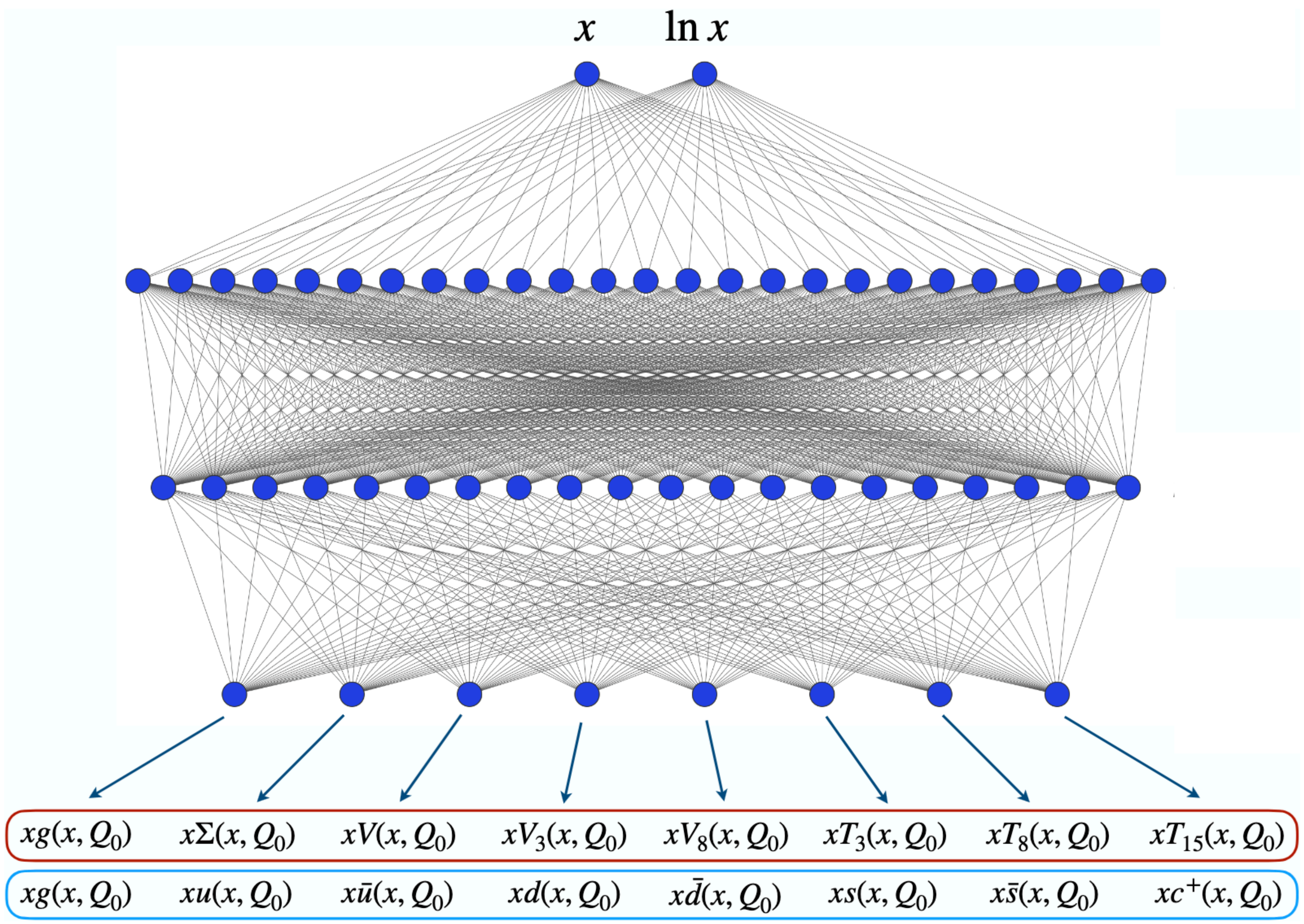
- $\mathcal{O}(5000)$ **datapoints** that span a wide range of kinematic regions and probe various channels $\Longrightarrow$ **Large space of functional forms**
- Precision of the data reach the **percent level accuracy**; mostly from correlated systematic uncertainties
- Significant amount of the datasets ($\mathcal{O}(500)$ **datapoints**) were introduced in the NNPDF4.0 release (LHC Run II data)

Parametrisation: Different Approaches

	NNPDF4.0	MSHT20	CT18	ABMP16
Released	*Sept 2021*: LHAPDF grids + **fitting code**	*Dec 2020*: LHAPDF grids	*Dec 2019*: LHAPDF grids	*Jan 2017*: LHAPDF grids
Parametrisation	Neural networks (**hyperoptimised**)	Functional form + Chebyshev	Functional form + Bernstein	Functional form
Error estimate	Monte Carlo (**closure + future tested**)	Hessian (dynamic tolerance)	Hessian (dynamic tolerance) + Lagrange mult.	Hessian (no tolerance)
Theory settings	NNLO QCD, GM-VFN (+ NLO electroweak) aN3LO	NNLO QCD GM-VFN aN3LO	NNLO QCD GM-VFN	NNLO QCD, FFN

Parametrisation: The NNPDF Approach

NNPDF relies on **DNNs** to parametrise the PDFs using **Stochastic Gradient Descent (SGD)** as Minimisers



PDFs are usually extracted in the so-called **Evolution Basis**.

Momentum Sum Rules:

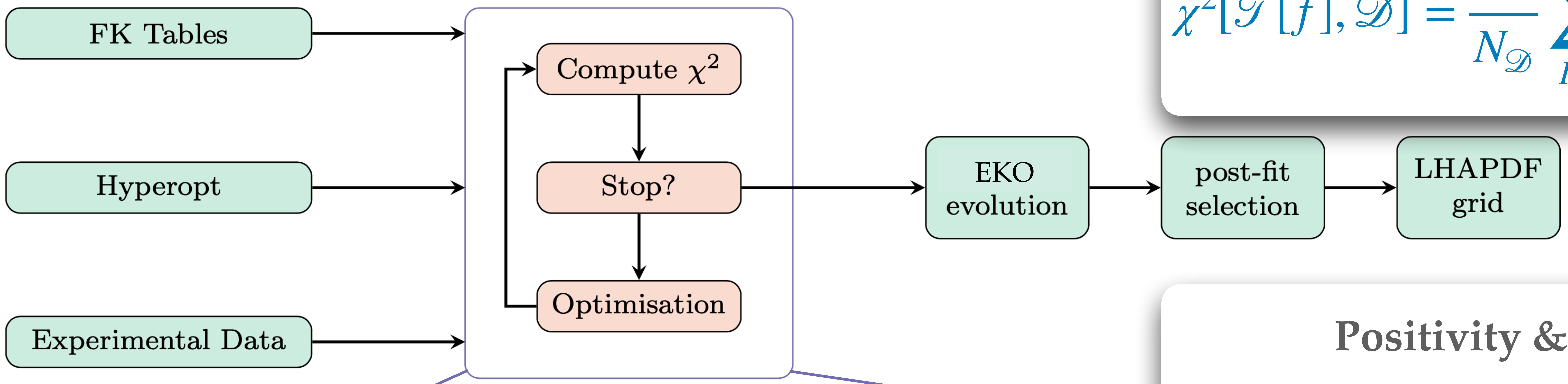
$$\int_0^1 dx V_3(x,Q) = 1$$
$$\int_0^1 dx V(x,Q) = \int_0^1 dx V_8(x,Q) = 3$$

PDF Integrability:

$$\lim_{x\to 0} x^2 f_k(x,Q) = 0, \quad \forall Q, \quad f_k = g, \Sigma$$
$$\lim_{x\to 0} x f_k(x,Q) = 0, \quad \forall Q, \quad f_k = V, V_3, V_8$$
$$\lim_{x\to 0} x f_k(x,Q) = 0, \quad \forall Q, \quad f_k = T_3, T_8$$

The NN fitting Procedures

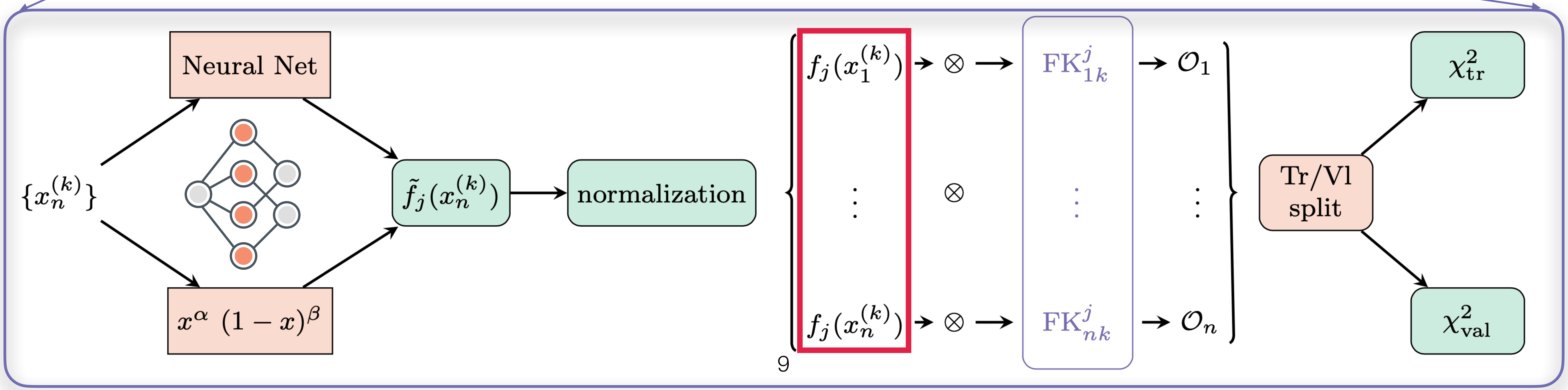
python TensorFlow PyTorch



$$\chi^2[\mathcal{T}[f],\mathcal{D}] = \frac{1}{N_{\mathcal{D}}}\sum_{I,J}\left(\mathcal{T}_I[f]-D_I\right) C_{IJ}^{-1}\left(\mathcal{T}_J[f]-D_J\right)$$

Positivity & Integrability constraints:

$$\chi^2_{\text{tot}} \to \chi^2_{\text{tot}} + \sum_{k=1}^{8}\Lambda_k \sum_{i=1}^{n_i}\mathcal{F}\left(\tilde{f}_k\left(x_i,Q^2\right)\right)$$



Uncertainty Propagation (1)

Problem: Given a set of dataset D, determine $p(f, D)$ in the space: $f : [0,1] \to \mathbb{R}$

Approach: Approximate $p(f, D)$ with its projection in the space parameters $p(\theta | D)$

$$xf_i\left(x, Q_0^2\right) = A_i x^{\alpha_i}(1-x)^{\beta_i}\mathscr{F}(x, \theta)$$

Determine $p(\theta | D) \sim p(D | \theta)p(\theta)$ as $\theta^* = \arg\max_\theta p(\theta \mid D)$

$$\chi^2 = \sum_{i,j}^{N_{\text{dat}}} \left[T_i[\theta] - D_i\right] \left(\text{cov}^{-1}\right)_{ij} \left[T_j[\theta] - D_j\right], \quad \text{cov}_{ij} = \delta_{ij}\left(\delta D_i^{\text{UNC}}\right)^2 + \sum_{k=1}^{N_{\text{COR}}} \delta D_{k,i}^{\text{COR}} \delta D_{k,j}^{\text{COR}}$$

Two prescriptions exist to compute the expected value and the associated uncertainties

$$E[\mathcal{O}] = \int \mathscr{D}f \mathscr{P}(f \mid D)\mathcal{O}(f) \qquad V[\mathcal{O}] = \int \mathscr{D}f \mathscr{P}(f \mid D)[\mathcal{O}(f) - E[\mathcal{O}]]^2$$

Monte Carlo

$$\mathscr{P}(f \mid D) \longrightarrow \{f_k\}$$
$$E[\mathcal{O}] \approx \frac{1}{N}\sum_k \mathcal{O}\left(f_k\right)$$
$$V[\mathcal{O}] \approx \frac{1}{N}\sum_k \left[\mathcal{O}\left(f_k\right) - E[\mathcal{O}]\right]^2$$

Hessian

$$\mathscr{P}(f \mid D) \longrightarrow f_0$$
$$E[\mathcal{O}] \approx \mathcal{O}\left(f_0\right)$$
$$V[\mathcal{O}] \approx \text{Hessian}$$

Uncertainty Propagation (2)

Monte Carlo Representation

Experimental uncertainties are propagated into the proton PDF fit by **fluctuating the central data** w.r.t. the uncertainties coming from the experimental inputs

$$\mathscr{D}_k = \mathscr{D}_k^{(0)} + \sum_{\ell=1}^{n_D} \sqrt{\mathrm{Cov}_{k\ell}} \times \delta_\ell$$

Instances of such samplings are called **"Pseudodata Replicas"**. Each of the pseudodata replica is then fitted to a NN with different training / validation random seeds.

The final output - which defines the PDF distribution - is an ensemble of PDF replicas.

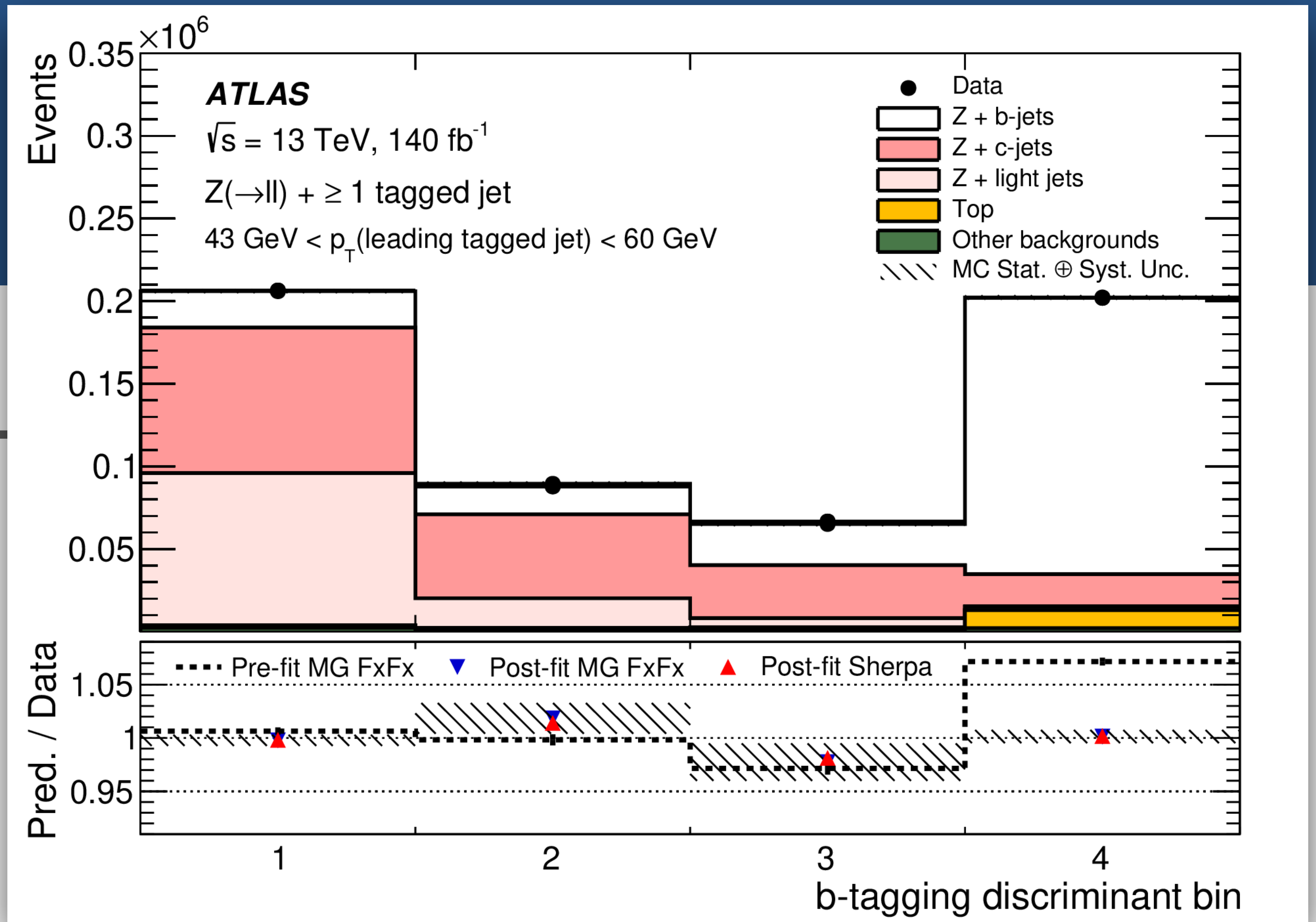
r_1 → $\mathrm{NN}(\theta_1)$ → PDF_1

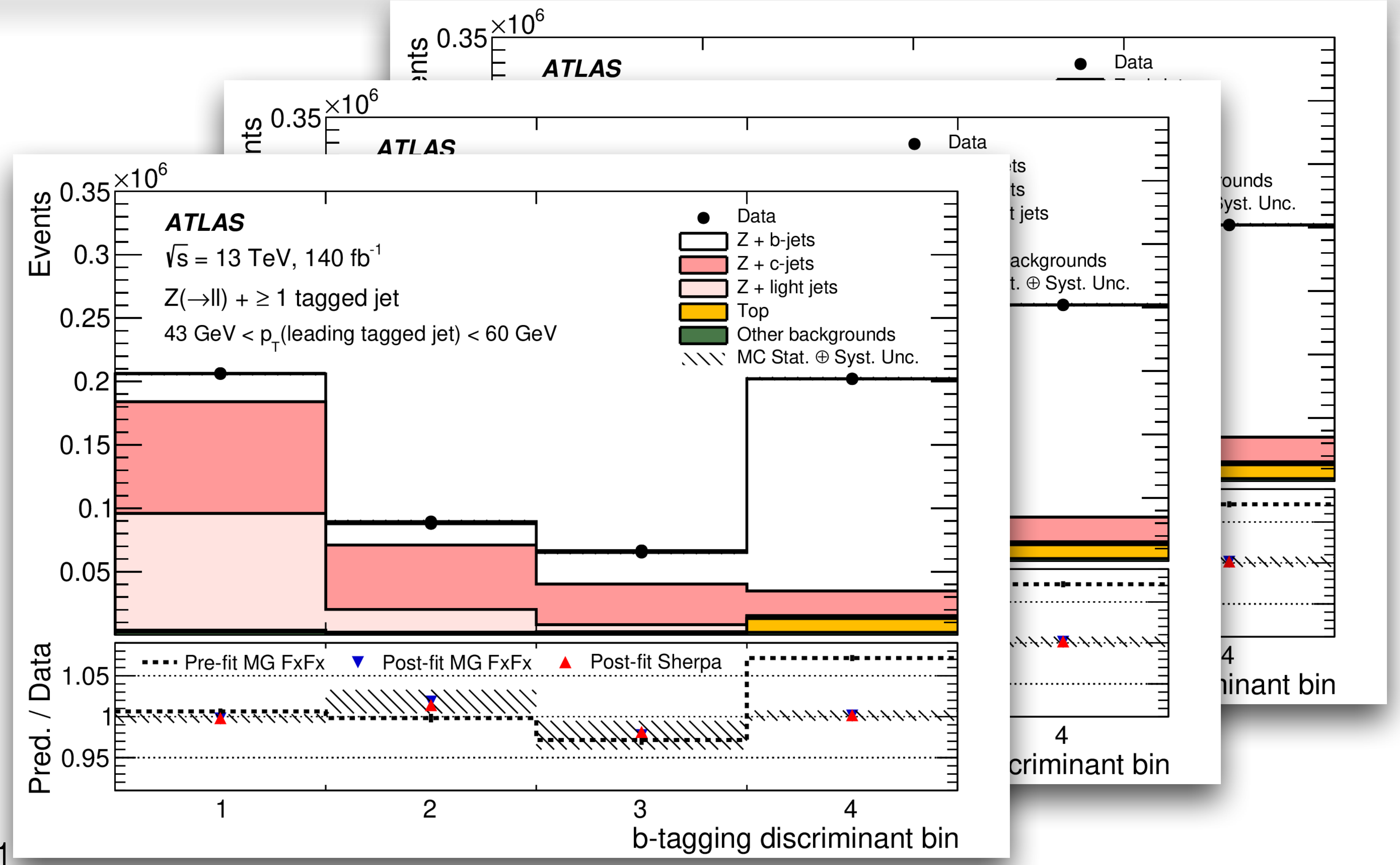
r_2 → $\mathrm{NN}(\theta_2)$ → PDF_2

⋮

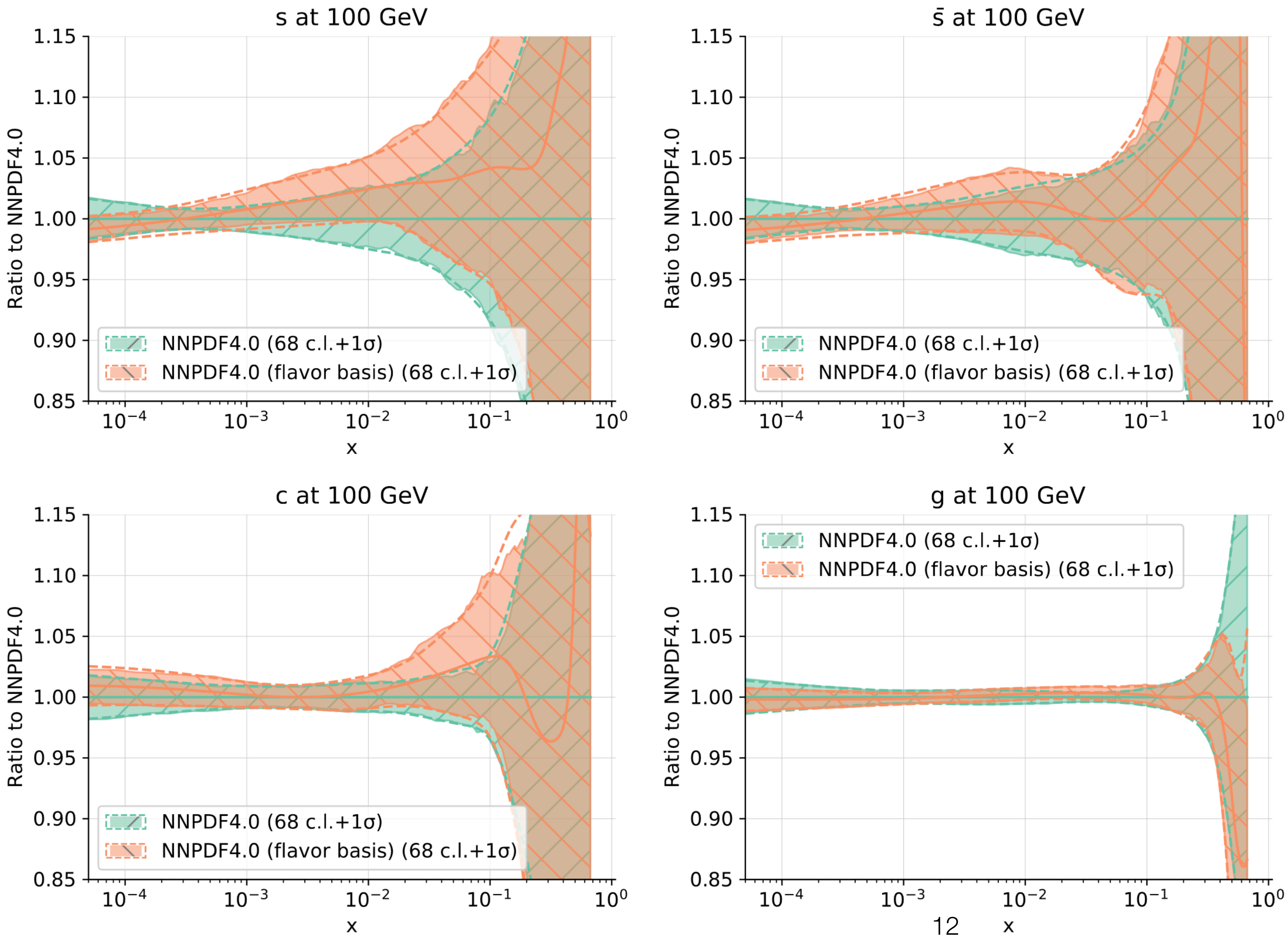
r_n → $\mathrm{NN}(\theta_n)$ → PDF_n

Generate Replicas of the Datasets





Parametrisation Basis Independence



- Practically, we **do not expect** the two basis to be trivially identical: **completely different methodology** (e.g. flavour basis do not have small-x preprocessing exponents, etc.)
- Due to its complexity (especially regarding **Regge behaviour** at small-x), flavour basis require much **bigger architecture**
- The two basis are in **excellent agreement**, with differences fully compatible with the uncertainties

2. Hyperparameter Optimisation (HPO)& Distributed Training on High-Performance Computers (HPCs)

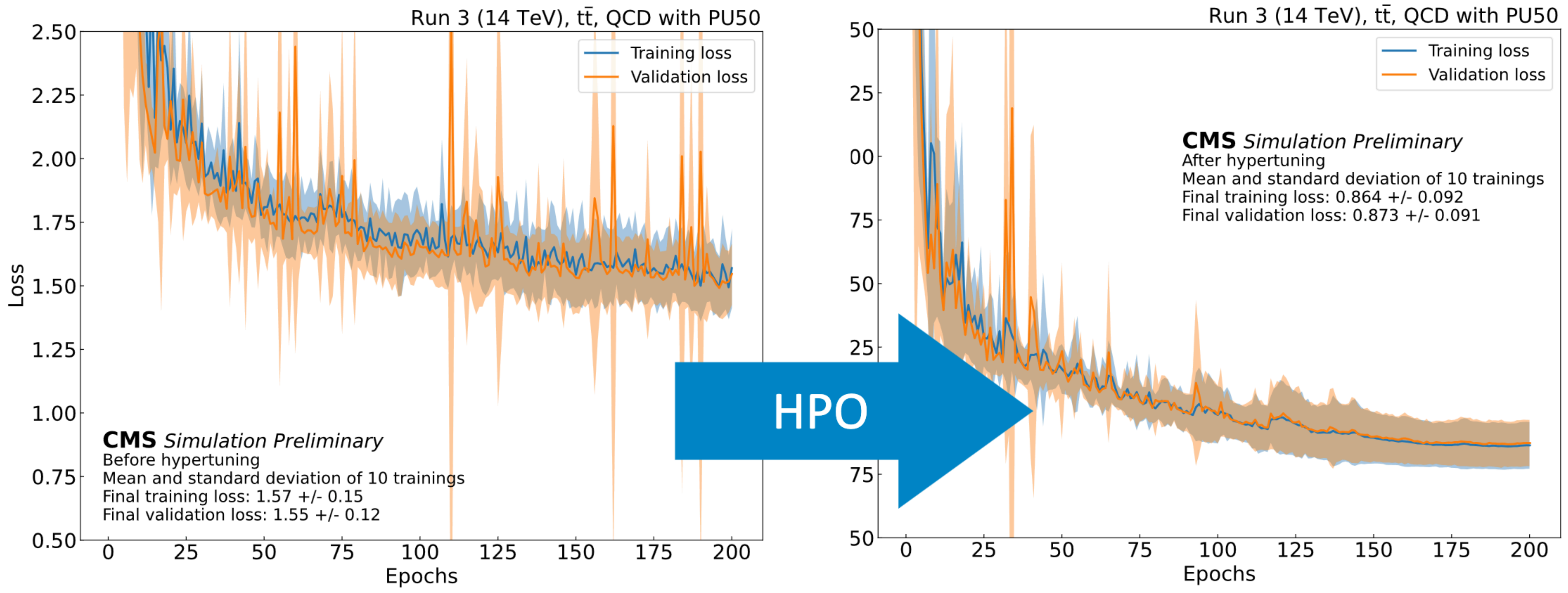
HPO Interlude (1)

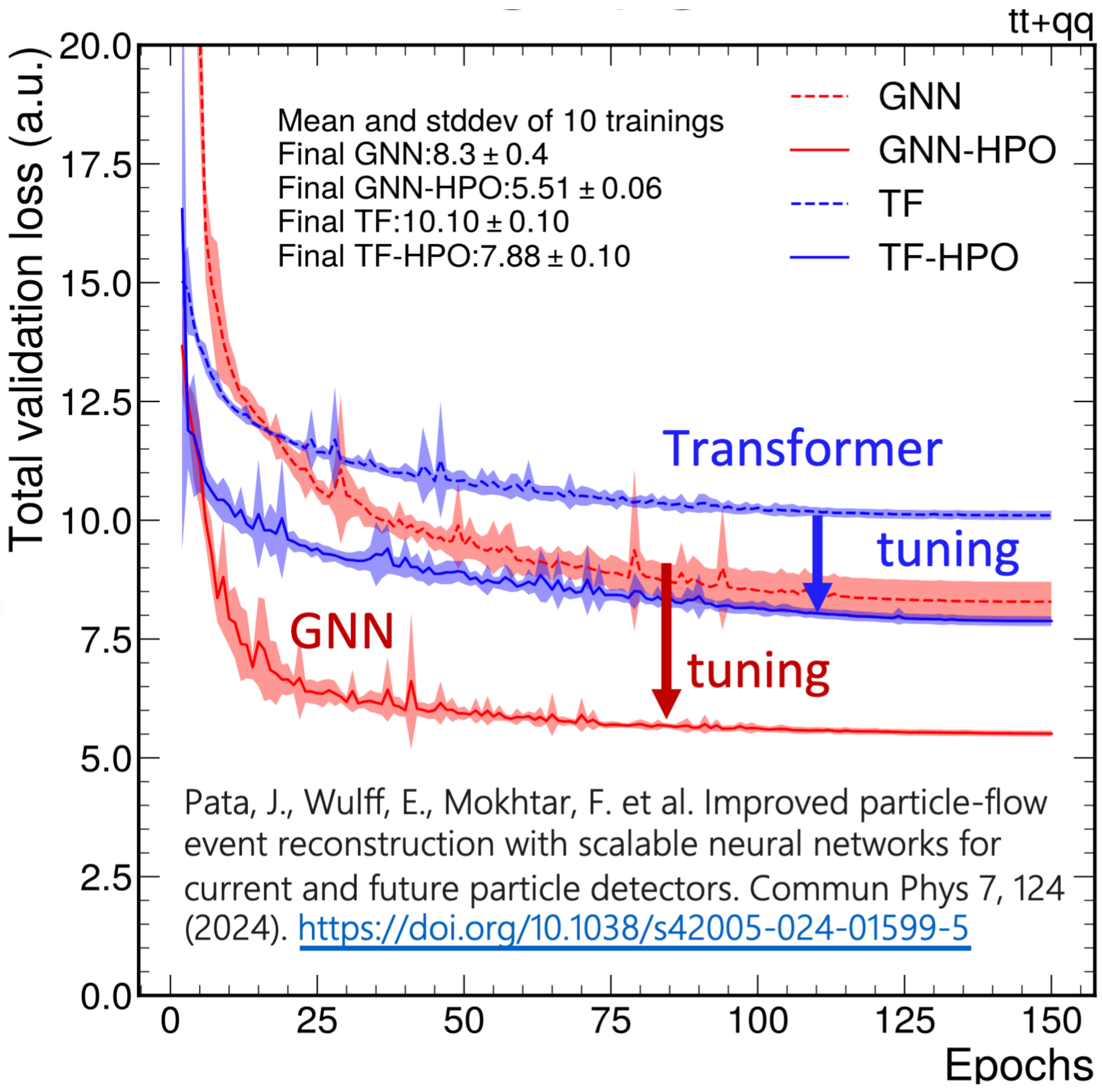
One of the main reasons to resort to Neural Network (NN) was **to reduce biases** in defining a functional form, however:

❌ The hyperparameters θ that define the NN have to be chosen

❌ Random and/or Manual selection of hyperparameters θ are tedious and not guaranteed

✅ Perform an **automated scan of the search space** by running fits with thousands of hyperparameter combinations using a **suitable metric** ‼️





HPO Interlude (2)

One of the main reasons to resort to Neural Network (NN) was **to reduce biases** in defining a functional form, however:

❌ The hyperparameters θ that define the NN have to be chosen

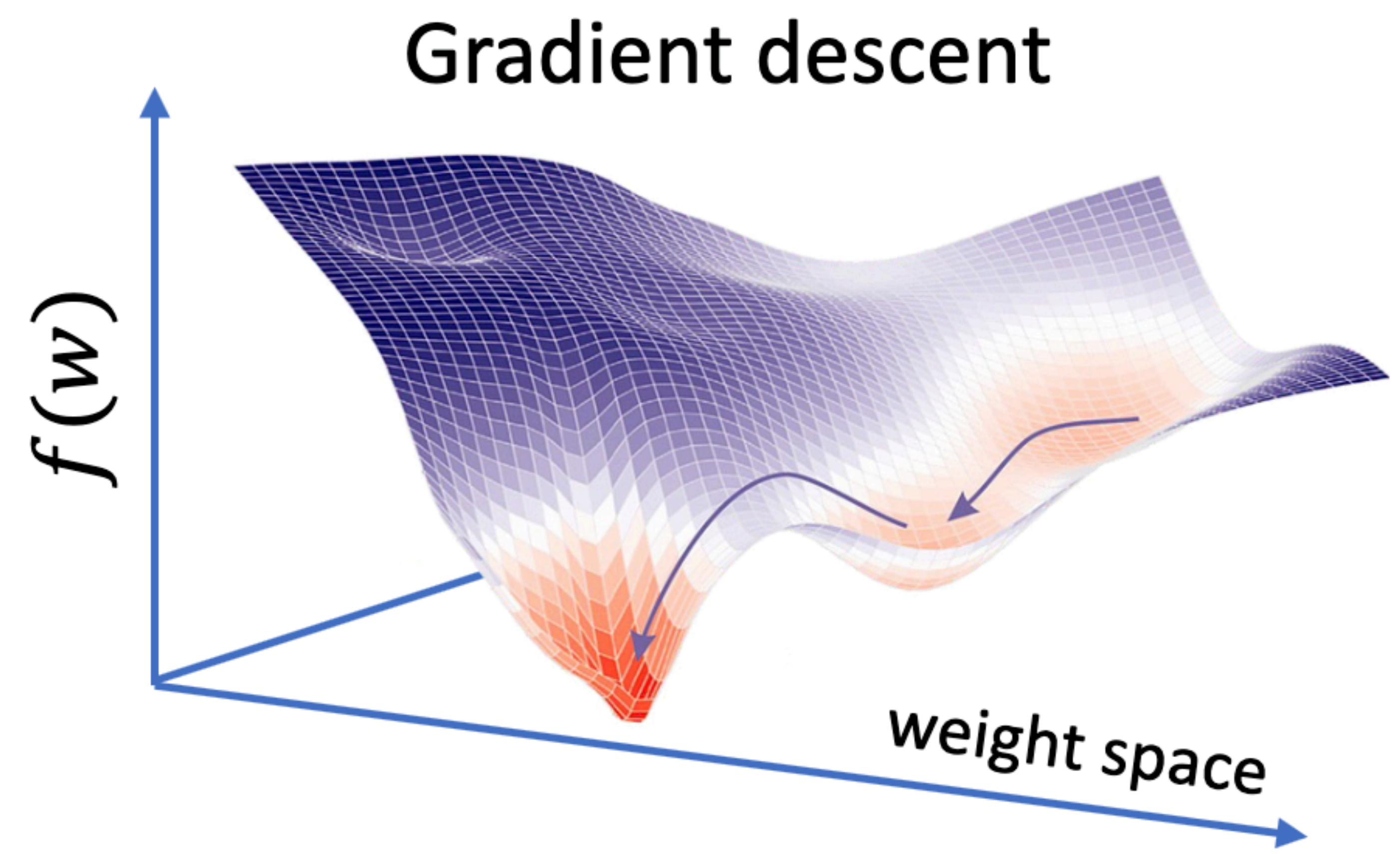
❌ Random and/or Manual selection of hyperparameters θ are tedious and not guaranteed

✅ Perform an automated scan of the search space by running fits with thousands of hyperparameter combinations using a **suitable metric** **!!**

HPO of an **"Objective Function"** $f(w,\theta)$ happens in **TWO** stages:

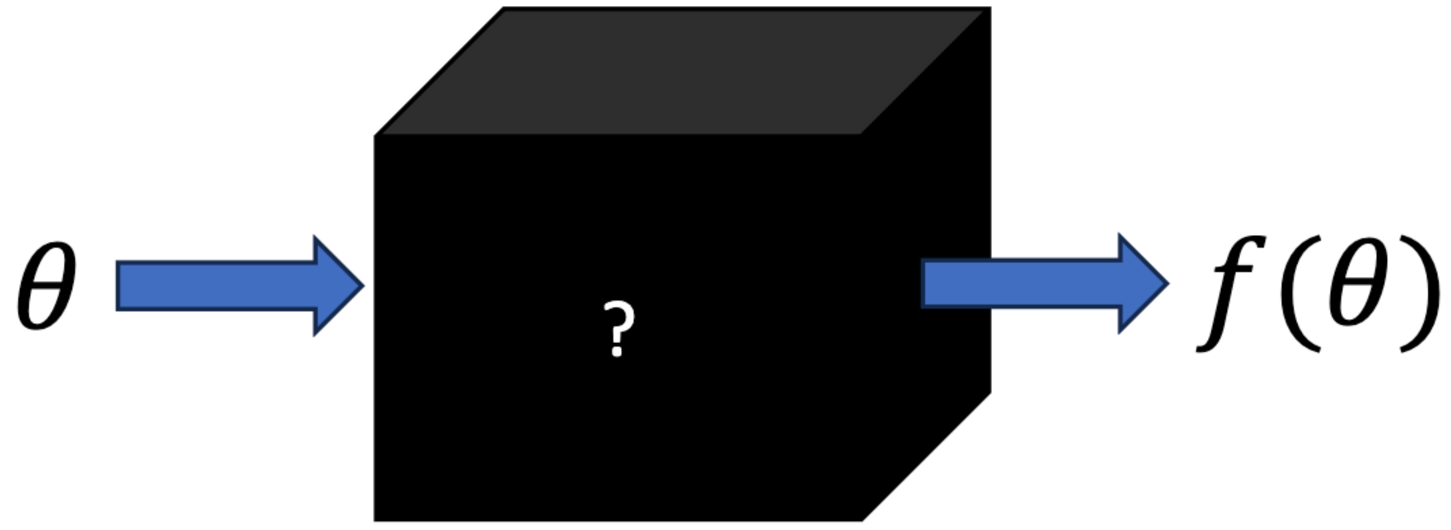
- **Training:** Optimising f w.r.t. w via Gradient Descent $\Longrightarrow$ Search for $w^* = \arg\min_{w} f(w,\theta)$
- **HPO:** Optimise f w.r.t. θ via hyperparameter scan $\Longrightarrow$ search for $\theta^* = \arg\min_{\theta} f(w,\theta)$

$f(w,\theta)$ is differentiable w.r.t. w



$f(w,\theta)$ non-differentiable w.r.t. θ

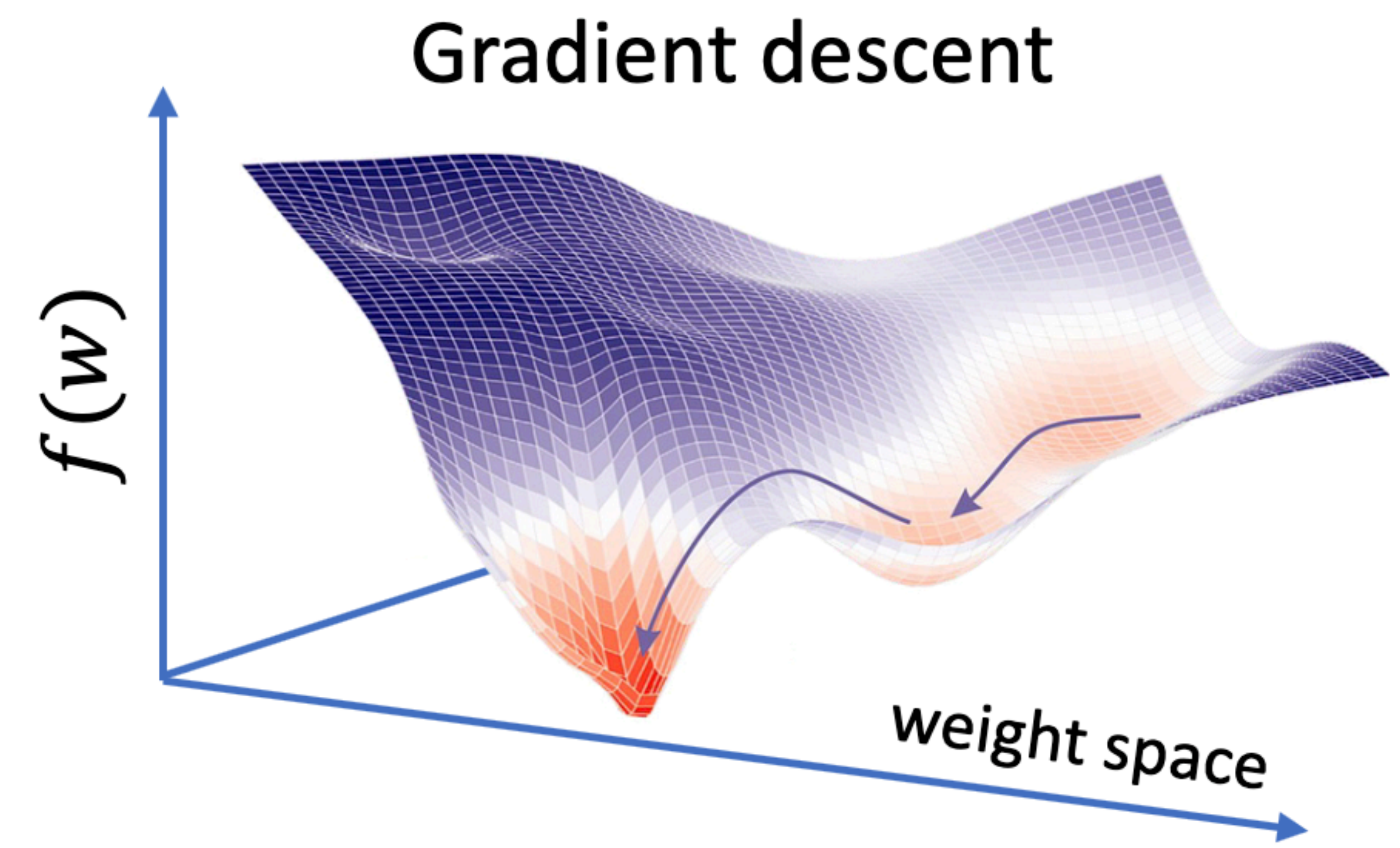
- Black-box optimization
- No straightforward update rule for θ



HPO Interlude (3)

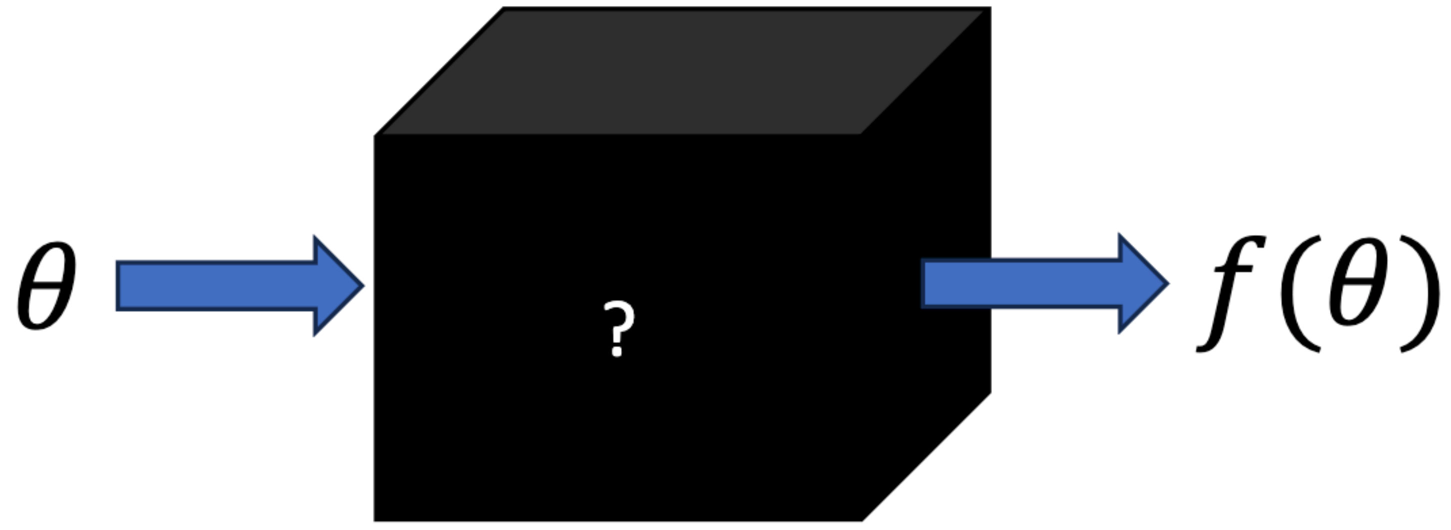
- **Problem Statement:** we want to search for $\theta^* = \arg\min_{\theta} f(w, \theta)$ by querying values of f in a given hyperparameter space; but not computing its Gradient w.r.t. θ
  - w are the learnable weights of the DNNs
  - θ is the set of hyperparameters (HPs)
- Depending on the complexity of the Problems, the objective function f **itself can be very expensive to evaluate**
- HPO is **extremely compute-resource intensive**:
  - Require **efficient & smart search algorithms**
  - Benefits greatly from **HPC resources**

$f(w, \theta)$ is differentiable w.r.t. w

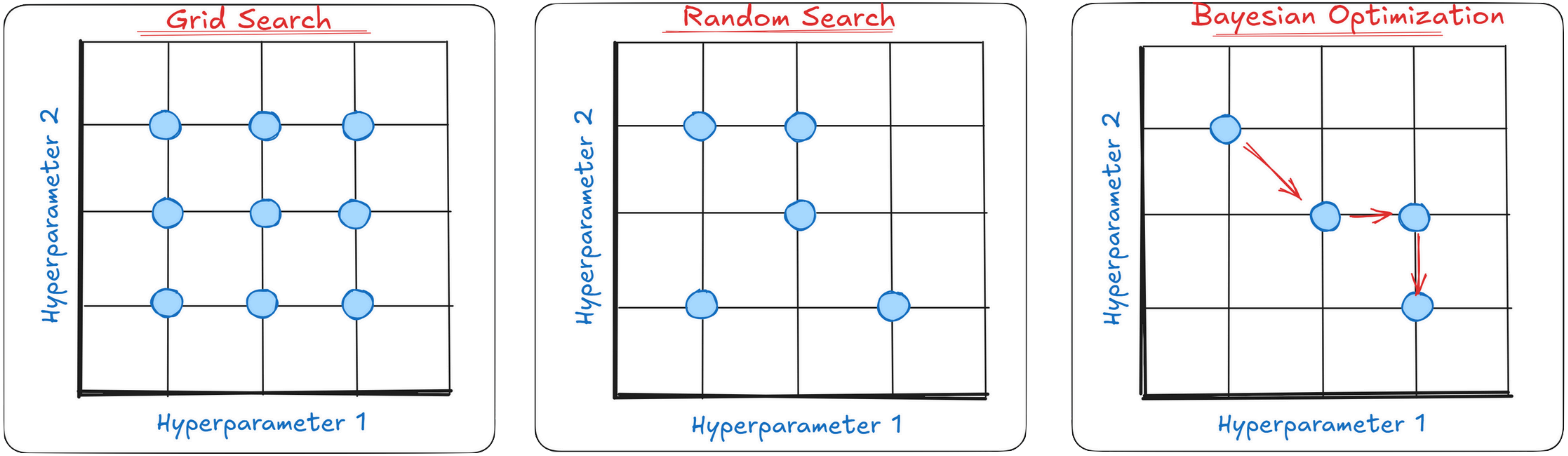


$f(w, \theta)$ non-differentiable w.r.t. θ

- Black-box optimization
- No straightforward update rule for θ

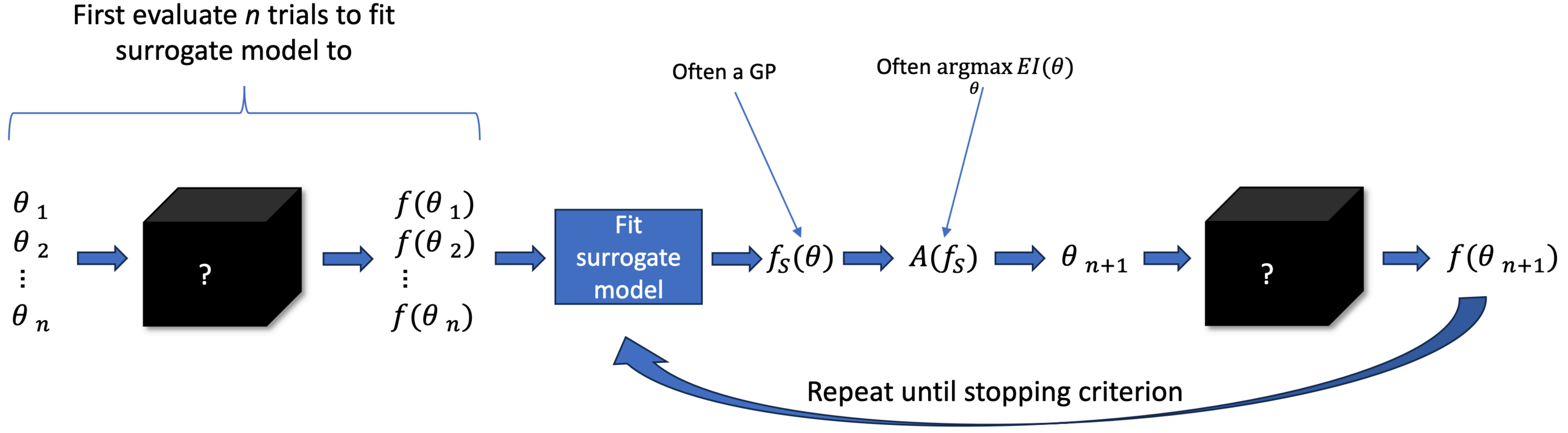


Automatic HPO - Search Algorithms



- **Static Search Algorithms: Grid Search** (Deterministic, Exhaustive Search on the Grid), **Random Search** (Stochastic, Exhaustive search on the random points)
- **Adaptive & Model-Based Search Algorithms: Evolutionary Search**, **Bayesian Optimisation Search** (Surrogate Model: estimates $f(\theta)$ and its uncertainties given some HPs, Faster convergence)

Bayesian Optimisation: Tree-Parzen Estimator



Tree Parzen Estimator (TPE) Bayesian Optimisation: The goal of an optimisation algorithm is to find the combination of hyperparameters such that $\theta^\star = \arg\min_\theta f(\theta)$.

$$\left\{\left(\theta_j, f_j\right)\right\}_{j=1}^{n} \longrightarrow p(\theta \mid f) = \begin{cases} \ell(\theta) & \text{if } f < f^* \\ g(\theta) & \text{if } f \geq f^* \end{cases} \longrightarrow \max(\text{EI}) \sim \max\left(\frac{\ell(\theta)}{g(\theta)}\right)$$

Adapted from Eric Wulff's slides @CERN OpenLab

HPO for PDF Determination: NNPDF4.0 Approach

✅ Perform an automated scan of the search space by running fits with thousands of hyperparameter combinations using a **suitable metric** ‼

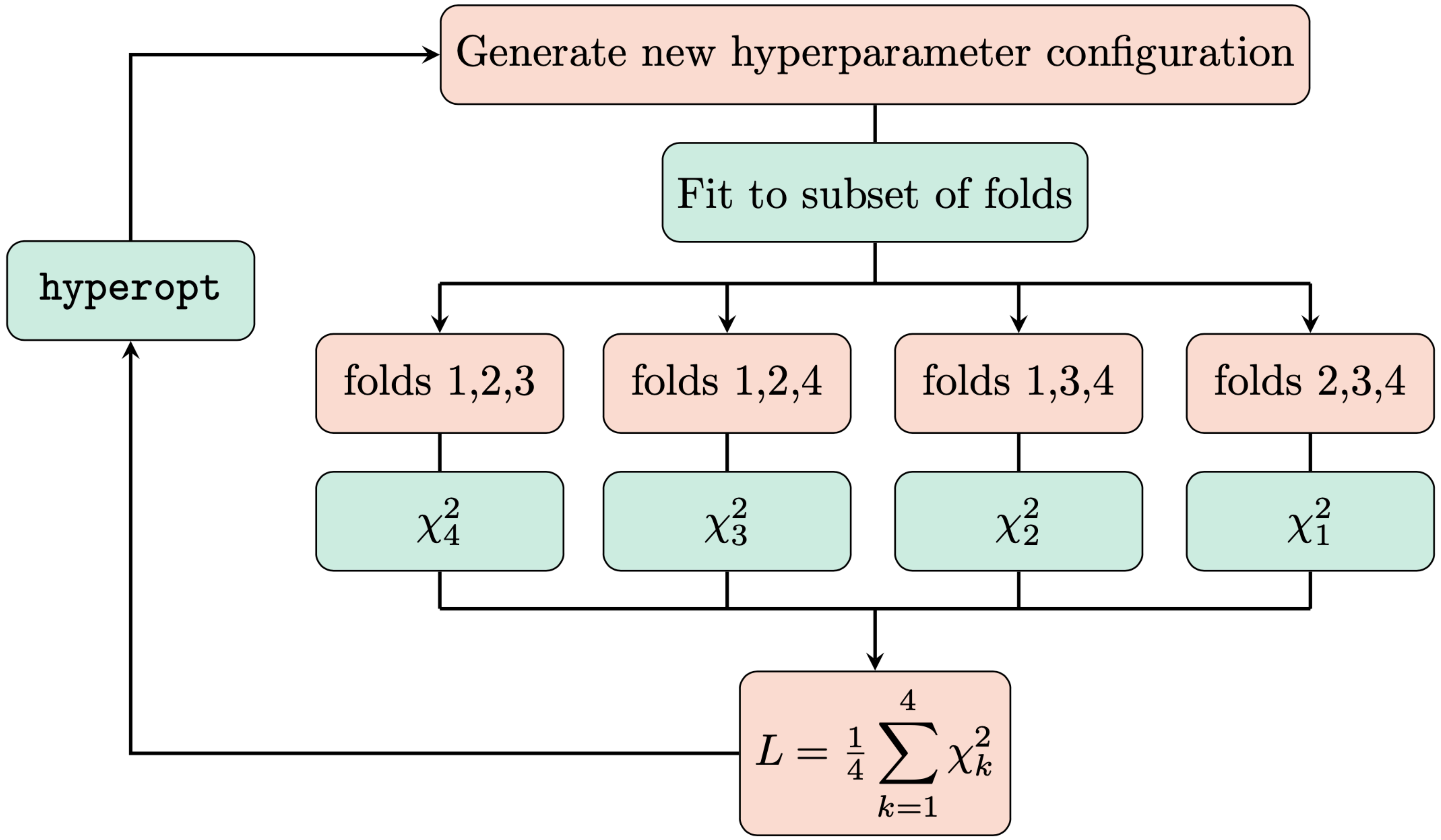
"When a measure becomes a target, it ceases to be a good measure" Goodhart's law

The choice of **figure of merit is crucial in obtaining a "Good Fit"** (smoothness of the PDFs, generalisation power to future experimental data, time/iterations it takes to complete a fit, etc.)

In NNPDF4.0, the figure of merit is defined in terms of **k-fold cross validation method**. For each hyperparameter configuration, we run 4 fits to the **central** experimental data, and in each of these fits, **the n-th fold is left out.**

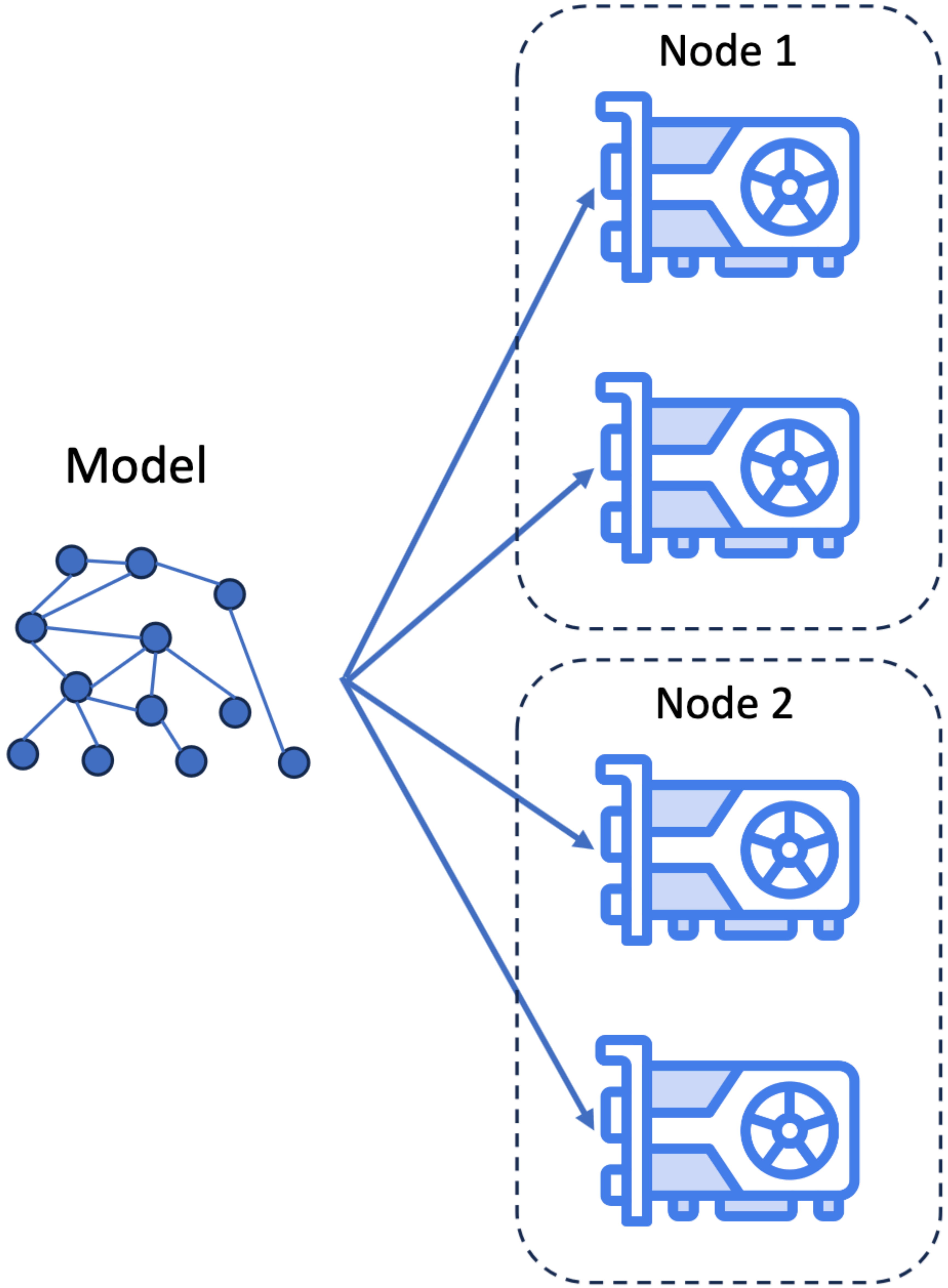
The metric is then defined as THE MODEL that yields the "BEST" k-fold Loss

Is this the Best we can do?



The need for HPCs?

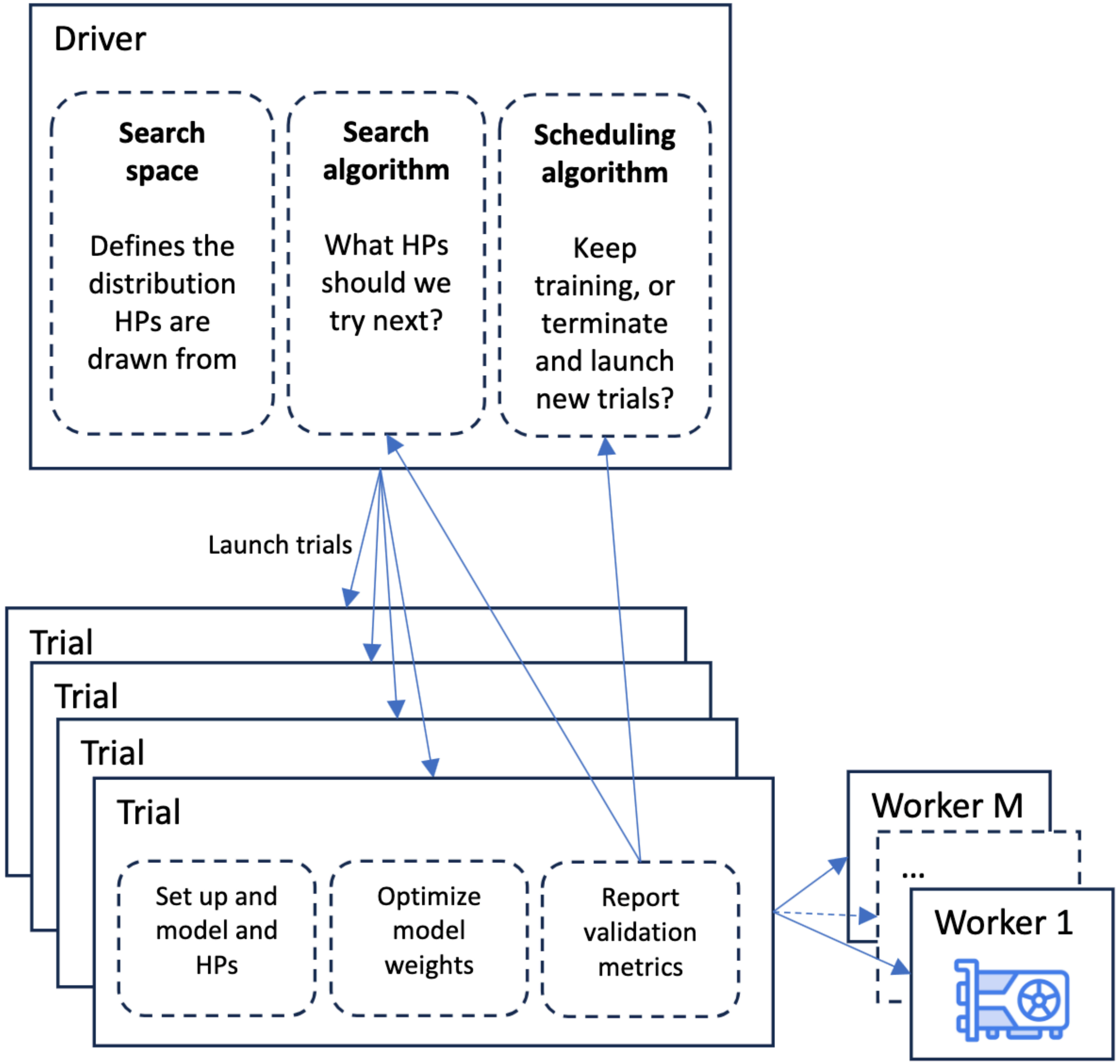
- Monte Carlo (MC) PDF fits require **hundreds to thousands** of independent replica trainings
- Each fit is **computationally expensive**: theory evaluations, χ^2 minimisation, NN training
- HPOs **multiplies the total number of required runs**
- Replica fits are **naturally parallel** and ideal for distributed computing
- GPUs accelerate training, while HPC systems scale across many nodes/devices
- Faster turnaround enables rapid methodology development and validation studies
- Precision PDF physics **increasingly depends on advanced computing** infrastructure



Distributive HPO Workflows

- **Driver**:
  - Define the hyperparameter search space to evaluate f
  - Generates hyperparameter trial configurations
  - Launches, tracks, and terminates model trials

- **Hyperparameter trials**:
  - Sets up model trials and hyperparameter configurations
  - Perform the actual Neural Network training
  - Reports metric back to “Driver”

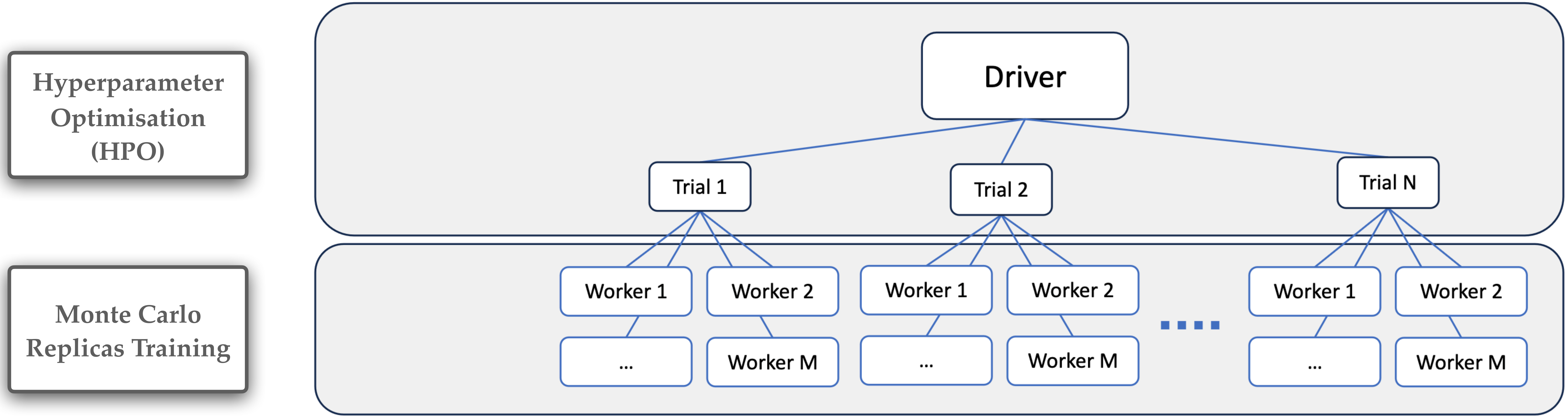


Adapted from Eric Wulff’s slides @CERN OpenLab

Distributed Training

- What is Distributive Training?
  - Training models across multiple devices/cluster nodes
  - Enables scaling to larger models & hyperparameter space
- Why it is Relevant?
  - Faster training times & Scanning of the HP space
  - Overcome the memory constraints

For PDF determination using MC approach, there are two levels or parallelisation:



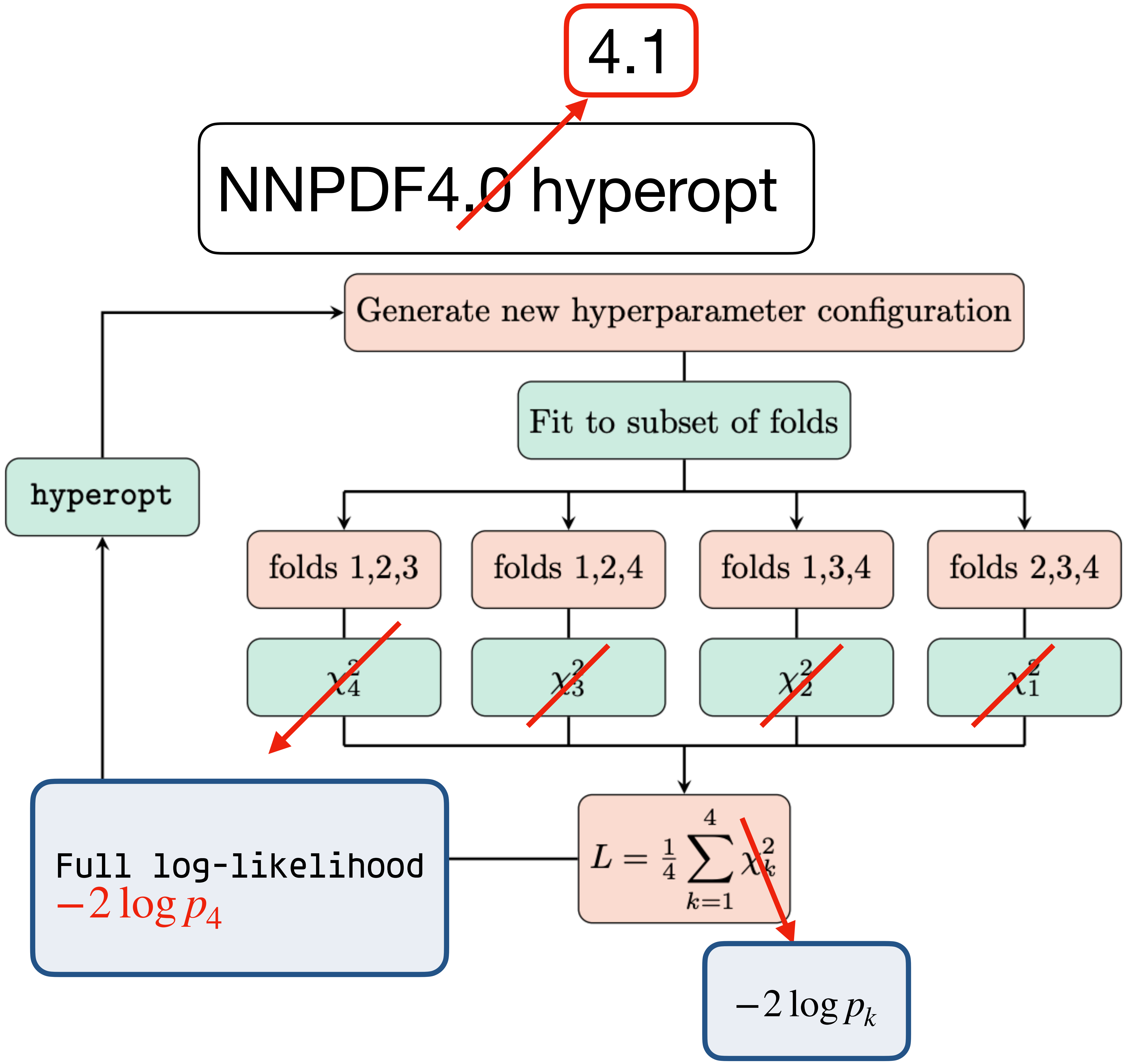
Snapshot of Distributed HPO on HPC Systems

- HPO Jobs are submitted through a Scheduler (e.g. SLURM) by a user
- Head node generates hyperparameter trial configurations
- Nodes train the model trials in parallel

Adapted from Eric Wulff’s slides @CERN OpenLab

3.HPO for PDF Ensemble

HPO Metric for PDF Ensemble



- The full dataset is partitioned in different folds
- Figure of merit is defined as the ~~χ^2~~ of the excluded folds

Full log-likelihood

- Minimise the figure of merit and ~~select the best HPs configurations~~

Select an ensemble of samples maximising the likelihood

Log-likelihood Metric Definition

$$p_k(\theta) = \frac{1}{N(\theta)} \exp\left[-\frac{1}{2} \left(y^{(k)} - \overline{T}^{(k)}(\theta)\right) \left(C_y^{(k)} + C_{PDF}^{(k)}(\theta)\right)^{-1} \left(y^{(k)} - \overline{T}^{(k)}(\theta)\right)^T \right]$$

Central values of fold k

Experimental covmat for fold k

Theory prediction for fold k, computed using a fit which does not include fold k itself

PDF error for fold k, computed using a fit which does not include fold k itself

Normalisation of the likelihood

$$\sqrt{\det 2\pi \left(C_y^{(i)} + C_{\mathrm{PDF}}^{(i)}(\theta)\right)}$$

HPO on PDF Ensemble using GPUs

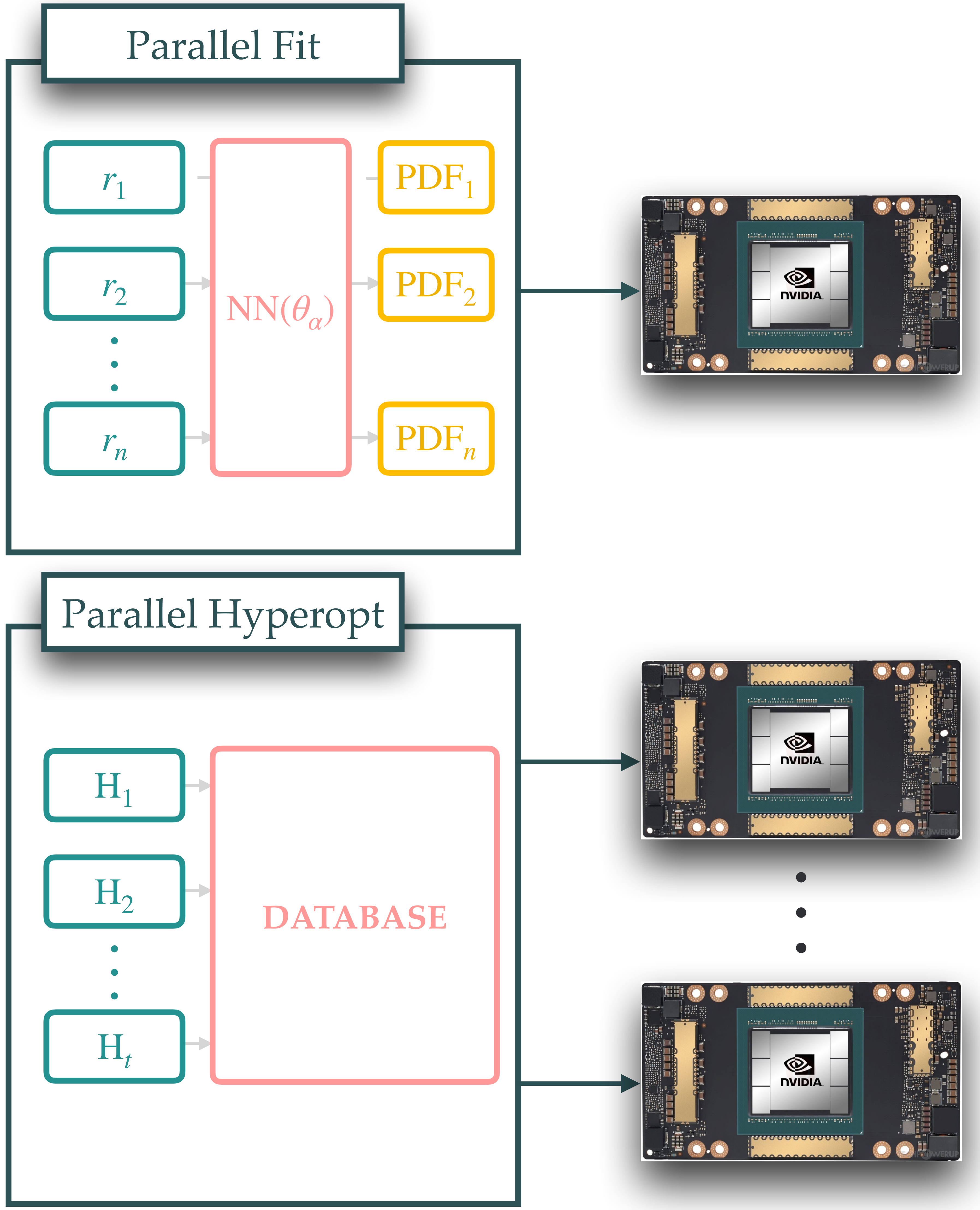
netherlands eScience center

Significant improvements **on two main fronts**, namely the hyperparameter optimisation procedure and at the level of the fits themselves.

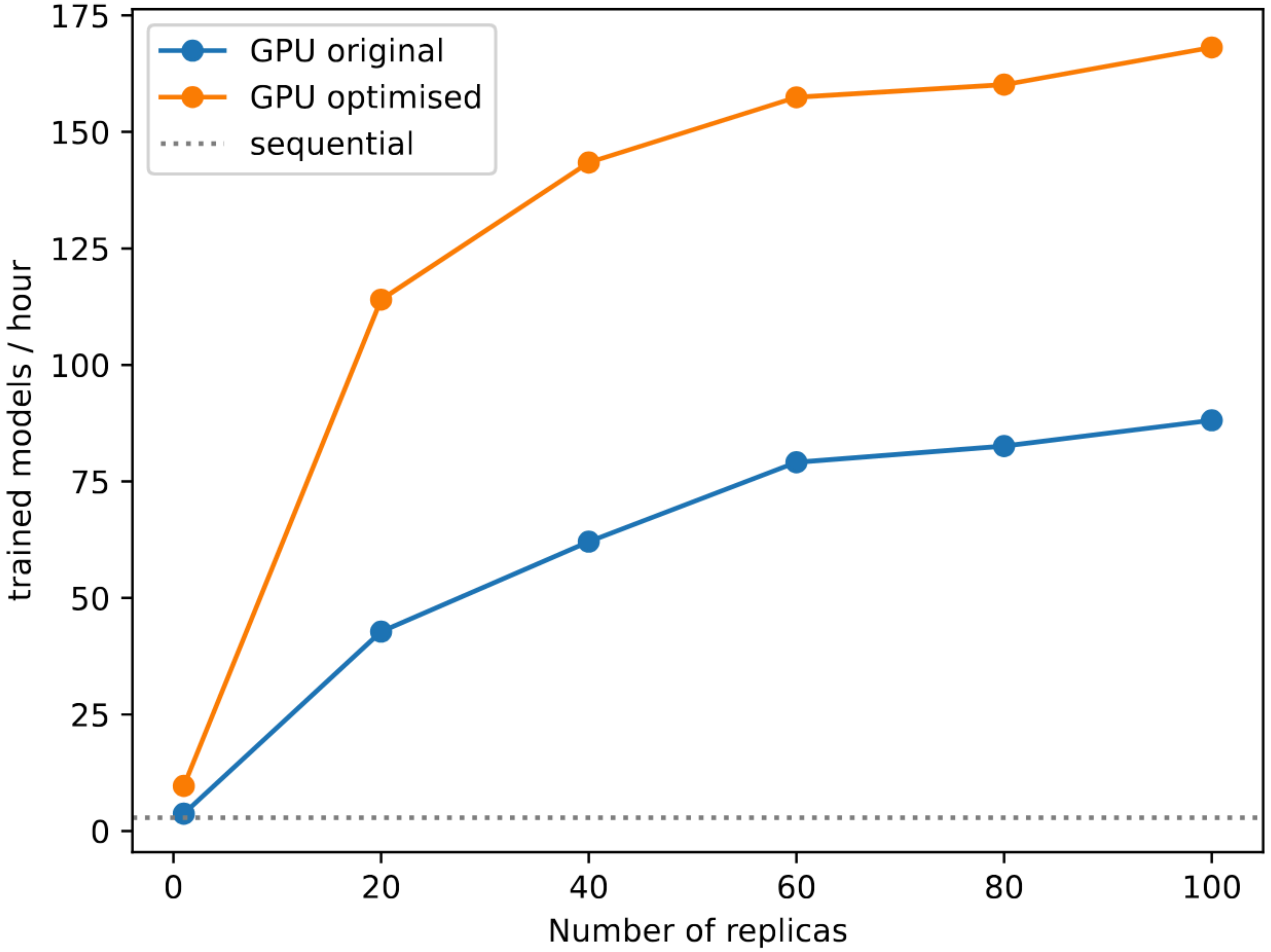
✦ **Simultaneous fit of multiple replicas:**
- Tensorflow allows the exact same codebase to be used for both CPU and GPU
- Redesign of the framework in order to share memory-heavy objects across all the replicas
- Resort to single PDF neural network model

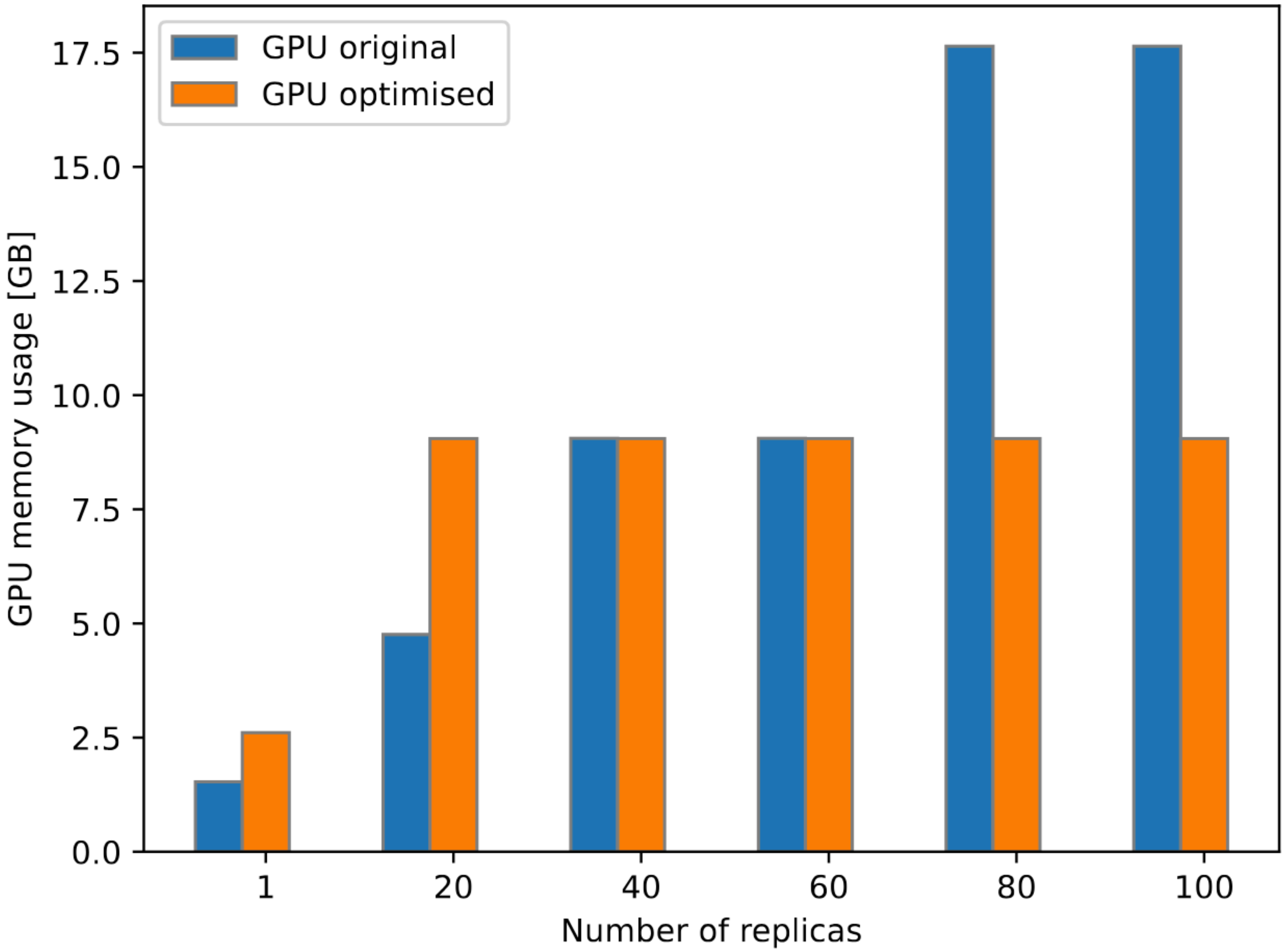
⟹ **Running ~150 replicas at once on a A100 Nvidia GPU is now as fast as a fit of one single replica.**

✦ **Distributed asynchronous Hyperparameter Optimisation:**
- Evaluate trials in parallel across many different GPUs
- Each instance of the worker shares the same database (MongoDB)



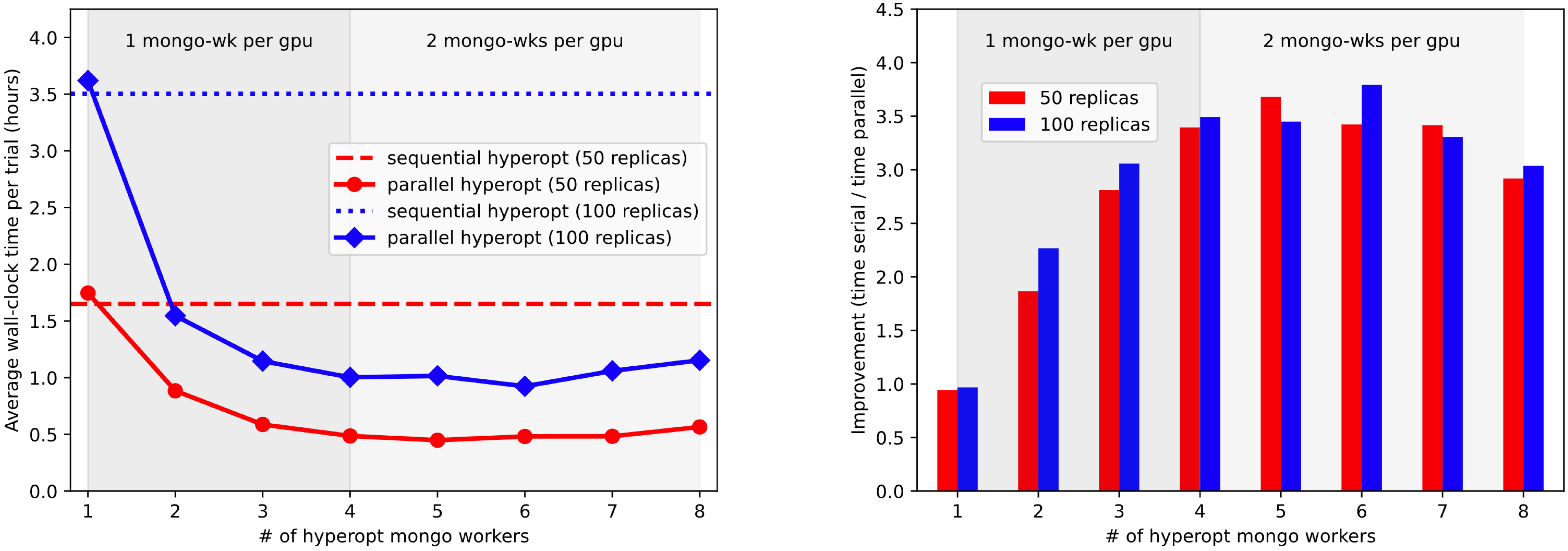
Scaling of MC replicas training on a GPU





# Replicas	10	50	100
Energy reduction	78%	87%	91%

Scaling of trials across multiple GPUs



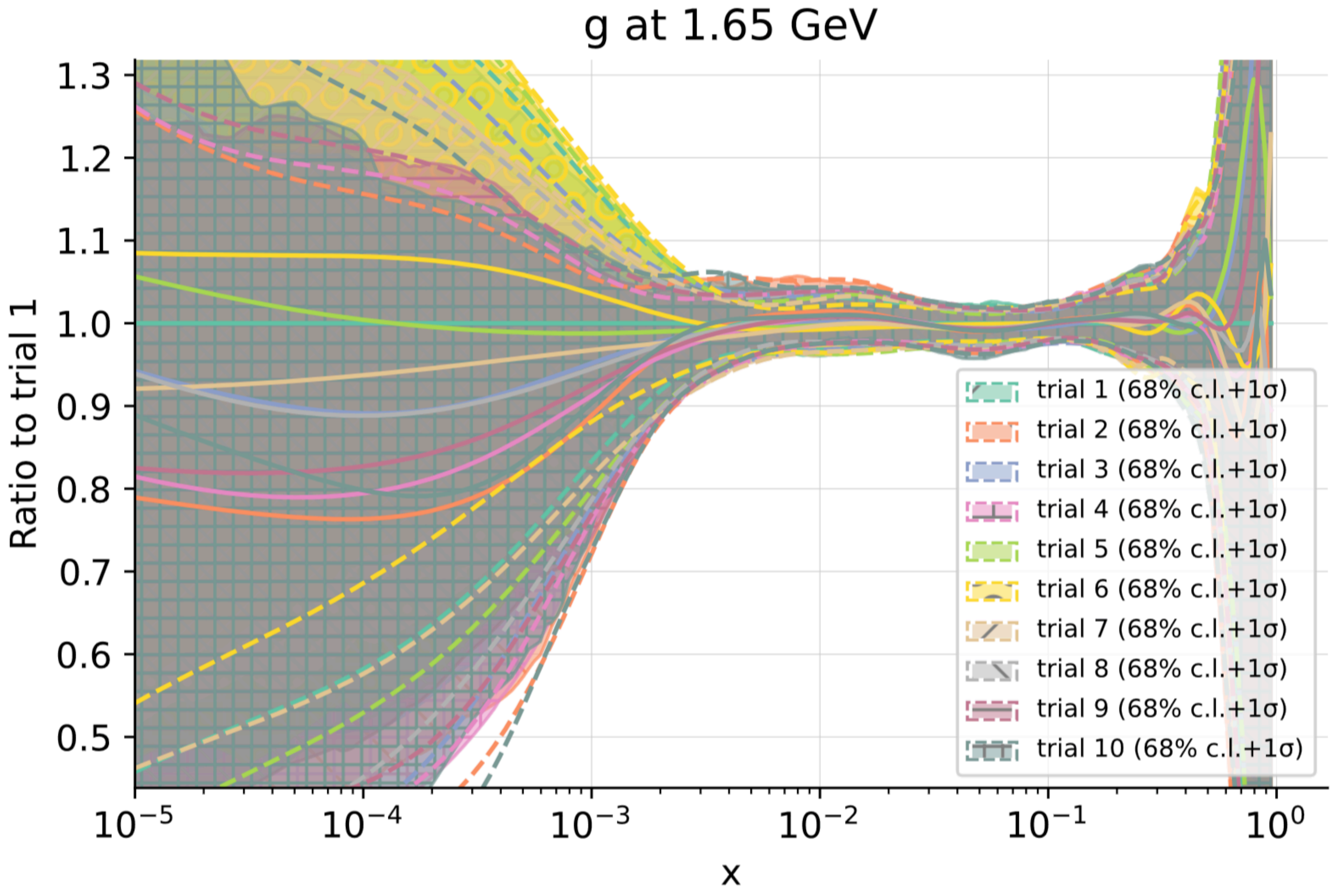
HPO Evolution

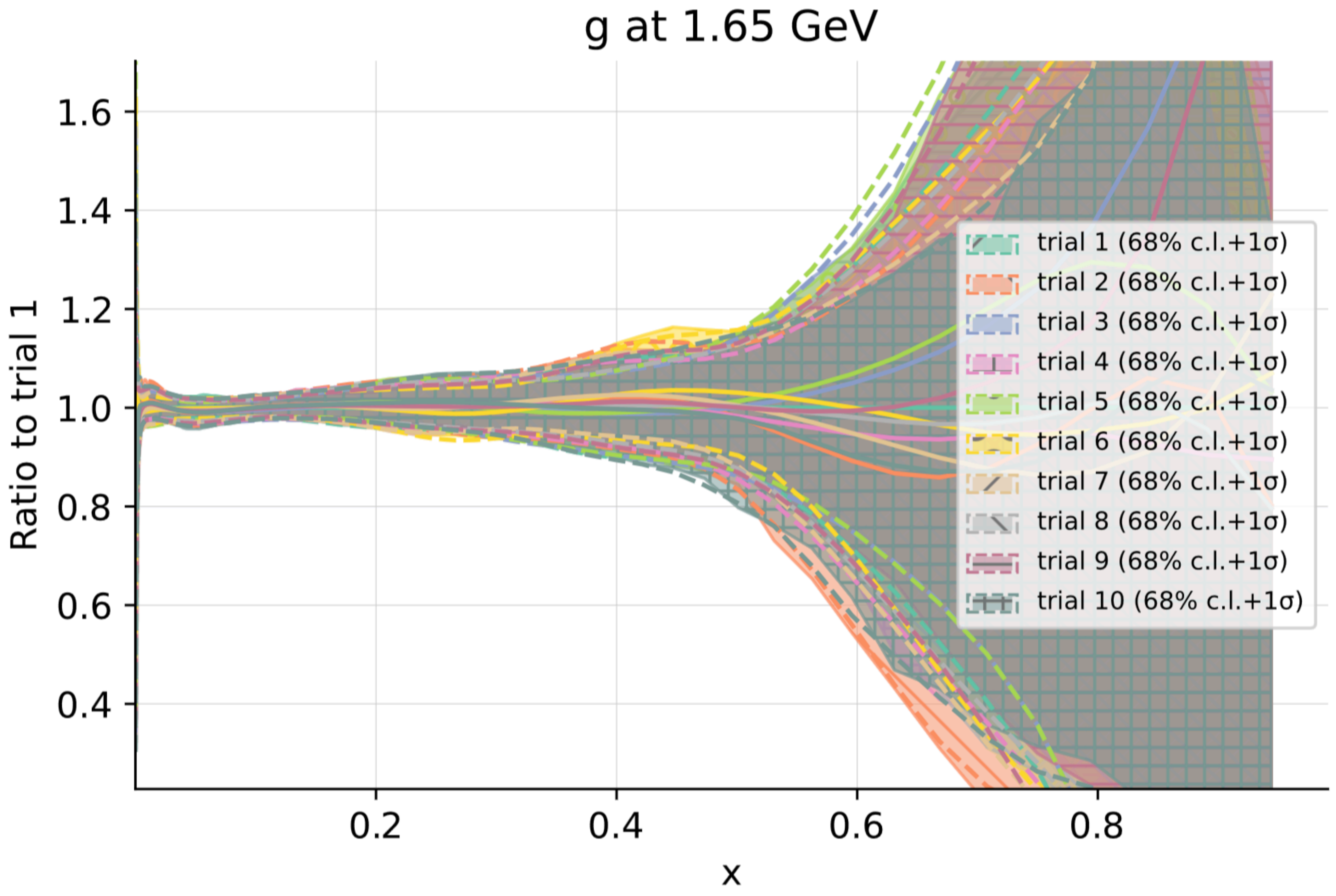
Thermalisation Transition

Highlights of Best Trials

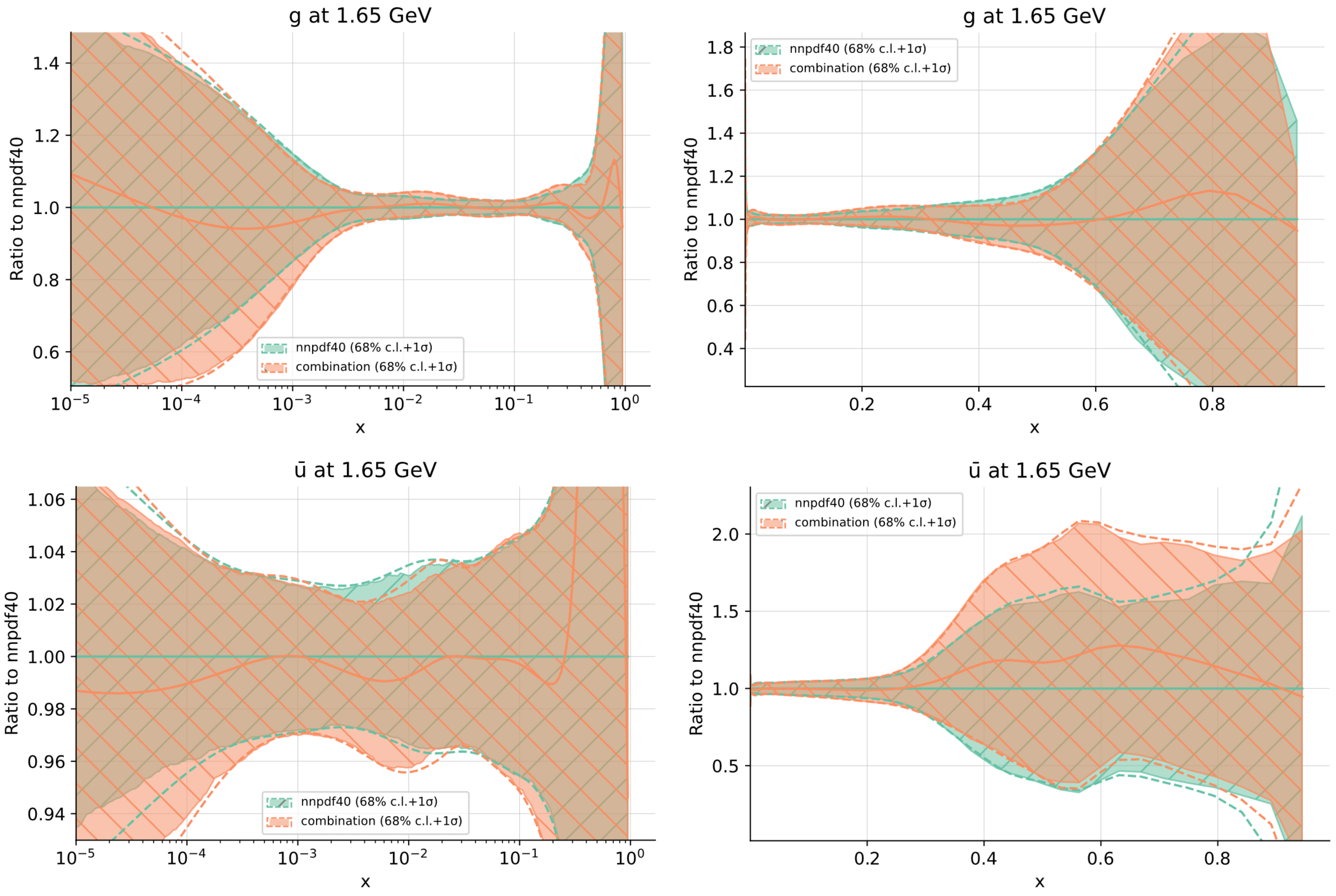
Trial	Architecture	Activation	Optimizer	Learning rate	Clipnorm	Epochs
925	[15, 16, 19, 8]	tanh	Adam	0.00618	5.98e-7	18704
550	[13, 16, 18, 8]	tanh	Adam	0.00816	4.67e-7	20089
870	[15, 16, 18, 8]	tanh	Adam	0.00785	6.06e-7	19298
760	[22, 16, 16, 8]	tanh	Adam	0.00763	8.96e-7	17993
859	[15, 17, 17, 8]	tanh	Adam	0.00850	4.97e-7	19303
463	[12, 16, 19, 8]	tanh	Adam	0.00979	1.93e-6	18683
729	[19, 17, 17, 8]	tanh	Adam	0.00695	6.21e-7	18880
666	[12, 12, 17, 8]	tanh	Adam	0.00491	8.02e-7	22204
898	[24, 15, 19, 8]	tanh	Adam	0.00891	4.65e-7	19564
438	[14, 15, 19, 8]	sigmoid	Adam	0.01000	1.95e-6	19737

Despite **completely different architecture, optimisers, etc.**, the best trials yield PDF distributions that are in **excellent agreement** with the **differences well within the uncertainties**





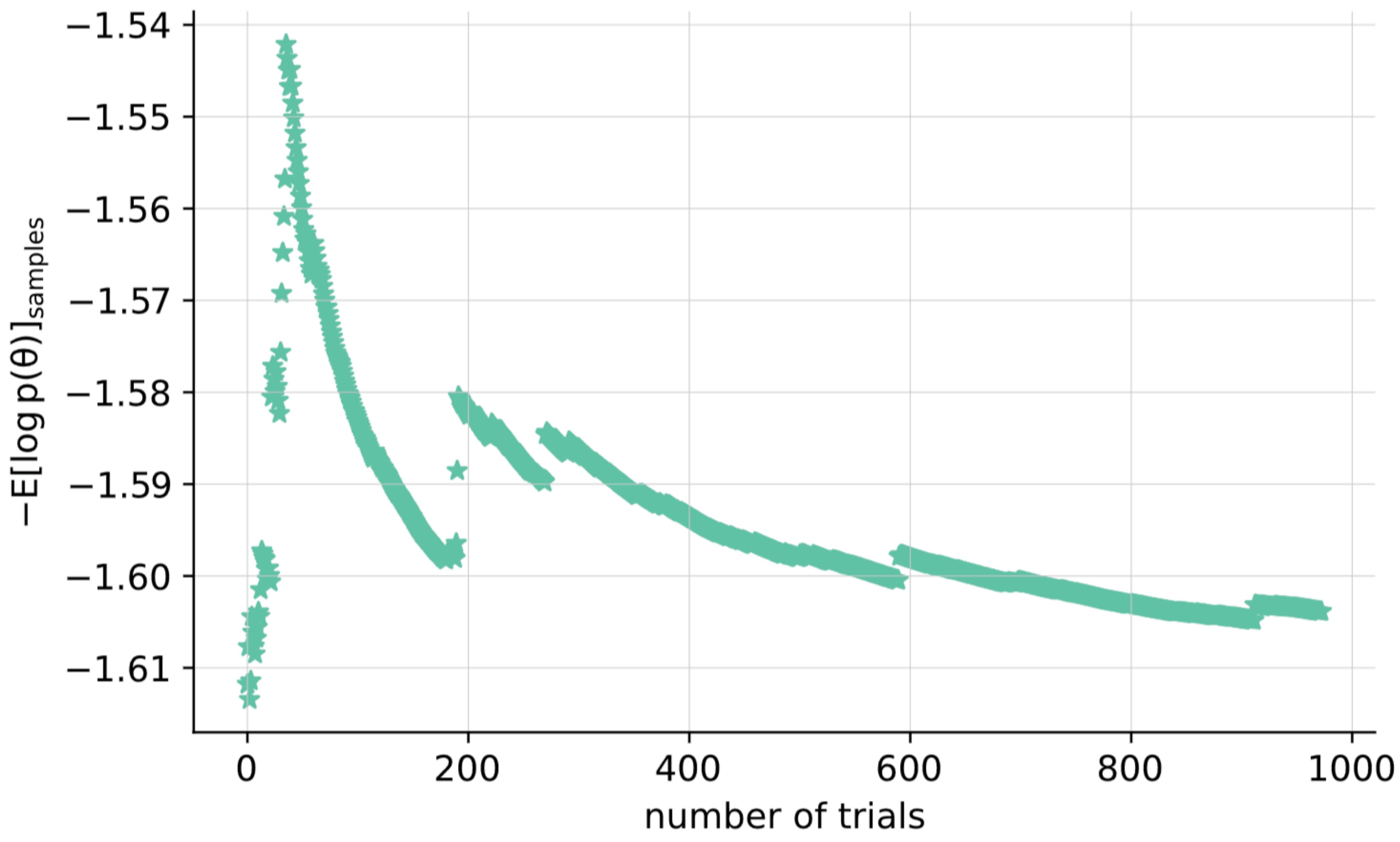
Results at the level of PDF Distributions

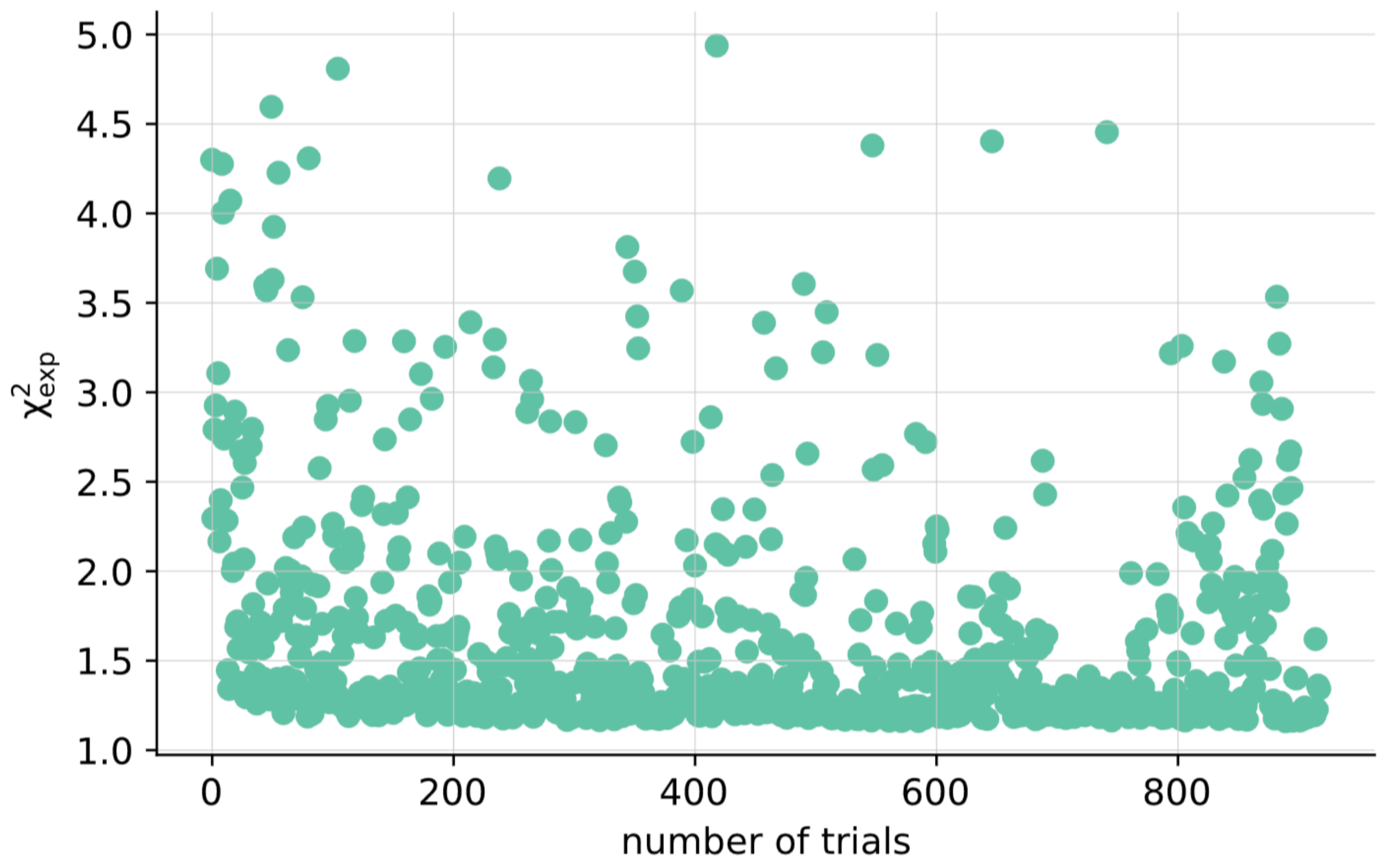


- Samples from the best trials are used to construct a **combined** PDF distribution

- The combined results are **consistent with the published baseline NNPDF4.0** but with **slightly larger** uncertainties (in the extrapolation regions)

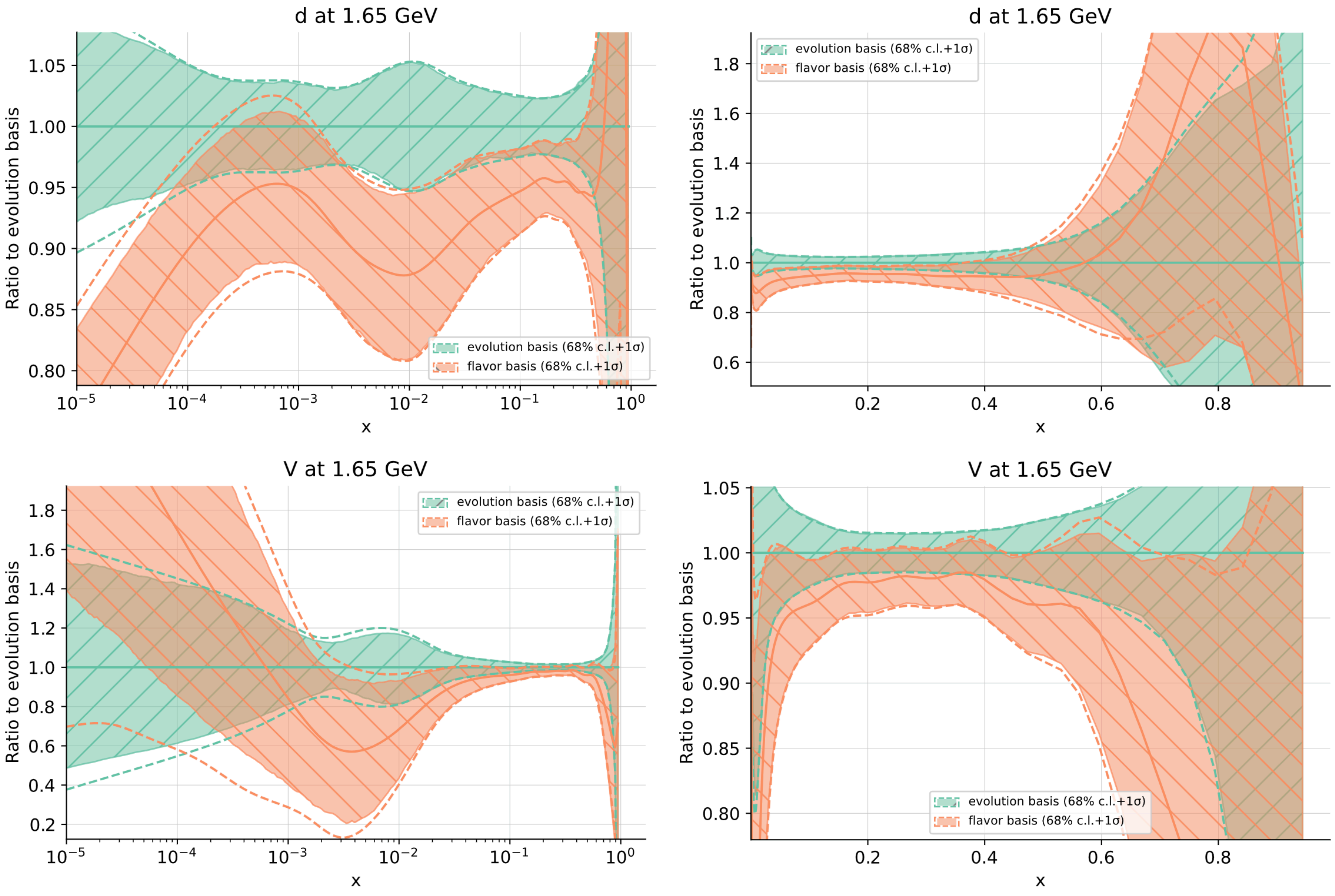
HPO in the Flavour Basis?





Training in the flavour basis shows a **larger fraction of non-convergent trials**, indicating a more **challenging optimisation landscape** and motivating further tuning of architectures and training strategies

PDF Distributions from Flavour Basis Fits

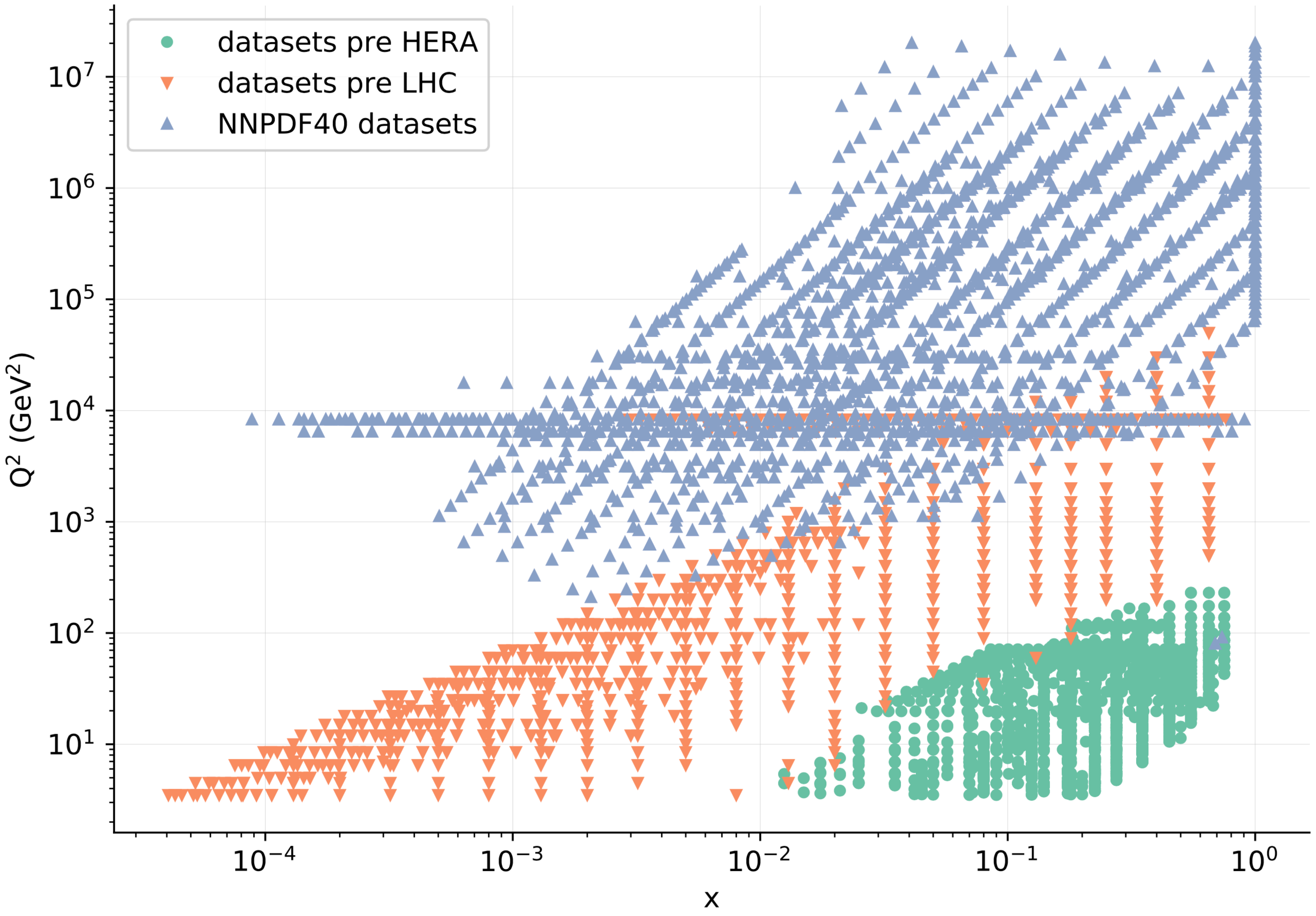


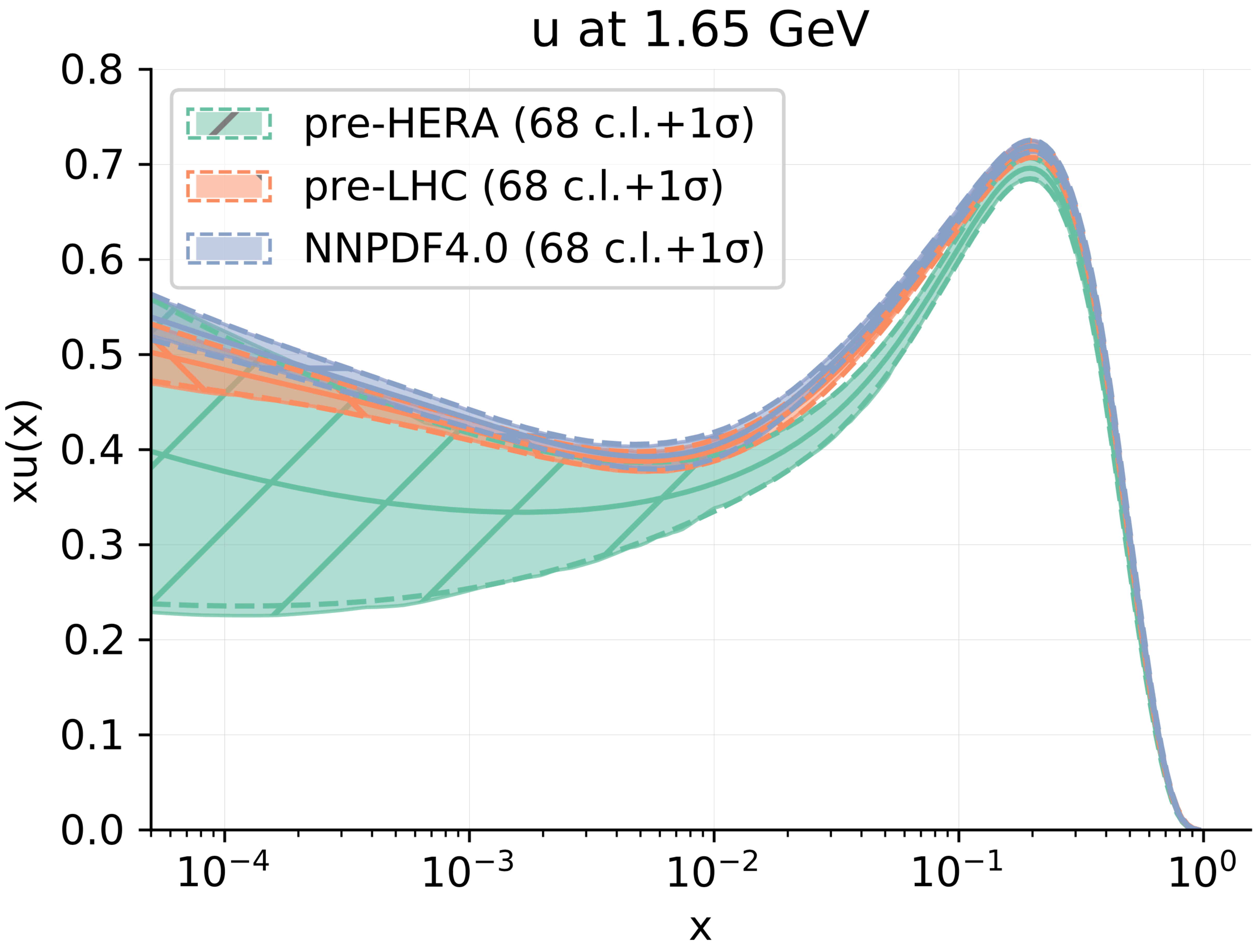
- Flavour basis fits exhibit much **larger uncertainties**
- Central values can **differ** by more than **one-sigma** even in the data regions

3.Uncertainty Validations

Future Tests (1)

Fit Data to specific kinematic regions, and then checks the generalisation (extrapolation) to unseen experimental data:





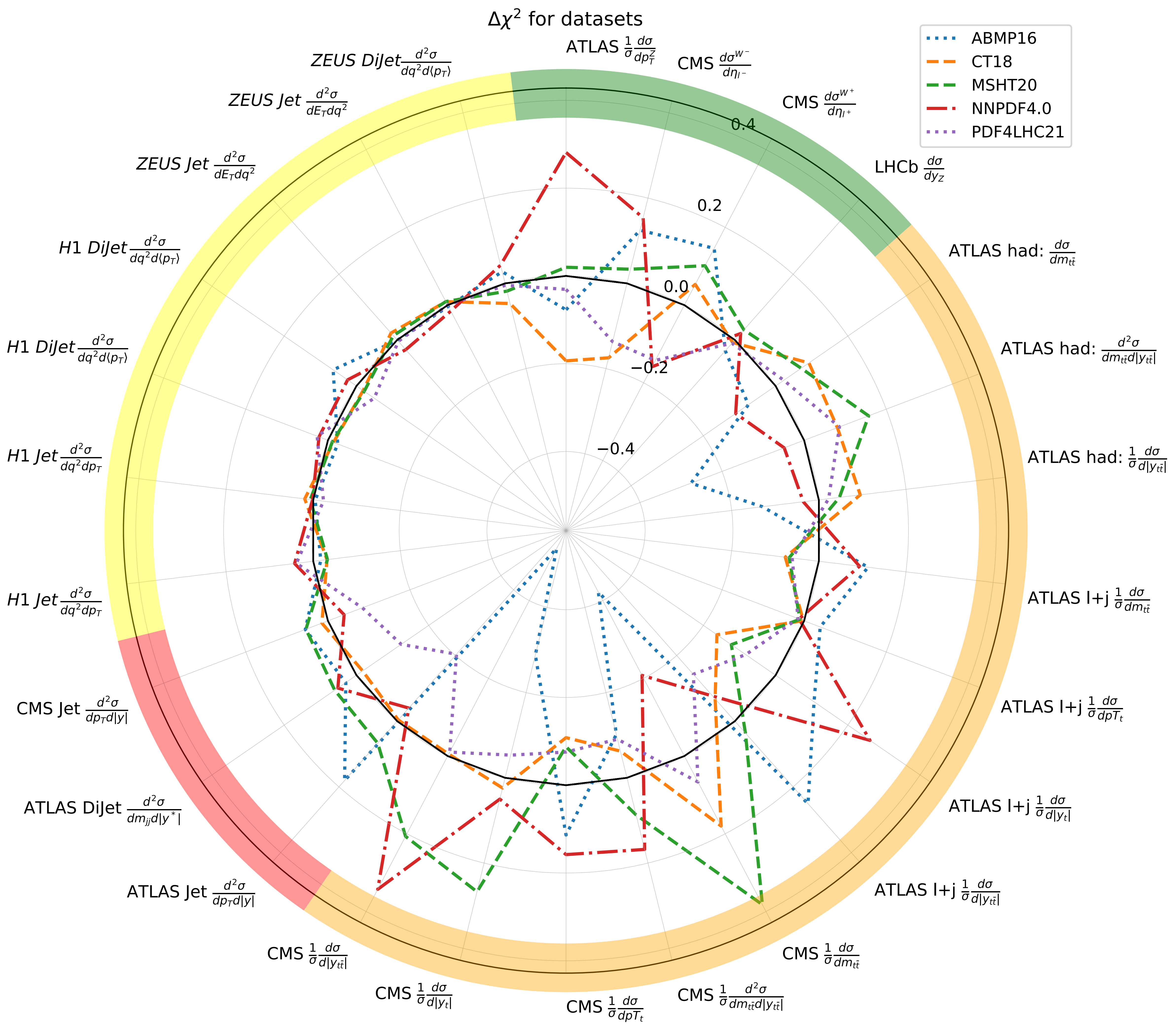
Future Tests (1)

Fit Data to specific kinematic regions, and then checks the generalisation (extrapolation) to unseen experimental data:

χ^2: FITTED VS EXTRAPOLATED: WITHOUT/WITH PDF UNC.

PROCESS	PRE-HERA	PRE-LHC	NNPDF4.0
FT DIS (NC)	1.05	1.18	1.23
FT DIS (CC)	0.80	0.85	0.87
FT DY	0.92	1.27	1.59
HERA	27.20/1.23	1.22	1.20
COLL. DY (TEV.)	5.52/1.02	0.99	1.11
COLL. DY (LHC)	18.91/1.31	2.63/1.58	1.53
TOP QUARK	20.01/1.06	1.30/0.87	1.01
JETS	2.69/0.98	2.12/1.10	1.26
TOTAL OUT OF SAMPLE	19.48/1.16	2.10/1.15	–

Future Tests (2)



- Relative change in the total χ^2 due to a change in the input PDF

$$\Delta\chi^{2(i)} = \frac{\chi^{2(i)}_{\rm exp+th} - \left\langle \chi^2_{\rm exp+th} \right\rangle_{\rm pdfs}}{\left\langle \chi^2_{\rm exp+th} \right\rangle_{\rm pdfs}}$$

where

$$\left\langle \chi^2_{\rm exp+th} \right\rangle_{\rm pdfs} = \frac{1}{n_{\rm pdfs}} \sum_{i=1}^{n_{\rm pdfs}} \chi^{2(i)}_{\rm exp+th}$$

- **No systematic outlier** seen in the data description despite noticeable differences at the level of PDF

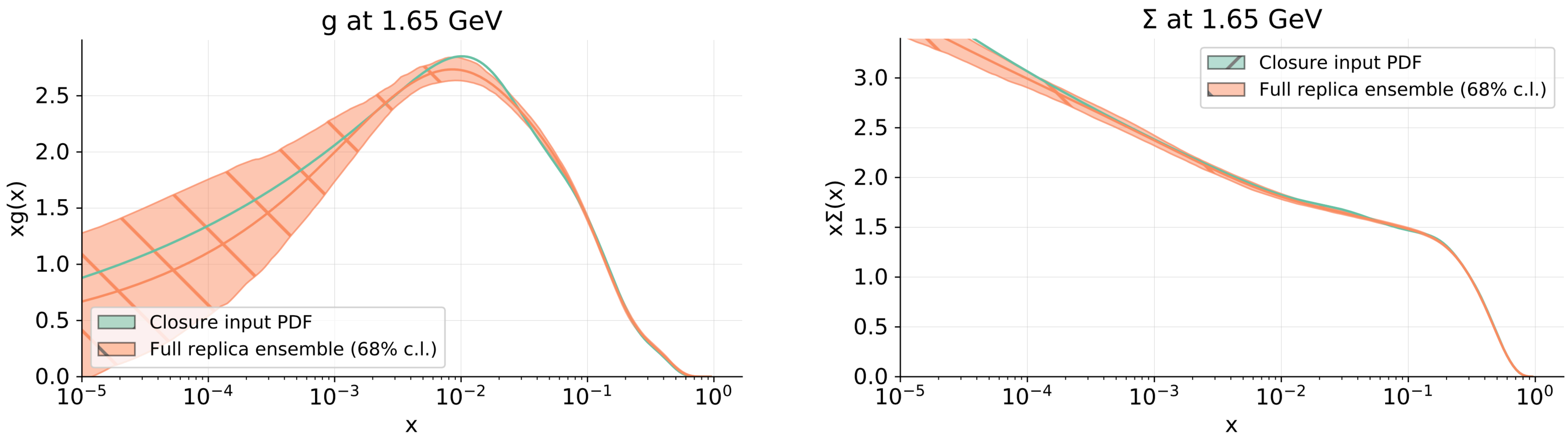
- As anticipated, PDF4LHC21 represents the **average** (with $\Delta\chi^2 \sim 0$)

Closure Tests

Generate **"toy data"** based on some known PDF and check a posteriori that the true **underlying law** $\mathscr{F}$ is reproduced within errors. Fit replicas to pseudodata in the standard way according to:

$$\mathscr{Y} = \mathscr{F} + \eta + \epsilon, \text{ where } \eta \sim \mathscr{N}(0,C) \text{ and } \epsilon \sim \mathscr{N}(0,C)$$

If the uncertainty associated to the PDF replicas is faithfully reproduced, then the **bias-to-variance ratio** should be unity, ie. $\mathscr{R}_{bv} \equiv \sqrt{\mathbf{E}_\eta[\text{ bias }]/\mathbf{E}_\eta[\text{ variance }]} = 1$.



Closure Tests

Generate **"toy data"** based on some known PDF and check a posteriori that the true **underlying law** $\mathscr{F}$ is reproduced within errors. Fit replicas to pseudodata in the standard way according to:

$$\mathscr{Y} = \mathscr{F} + \eta + \epsilon, \text{ where } \eta \sim \mathcal{N}(0,C) \text{ and } \epsilon \sim \mathcal{N}(0,C)$$

If the uncertainty associated to the PDF replicas is faithfully reproduced, then the **bias-to-variance ratio** should be unity, ie. $\mathscr{R}_{bv} \equiv \sqrt{\mathbf{E}_\eta[\text{ bias }]/\mathbf{E}_\eta[\text{ variance }]} = 1$.

Dataset	$\sqrt{\text{bias/variance}}$	$\xi_{1\sigma}^{(\text{data})}$
DY	0.99 ± 0.08	0.69 ± 0.02
Top-pair	0.75 ± 0.06	0.75 ± 0.03
Jets	1.14 ± 0.05	0.63 ± 0.03
Dijets	0.99 ± 0.07	0.70 ± 0.03
Direct photon	0.71 ± 0.06	0.81 ± 0.03
Single top	0.87 ± 0.07	0.69 ± 0.04
Total	1.03 ± 0.05	0.68 ± 0.02

Open Source

Test conda package passing | DOI 10.5281/zenodo.10730835

NNPDF: An open-source machine learning framework for global analyses of parton distributions

The NNPDF collaboration determines the structure of the proton using Machine Learning methods. This is the main repository of the fitting and analysis frameworks. In particular it contains all the necessary tools to reproduce the NNPDF4.0 PDF determinations.

Documentation

The documentation is available at https://docs.nnpdf.science/

Github: https://github.com/NNPDF/nnpdf

Documentation: https://docs.nnpdf.science/

Search docs

- Getting started
- Fitting code: `n3fit`
- Code for data: validphys
- Storage of data and theory predictions
- Theory
- Chi square figures of merit
- Contributing guidelines and tools
- Releases and compatibility policy
- Continuous integration and deployment
- Servers
- External codes
- Tutorials
  - Running fits
  - Analysing results
  - Closure tests
  - Special PDF sets
  - Miscellaneous

Tutorials

This section contains tutorials for common things you might want to do using the code. Adding to the Documentation and Reviewing pull requests).

Running fits

- How to run a PDF fit
- How to run an iterated fit
- How to run a QED fit
- How to run a Polarized fit
- Including a general theory covariance matrix in a fit
- How to include a theory covariance matrix in a fit

Analysing results

- How to compare two fits
- How to generate a report
- How to run an analysis in parallel
- Using dask without a Scheduler
- How to plot PDFs, distances and luminosities
- Plotting non-trivial combinations of PDFs
- How to do a data theory comparison
- Interpreting the \(\mathcal{R}_O\) overfit metric

Closure tests

- How to run a closure test
- How to analyse a closure test

Special PDF sets

- Bundle PDFs with \(\alpha_s\) replicas
- How to transform a Monte Carlo PDF set into a Hessian PDF set

Summary & Outlook

- Monte Carlo (MC) PDF extractions requires large ensembles of independent fits and robust uncertainty propagation
- HPO is essential to reduce methodological bias, but further increases the computational cost
- GPUs and HPC systems provide the parallelism and scalability needed to make these workflows practical
- Distributed HPO and simultaneous replica training significantly reduce time-to-solution
- Ensemble-based model selection offers a promising direction to **better capture methodological uncertainties**

“Wanderer above the Sea of Fog” by Caspar David Friedrich