Bayesian Neural Network for Robust UQ in PDFs

Dakshansh Chawda[1], Raphaël Bonnet-Guerrini[2], Stefano Carrazza[1], Stefano Forte[1], Eva Groenendijk[1], Vincenzo Piuri[2], and Ramon Winterhalder[1]

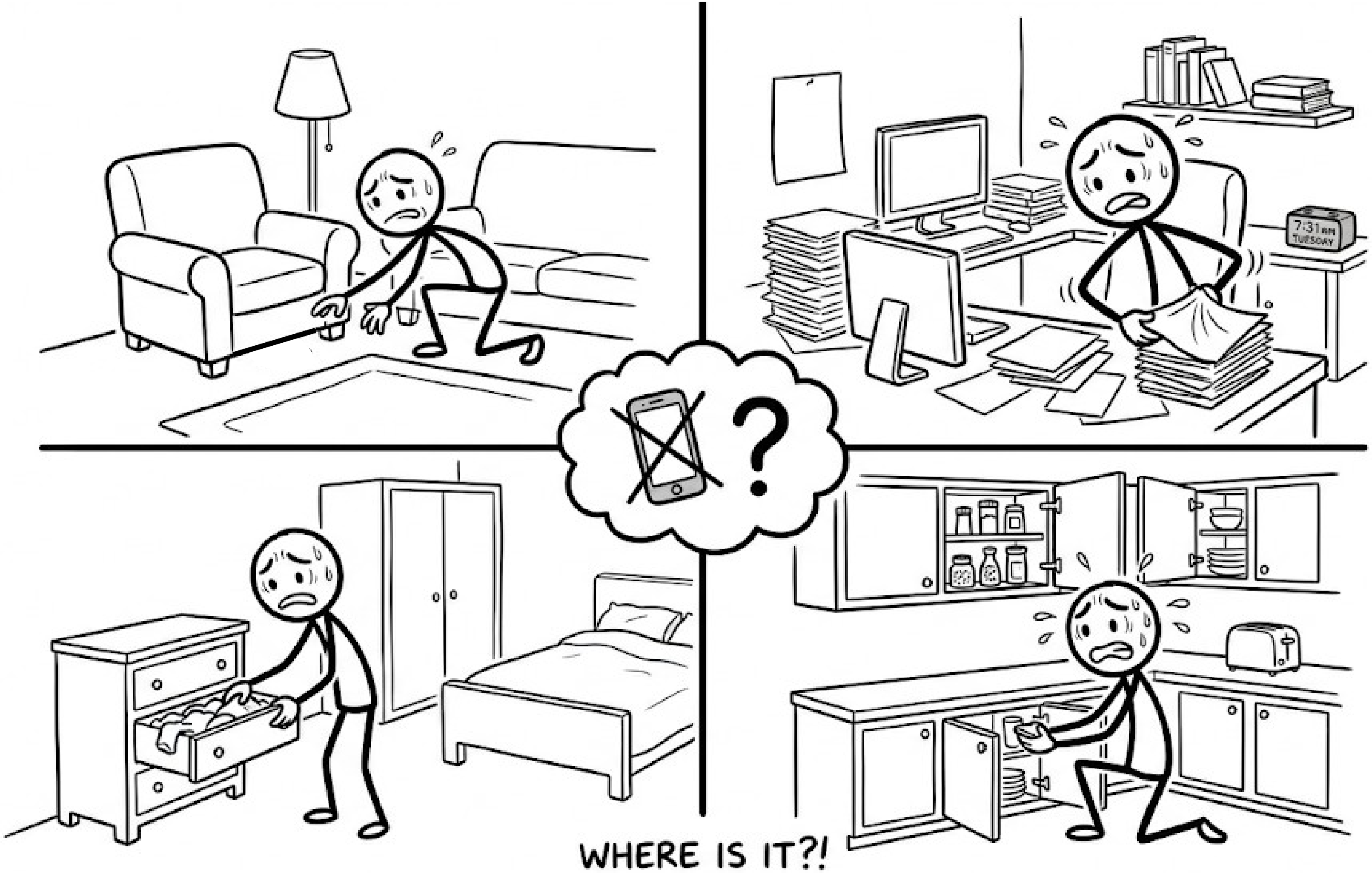
[1]Physics Department, University of Milan; [2]Computer Science Department, University of Milan

UNIVERSITÀ DEGLI STUDI DI MILANO

AIPHY

TUESDAY MORNING! Yay!


?
Gulp
no!
7:31 AM
TUESDAY


7:31 AM
TUESDAY
WHERE IS IT?!

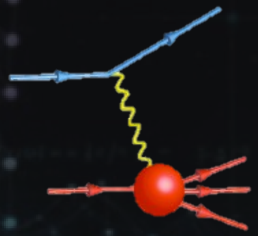
REVELATION AND RUSH!

Outline

- Introduction
  - PDFs
  - NNs
- The NNPDF Framework
- Bayesian Neural Network
- Results
- Outlook

Introduction: PDFs

$$\frac{d^2\sigma}{dx\,dy} = \frac{8\pi\alpha^2 ME}{Q^4}\left\{\left[\frac{1+(1-y)^2}{2}\right]2xF_1 + (1-y)F_L - \frac{M}{2E}xyF_2\right\}$$



- Structure functions: $F_i = \sum_a C_i^a \otimes f_a$

Process-dependent Coefficient functions

Universal Parton Distribution function $f_a(x, Q^2)$

(x: Momentum fraction)

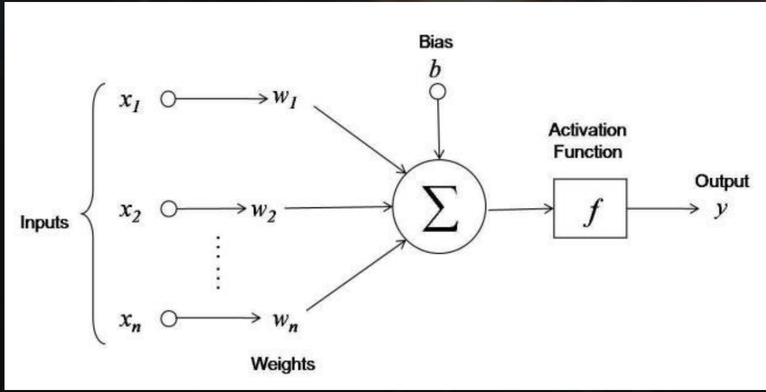
"such factorization holds for *most* observables involving a hard hadronic interaction"

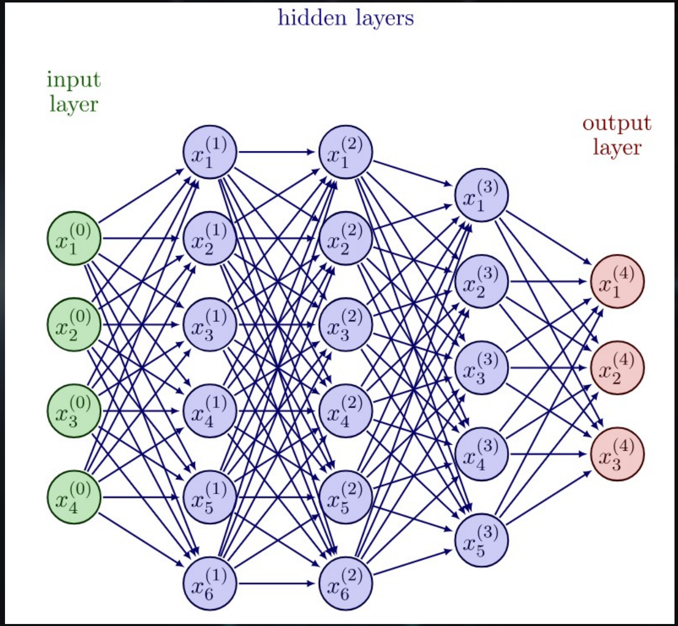
Introduction: NN

- Universal Approximation Theorem

$$\psi(x) := \sigma\left(\sum_{i=1}^{n} x_i w_i - b\right)$$

$$\sigma(x) := \frac{1}{1+\exp(-x)}$$





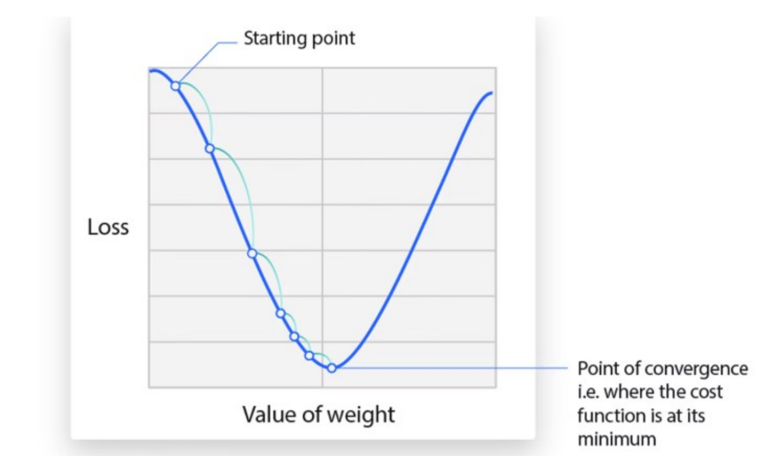
Introduction: Training a NN

- Data: (x, y) = Training + Test
- x → Model (w, b) → y^{train}
- Loss function

$$\mathcal{L}^{\mathrm{S}} := \mathcal{L}^{\mathrm{S}}(\mathbf{y}, \mathbf{y}^{\text{train}}) = \sum_i \left(y_i - y_i^{\text{train}}\right)^2$$

- Backpropagation

$$w_j^i \leftarrow w_i^j - \epsilon \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}w_j^i}, \quad \forall\, i, j$$



NN + PDF = NNPDF

What?

NNPDF : Neural Network Parton Distribution Function

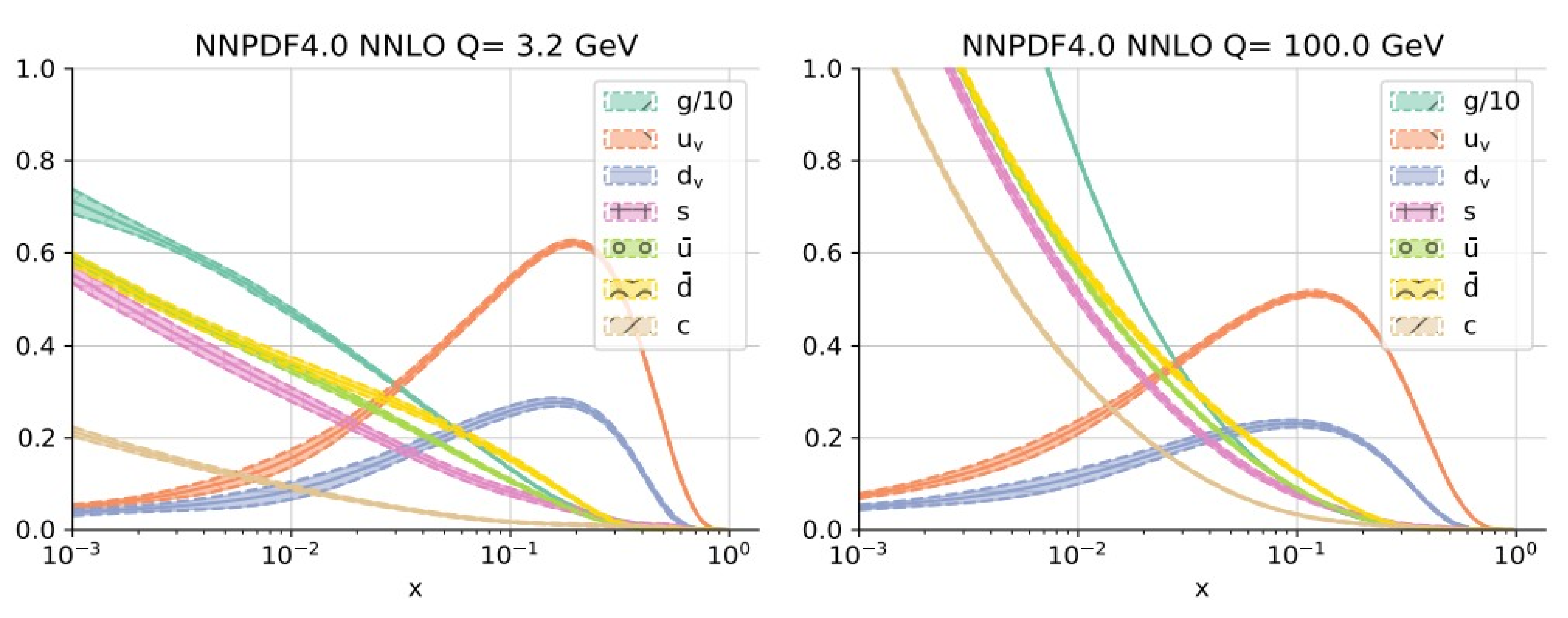
Why?

$$\sigma_{\text{had}} = \sum_{i,j} \int\int dx_1 dx_2 f_i(x_1, \mu_F) f_j(x_2, \mu_F) \hat{\sigma}_{ij}(x_1, x_2, \mu_F)$$

How?

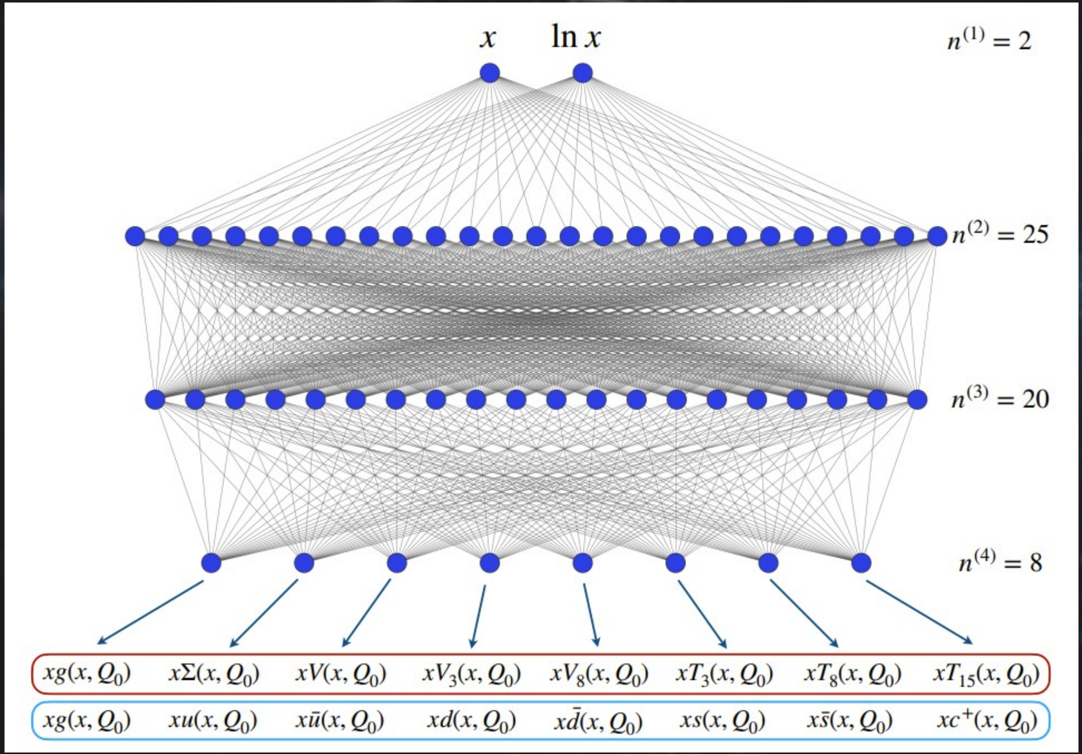
Theory (NNLO_QCD, NLO_EW, mixed QCD-EW processes) + Data

$$x f_k(x, Q_0; \boldsymbol{\theta}) = A_k x^{1-\alpha_k} (1-x)^{\beta_k} NN_k(x; \boldsymbol{\theta})$$



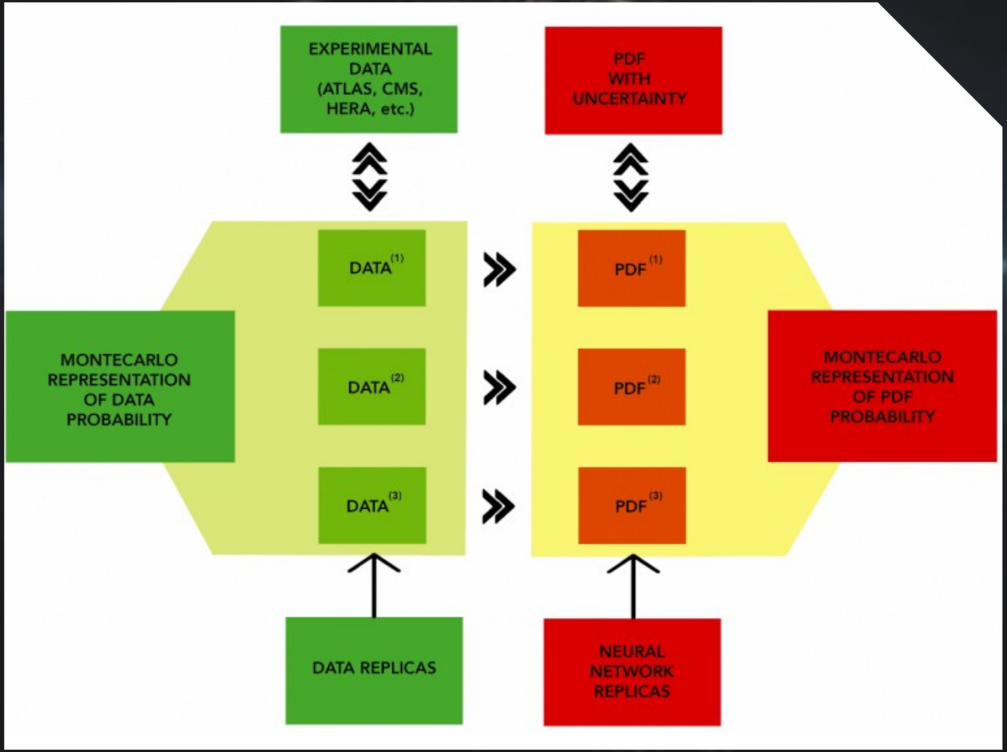
arxiv:2109.02653

The NNPDF Framework: Model Architecture



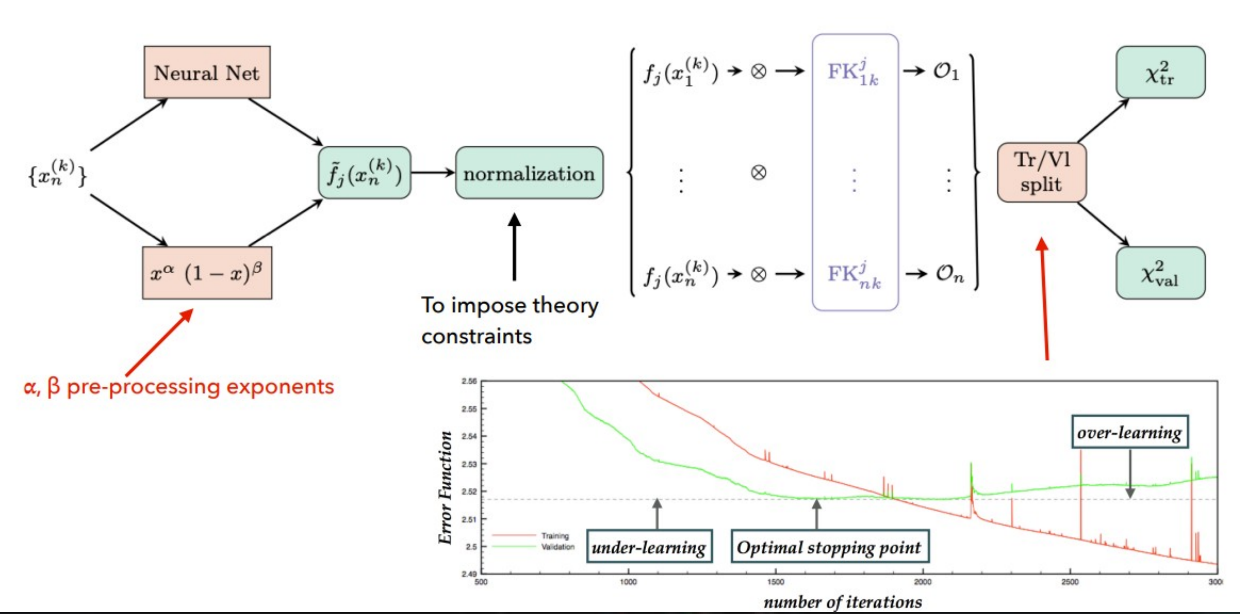
- Single DNN
- Free of parametrization bias
- NN > fixed parametrization; allows to include more data in fit

The NNPDF Framework: Methodology



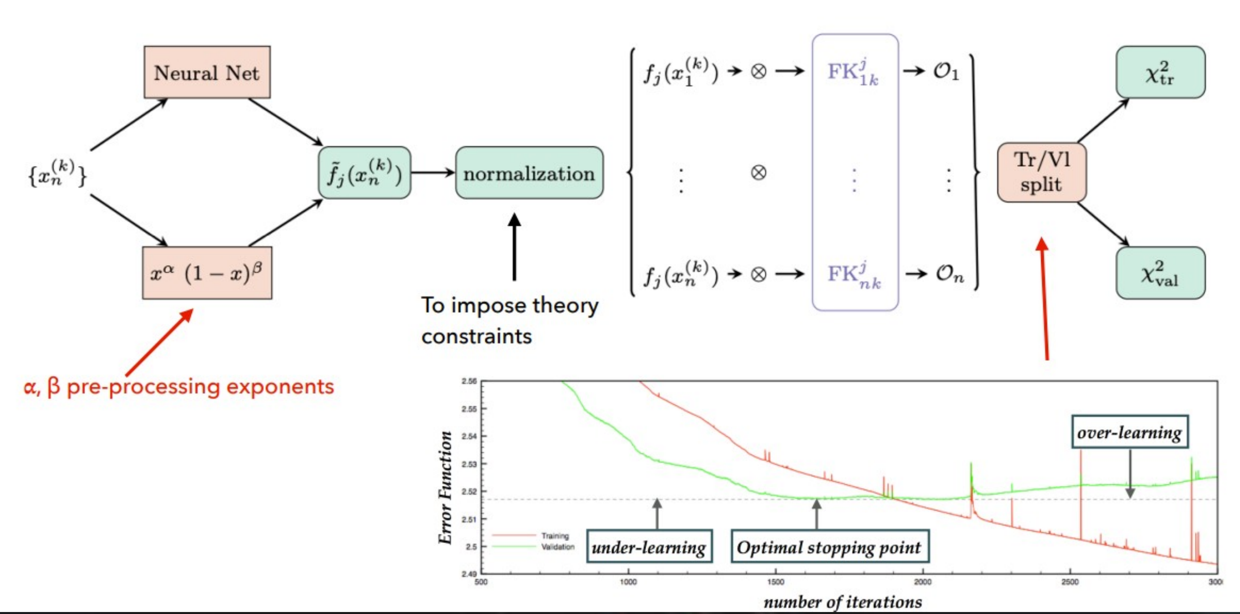
- Data space sampling projected to PDF space
- Propagate uncertainty via MC sampling

The *n3fit* recipe



arxiv:1907.05075

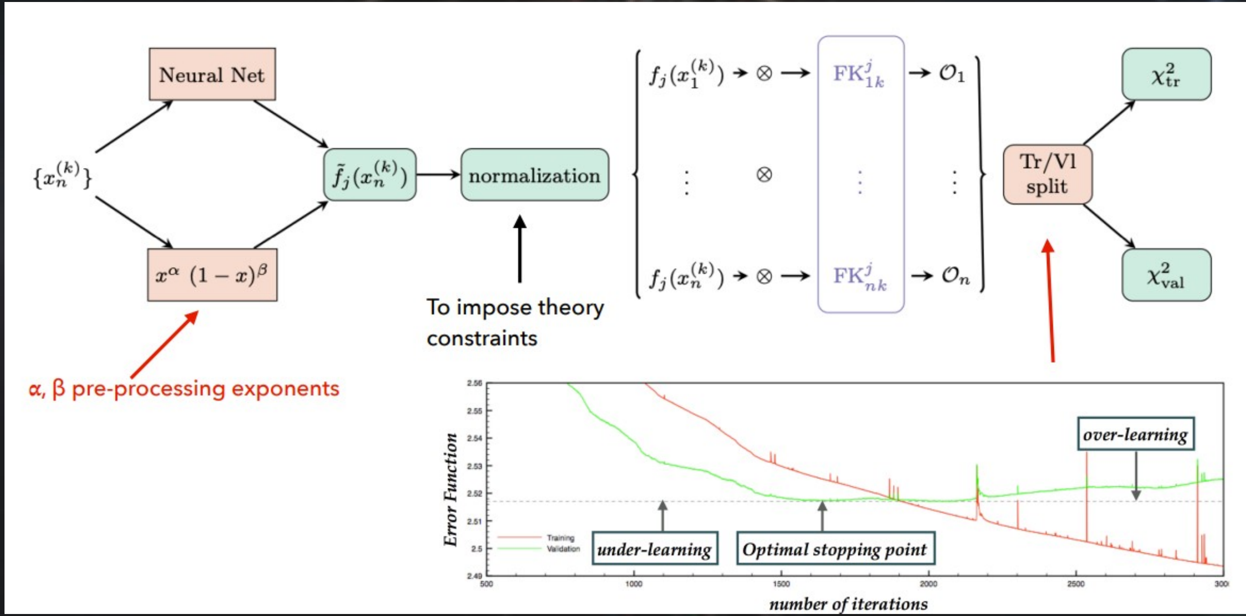
The *n3fit* recipe



[arxiv:1907.05075](https://arxiv.org/abs/1907.05075)

1. Ingredients
   1. Data
   2. Theory
   3. Q0
2. Parametrize PDFs at Q0
3. Evolve PDFs Q0 to Q
4. Compute observables
5. Fit PDFs to Data
6. Compute uncertainties

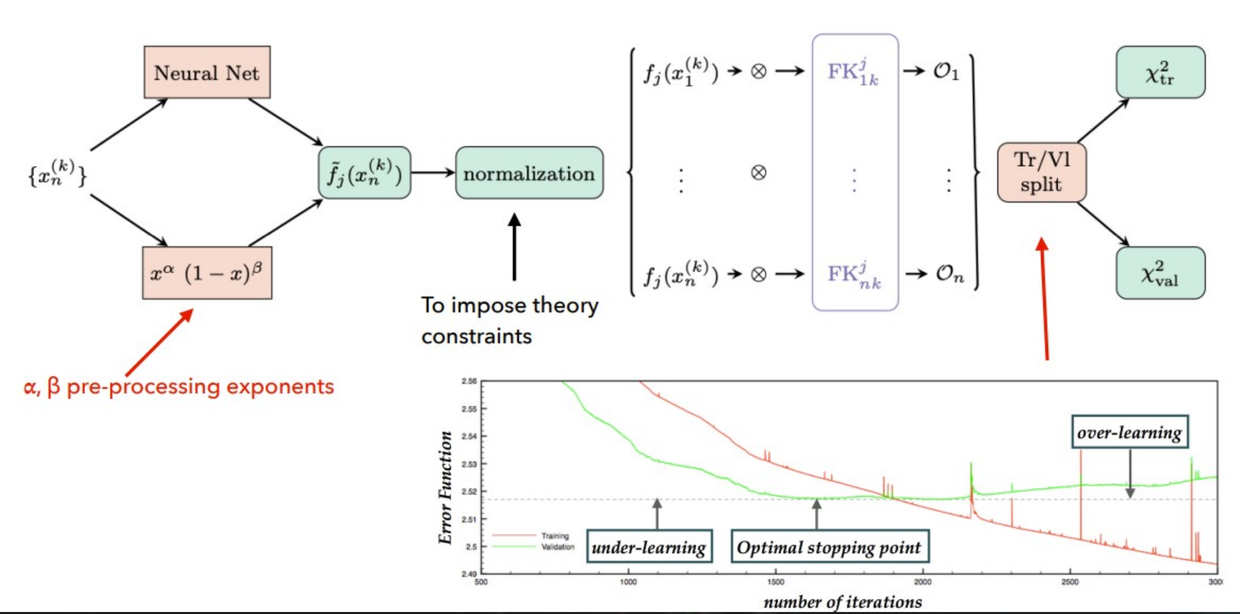
The *n3fit* recipe



arxiv:1907.05075

1. Ingredients
   1. Data
   2. Theory
   3. Q0
2. Parametrize PDFs at Q0
3. Evolve PDFs Q0 to Q
4. Compute observables
5. Fit PDFs to Data
6. Compute uncertainties

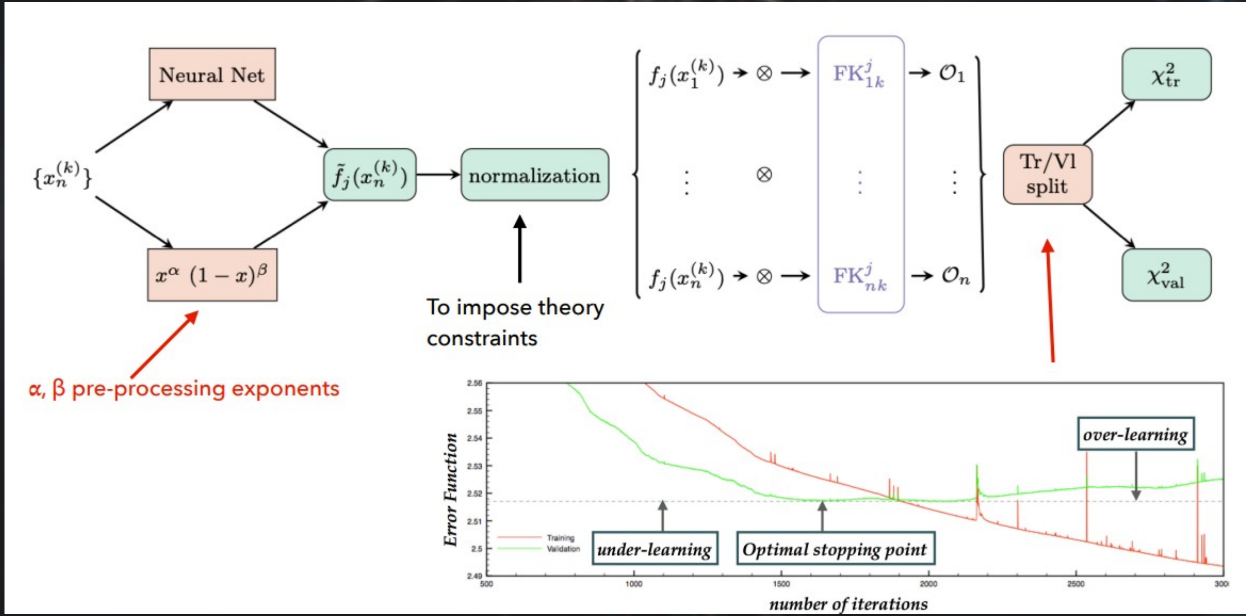
The *n3fit* recipe



[arxiv:1907.05075](https://arxiv.org/abs/1907.05075)

1. Ingredients
   1. Data
   2. Theory
   3. Q0
2. Parametrize PDFs at Q0
3. Evolve PDFs Q0 to Q
4. Compute observables
5. Fit PDFs to Data
6. Compute uncertainties

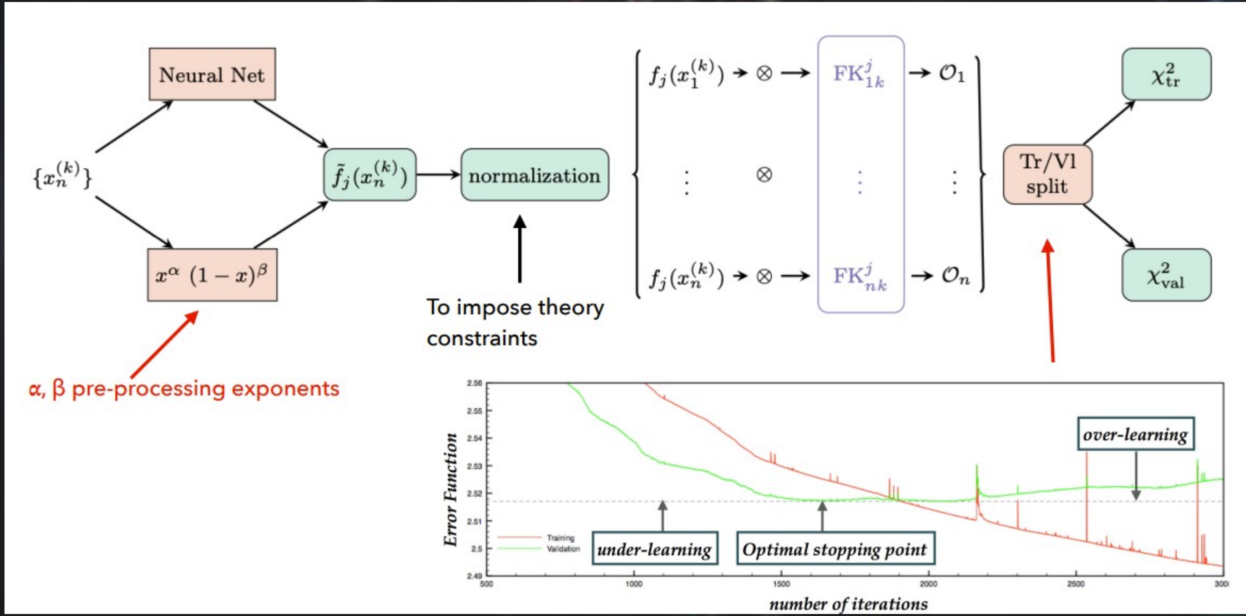
The *n3fit* recipe



arxiv:1907.05075

1. Ingredients
    1. Data
    2. Theory
    3. Q0
2. Parametrize PDFs at Q0
3. Evolve PDFs Q0 to Q
4. Compute observables
5. Fit PDFs to Data
6. Compute uncertainties

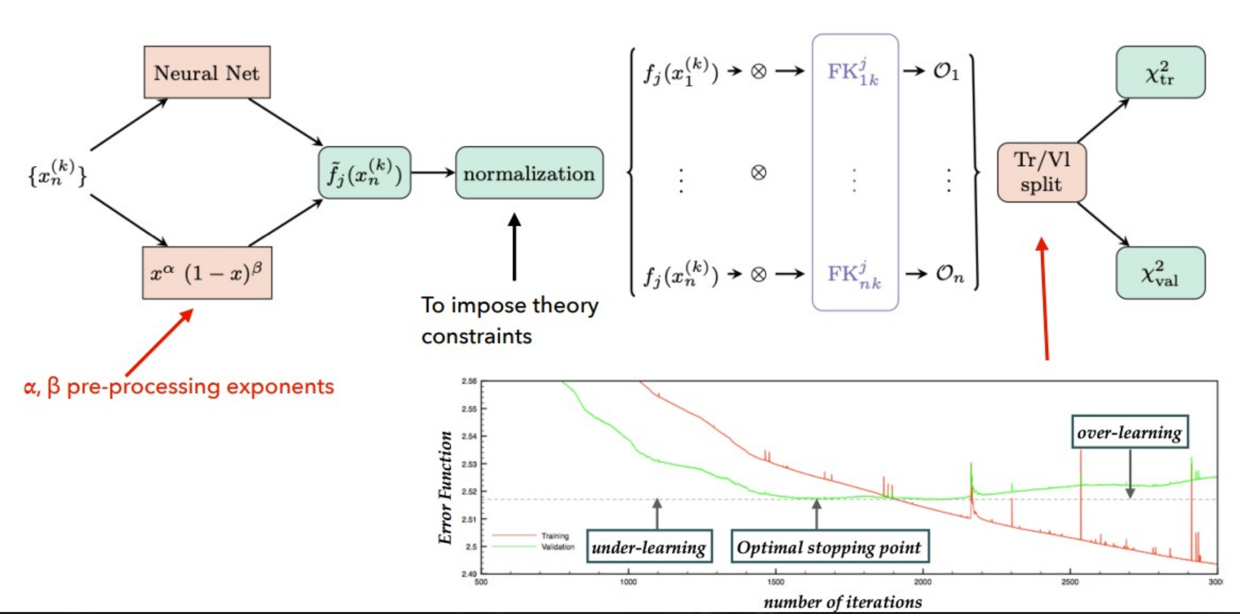
The *n3fit* recipe



arxiv:1907.05075

1. Ingredients
   1. Data
   2. Theory
   3. Q0
2. Parametrize PDFs at Q0
3. Evolve PDFs Q0 to Q
4. Compute observables
5. Fit PDFs to Data
6. Compute uncertainties

The *n3fit* recipe



arxiv:1907.05075

1. Ingredients
   1. Data
   2. Theory
   3. Q0
2. Parametrize PDFs at Q0
3. Evolve PDFs Q0 to Q
4. Compute observables
5. Fit PDFs to Data
6. Compute uncertainties

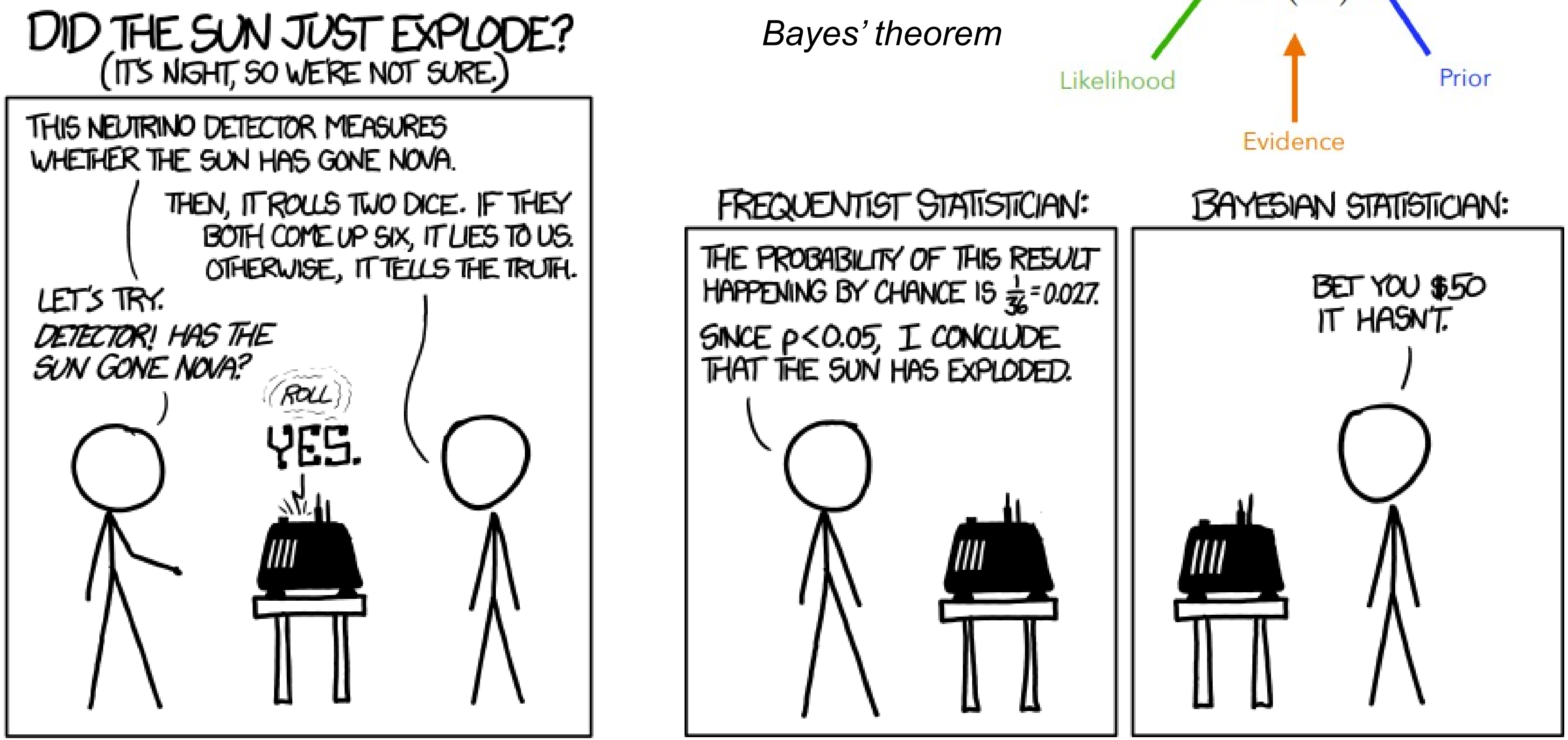
$$P(T|D) \approx P(\vec{\alpha}|D) = \frac{P(D|\vec{\alpha})\,P(\vec{\alpha})}{P(D)}$$

Bayes' theorem

Likelihood

Evidence

Prior



HTTPS://XKCD.COM/1132/

Bayesian Neural Network

- addresses the fundamental limitation of the "frequentist-like" Monte Carlo replica method:
  - Computational cost ⟶ ‘1 for 1000’

$$w \rightarrow \mu, \sigma$$

  - Better extrapolation in the extreme-x regions (with sparse data)
  - KL divergence

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}$$

$$BNN\ loss = \mathcal{L}(PDF) + KL$$

Bayesian Neural Network

- Robust Uncertainty Quantification

S-NN : UQ ⟵ Data fluctuation ⟵ Expt. Cov. Matrix

Each replica is a point estimate; different loss minima, no guarantee that minima covers true Uncertainty

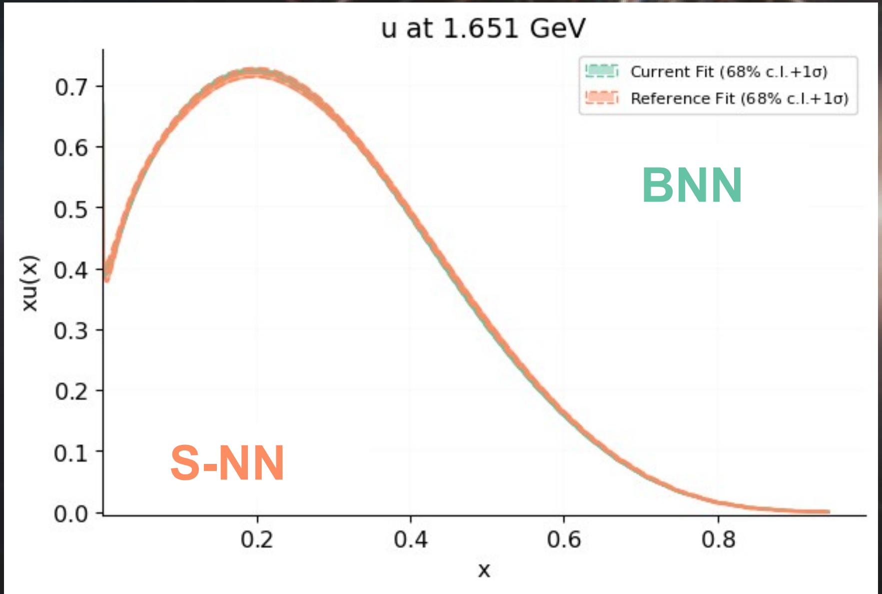
Network can confidently overfit

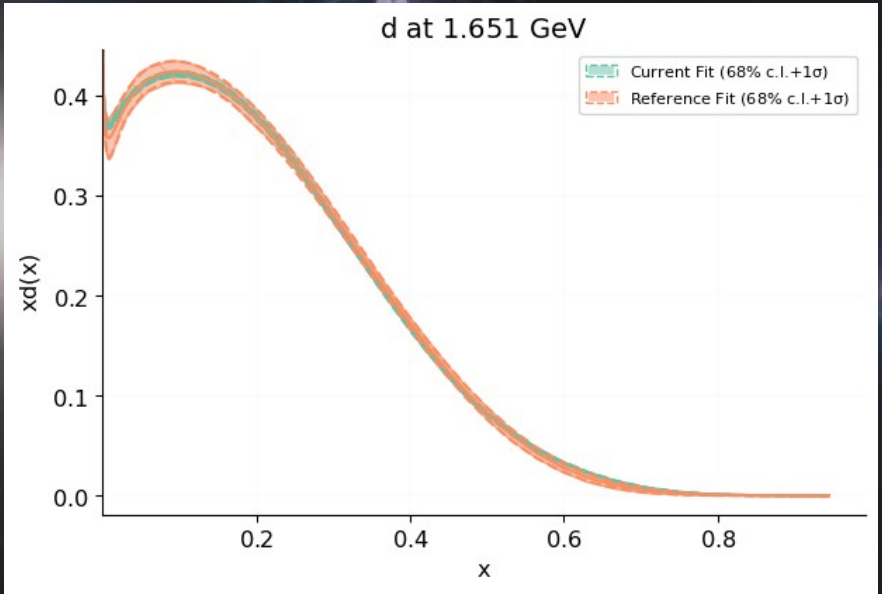
BNN : KL penalizes posterior for being complex than prior w/o being justified by the data

Network forced to be uncertain in sparse-data region

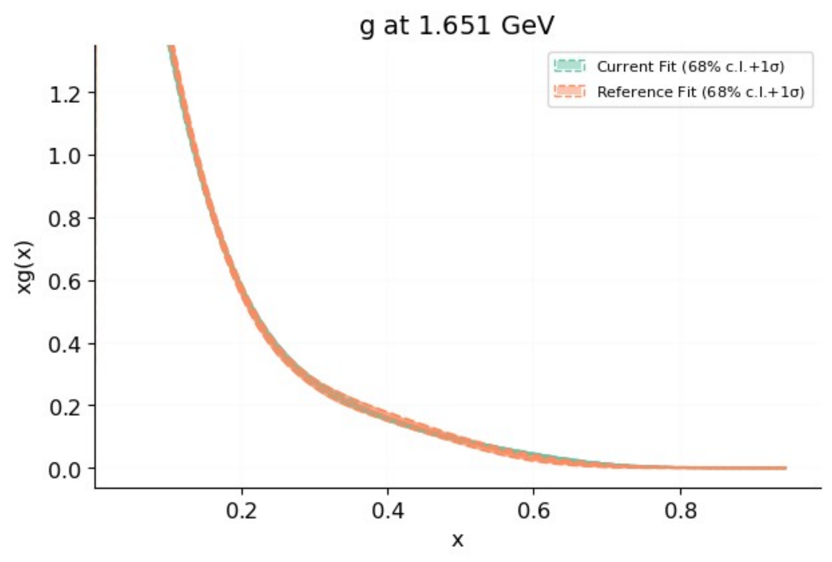
“BNN itself learns which directions in weight space are uncertain, rather than having uncertainty imposed externally through data fluctuations”

Results

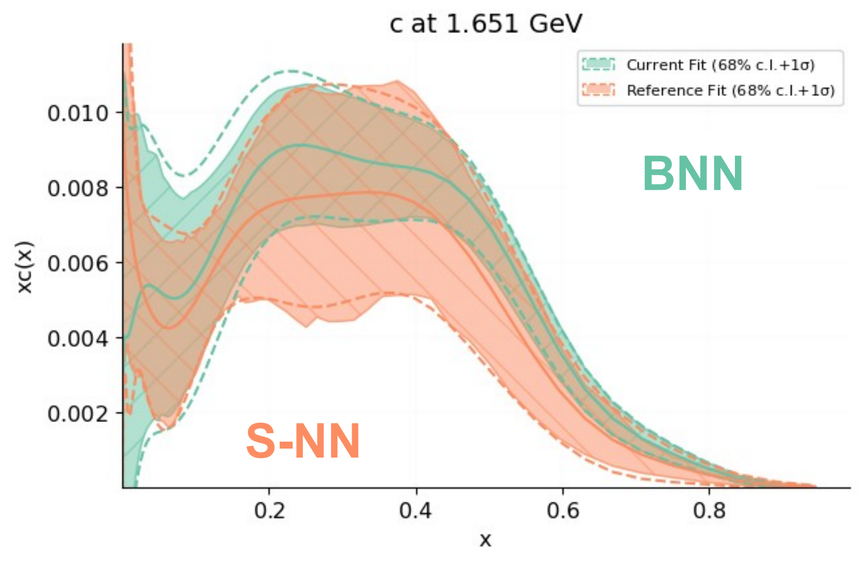

u at 1.651 GeV
Current Fit (68% c.l.+1σ)
Reference Fit (68% c.l.+1σ)
BNN
S-NN
xu(x)
x
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.2
0.4
0.6
0.8


d at 1.651 GeV
Current Fit (68% c.l.+1σ)
Reference Fit (68% c.l.+1σ)
xd(x)
x
0.0
0.1
0.2
0.3
0.4
0.2
0.4
0.6
0.8

Results

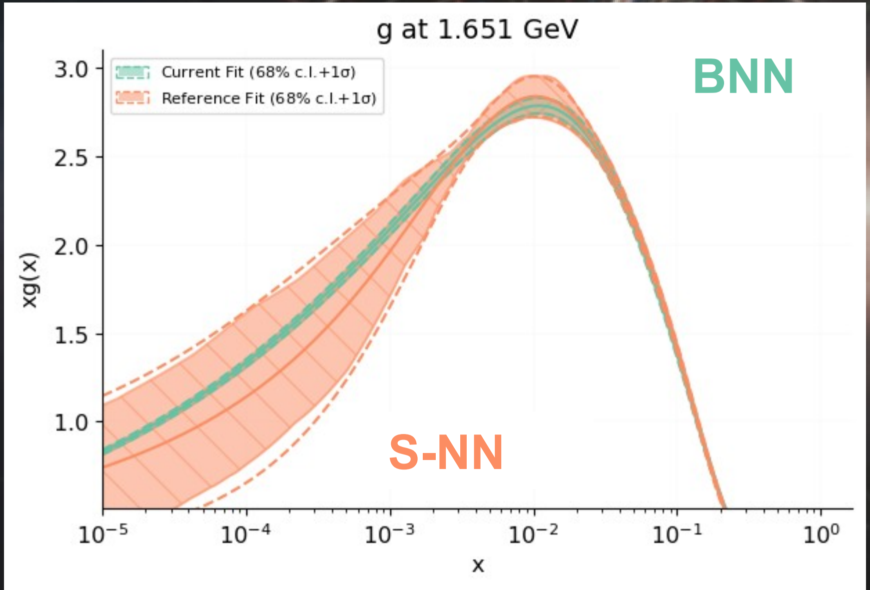


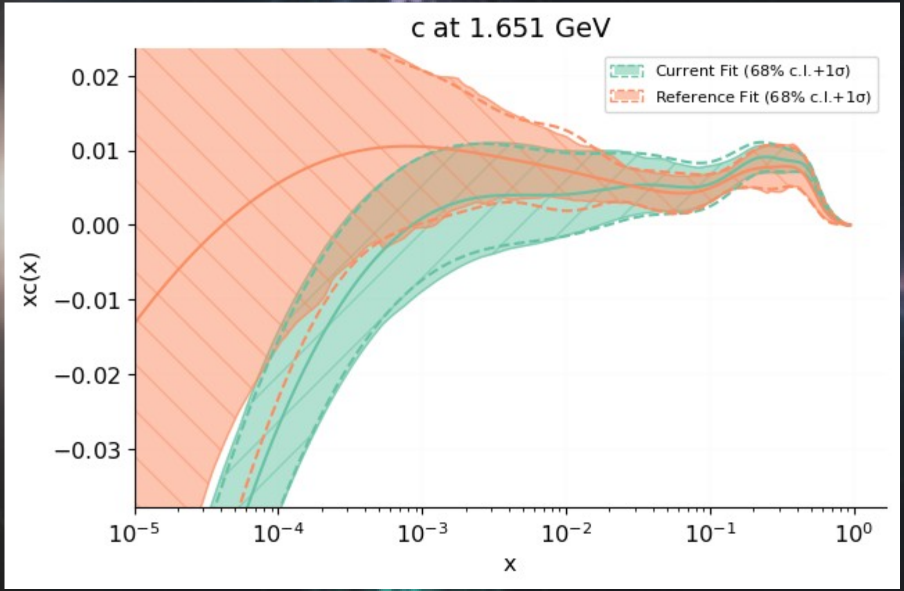
g at 1.651 GeV
Current Fit (68% c.l.+1σ)
Reference Fit (68% c.l.+1σ)
xg(x)
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8
x



c at 1.651 GeV
Current Fit (68% c.l.+1σ)
Reference Fit (68% c.l.+1σ)
BNN
S-NN
xc(x)
0.010
0.008
0.006
0.004
0.002
0.2
0.4
0.6
0.8
x

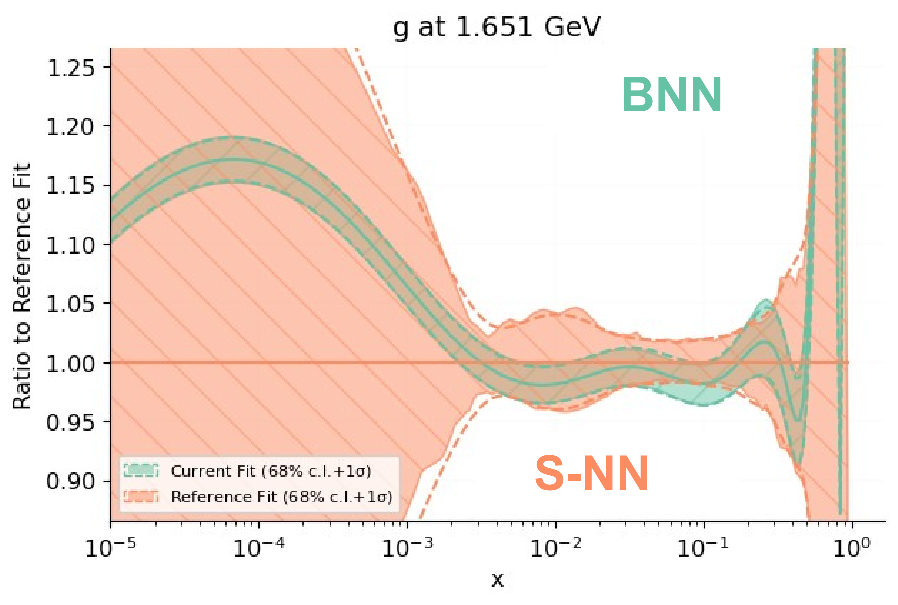
Results

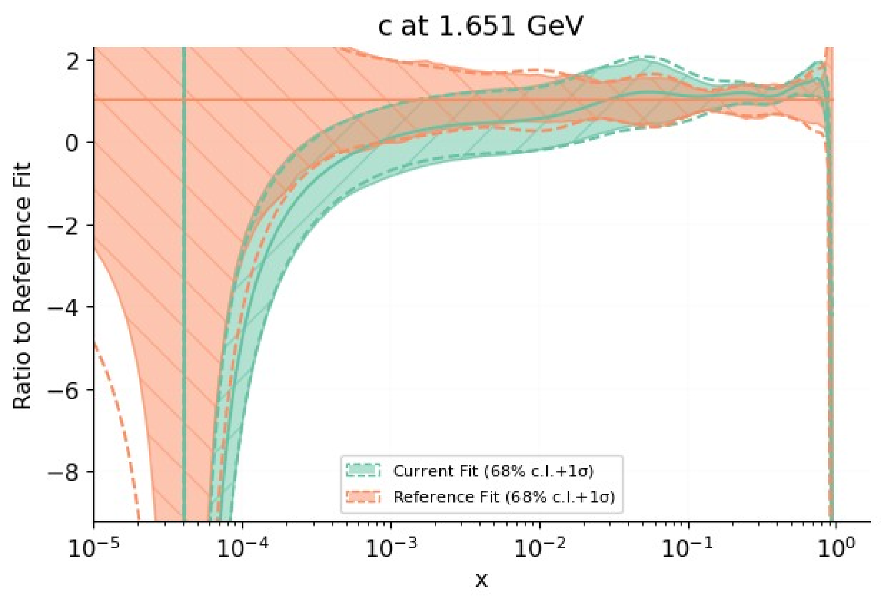




Log-log plots!

Room for improvement

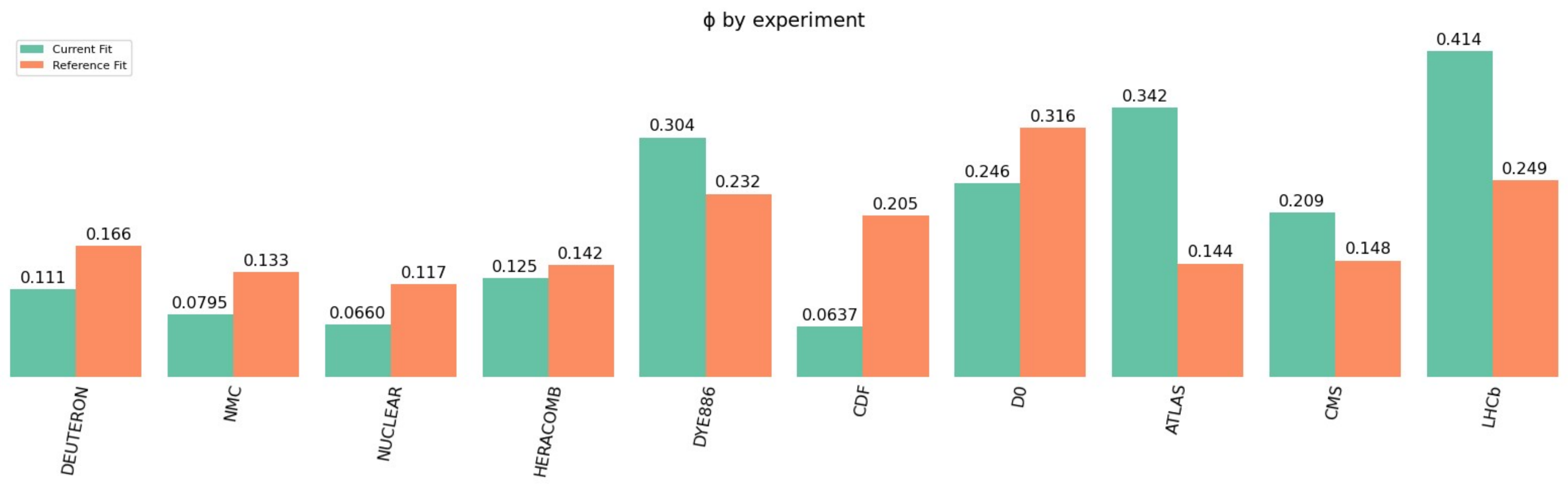




Room for improvement

Loss variance

	BNN	S-NN
Loss	1.21271	1.16983
< TL>	11683±0	11700±2000
<Loss>	1.248±0.036	1.193±0.015
ϕ	0.189±0.014	0.1506±0.0072



Outlook

- Theoretical Uncertainty : treated as "Frequentist", but truly "Bayesian"
  - Level 0: Should (at least) run ✓
  - Level 1: Test NNPDF methodology (F and B produce same result) ✓ (Almost!)
  - Level 2: Improve upon NNPDF methodology
- Hyper optimization of BNN: starting μ , σ for each weight, prior width, …
- Better Annealing Schedule ? $BNN\ loss = \mathcal{L}(PDF) + \beta \cdot KL$

Thank You!

This work was supported by the European Union's Horizon Europe research and innovation programme under the Marie Sklodowska-Curie grant agreement No 101168829, Challenging AI with Challenges from Physics:

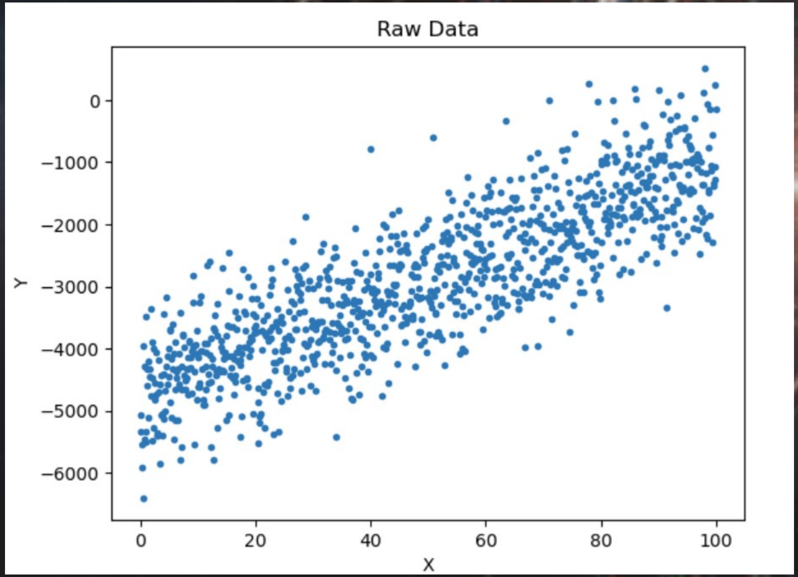
"How to solve fundamental problems in Physics by AI and vice versa (AIPHY)"

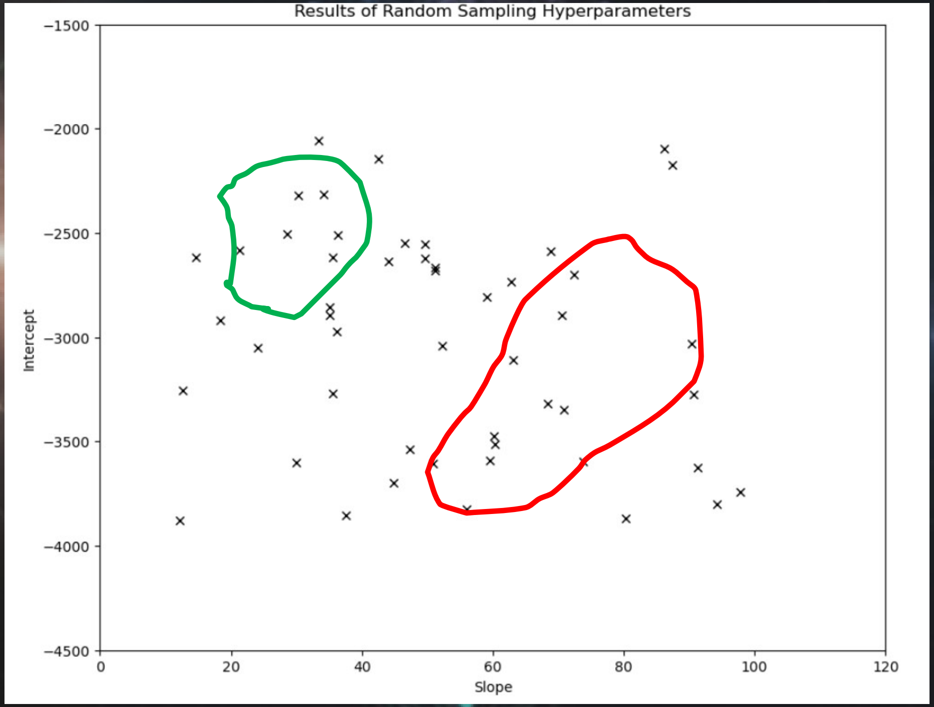
AIPHY

European Commission

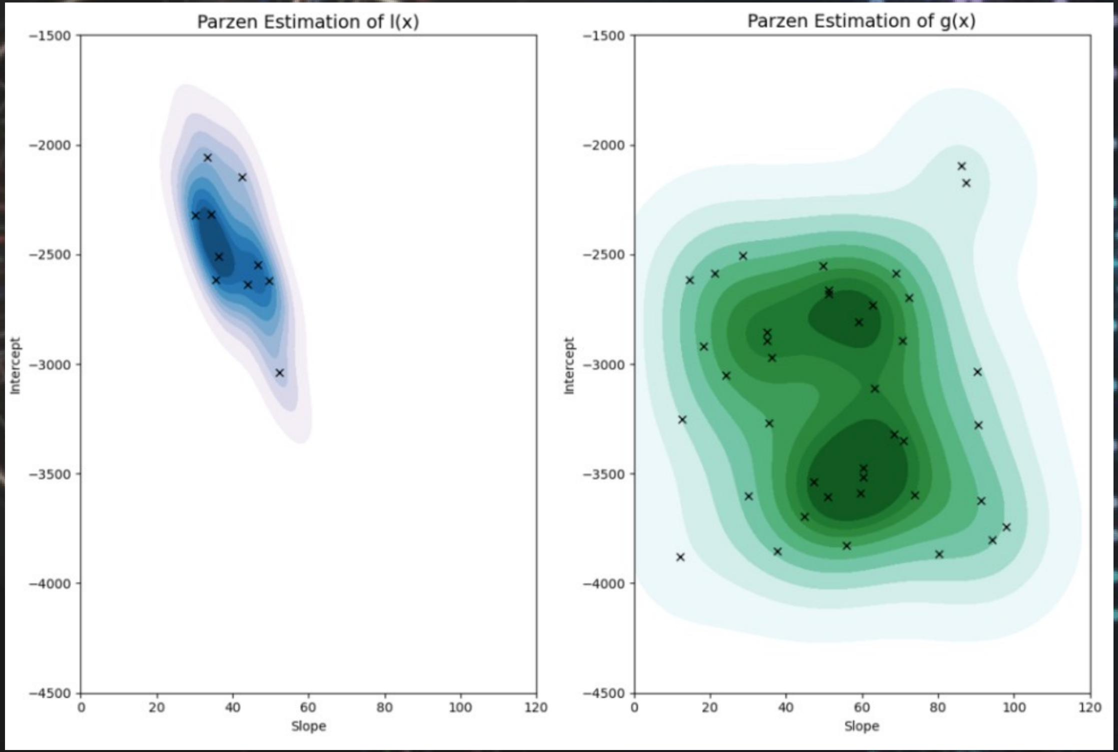
Extra Slides

Hyperoptimization: TPE





Hyperoptimization: TPE

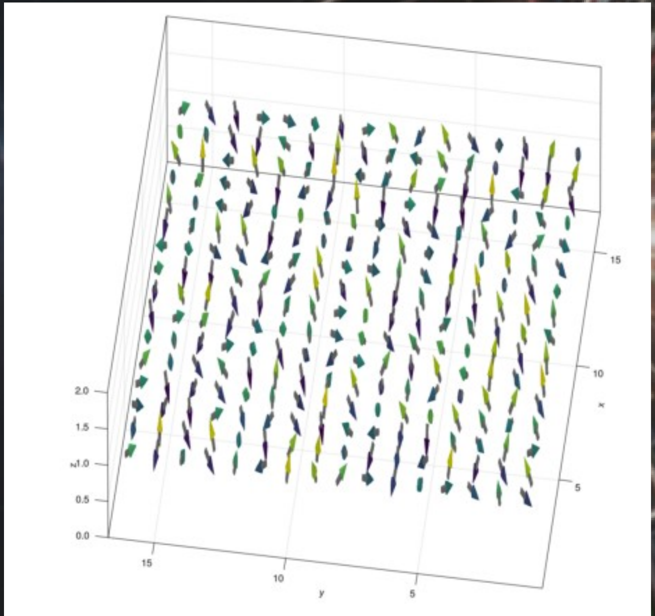


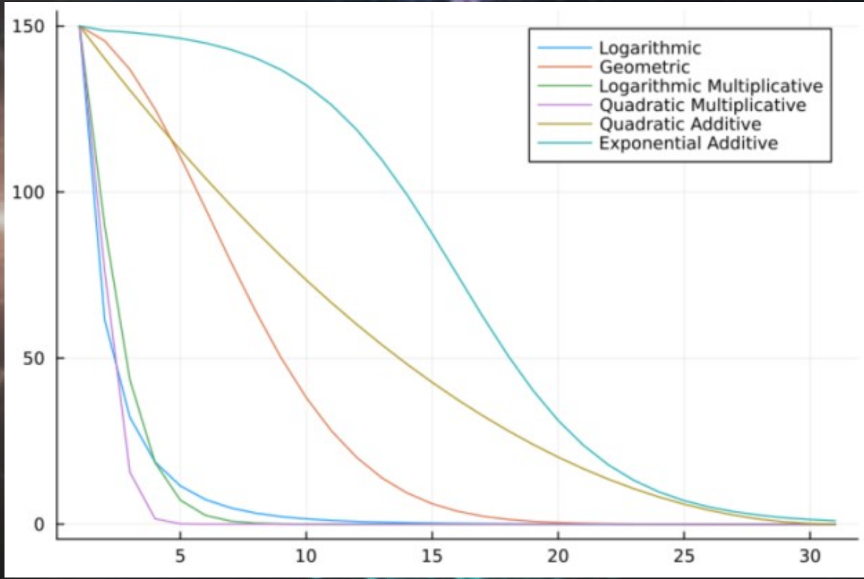
bad g(x)

good l(x)

max.
g(x)/l(x)

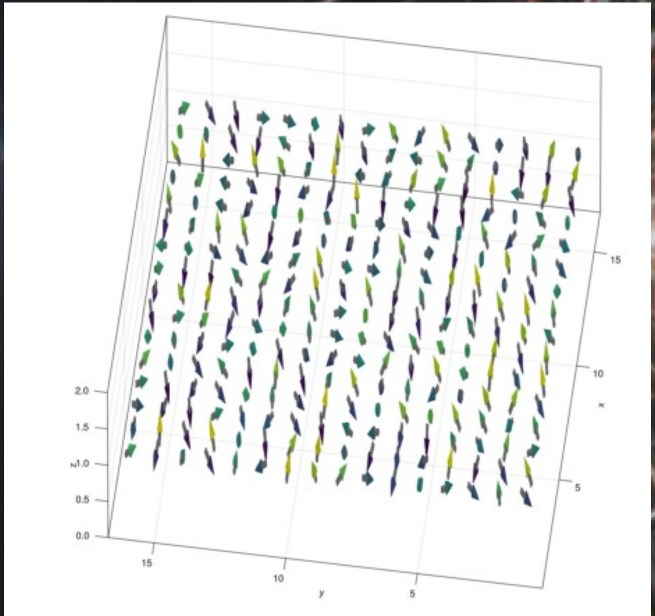
Simulated Annealing

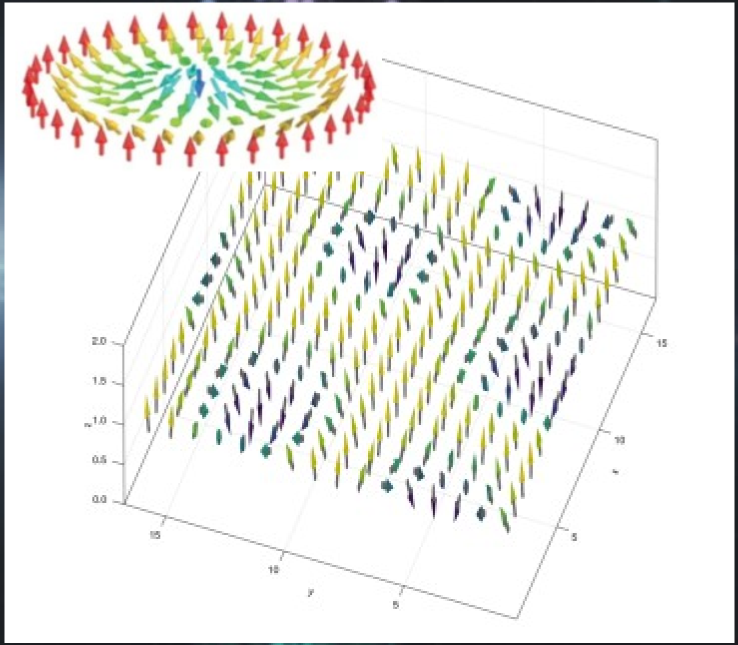




Eg.: a physical system

Simulated Annealing





Eg.: a physical system

“Magnetic Skyrmions”

Simulated Annealing

- BNN Loss = PDF Loss + KL Loss
- 'Spin' annealing ⟶ explore energy landscape
- KL annealing ⟶ explore loss landscape
  - Dominating PDF loss: frequentist-like,
  - Dominating KL loss: unphysical PDFs
  - Annealing brings a "balance"

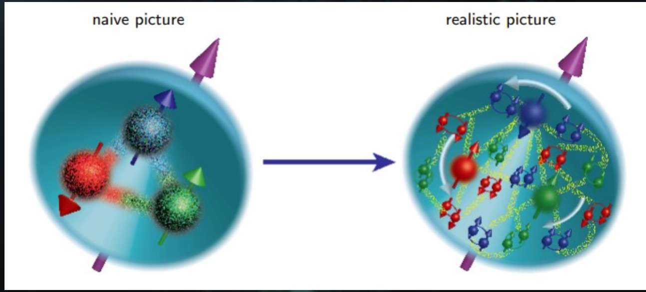
$$BNN\ loss = \mathcal{L}(PDF) + \beta \cdot KL$$

DGLAP equations

- PDFs evolve with the scale; evolution given by DGLAP equations:

$$\frac{\partial f_i(x,Q^2)}{\partial \ln Q^2} = \sum_j \frac{\alpha_s(Q^2)}{2\pi} \int_x^1 \frac{dz}{z} P_{ij}\left(\frac{x}{z}\right) f_j(z,Q^2)$$

- As scale increases, the available phase space for parton emission grows
  - more sea quarks and gluons
  - distributions at large x decrease, while those at small x increase



Theoretical Framework: From QCD to FK Tables

- Bottleneck in PDF fitting: convolution of PDF with cross section at EVERY training step

$$\sigma_{\text{had}} = \sum_{i,j} \int\int dx_1 dx_2 f_i(x_1, \mu_F) f_j(x_2, \mu_F) \hat{\sigma}_{ij}(x_1, x_2, \mu_F)$$

- FastKernel tables : expand PDF onto a grid of interpolation polynomials

$$f_i(x, Q^2) = \sum_{\alpha} f_i(x_\alpha, Q^2) I^{(\alpha)}(x)$$

$$\sigma_{\text{had}} = \sum_{i,j} \sum_{\alpha,\beta} f_i(x_\alpha, Q^2) f_j(x_\beta, Q^2) \Sigma_{ij}^{\alpha\beta}$$

Theoretical Framework: Theory Covariance Matrix

- NNPDF4.0 works at NNLO ~ splitting functions truncated at NNLO ~ MHOU
- Includes this as theory covariance matrix

$$C_{ij}^{\text{tot}} = C_{ij}^{\text{exp}} + S_{ij}^{\text{theory}}$$

- Estimated by Scale Variations
- Allows the fitted PDF to fluctuate more in a region with large S, without being penalized by loss
- Prevents overfitting theoretical inaccuracies

The NNPDF Framework: Closure Test

- Level 0: Methodology Consistency
  - single, noiseless pseudo-data for all replica fits;
  - expected: perfect match for each replica, loss vanishes
- Level 1: Data Fluctuations
  - pseudo-data + random noise
  - expected: methodology recovers the central value of the law despite noise
- Level 2: Full MC
  - proxy for a real-world fit
  - expected: PDF uncertainties (replica spread) matches the experimental uncertainties

The NNPDF Framework: Closure Test

- Level 0: Methodology Consistency
  - single, noiseless pseudo-data for all replica fits;
  - expected: perfect match for each replica, loss vanishes
- Level 1: Data Fluctuations
  - pseudo-data + random noise;
  - expected: methodology recovers the central value of the law despite noise
- Level 2: Full MC
  - proxy for a real-world fit
  - expected: PDF uncertainties (replica spread) matches the experimental uncertainties

The NNPDF Framework: Closure Test

- Level 0: Methodology Consistency
  - single, noiseless pseudo-data for all replica fits;
  - expected: perfect match for each replica, loss vanishes
- Level 1: Data Fluctuations
  - pseudo-data + random noise;
  - expected: methodology recovers the central value of the law despite noise
- Level 2: Full MC
  - proxy for a real-world fit;
  - expected: PDF uncertainties (replica spread) matches the experimental uncertainties