Small-x Dynamics at the Electron-Ion Collider via Machine Learning

Junaid Saif Khan | June 16, 2026

Gluon Saturation: Nonlinear Behavior

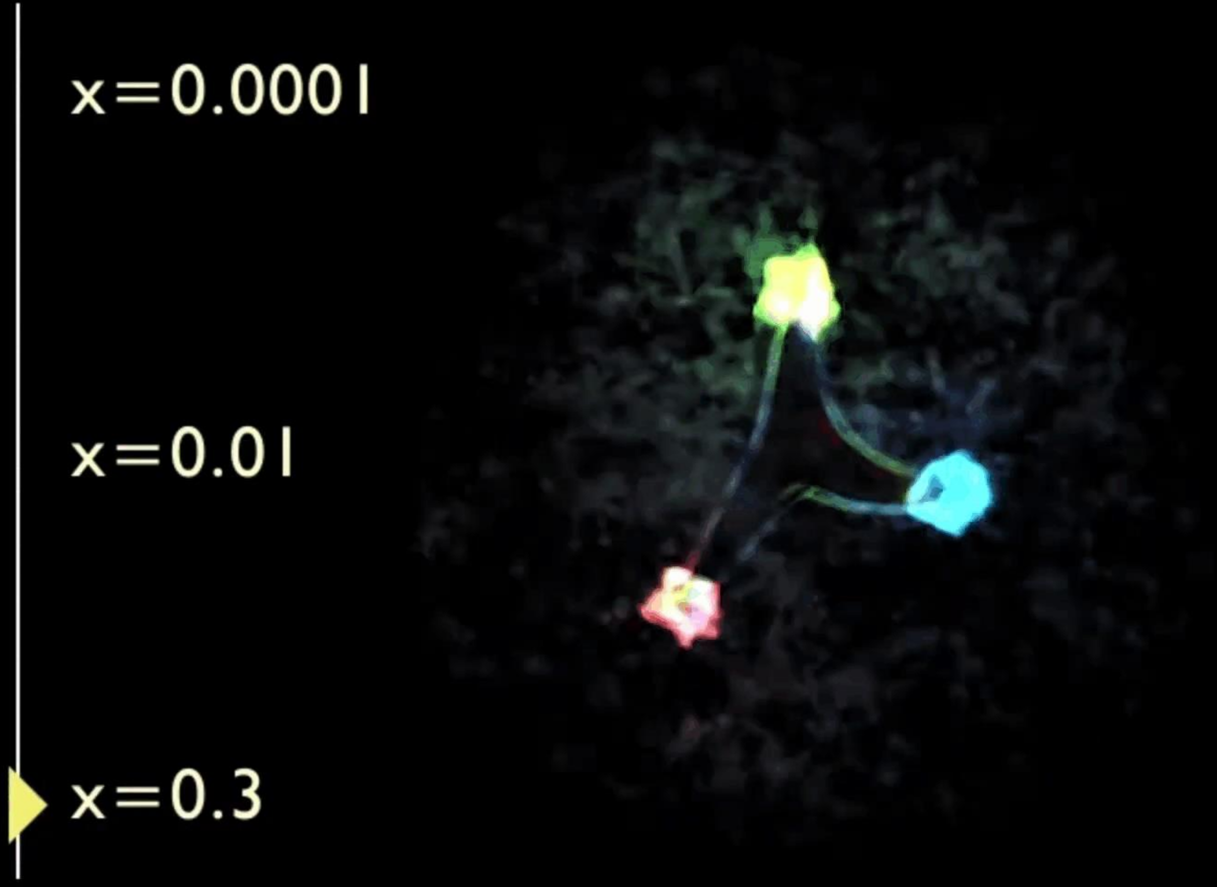
- **Nonlinear QCD effects are still needed to be studied**

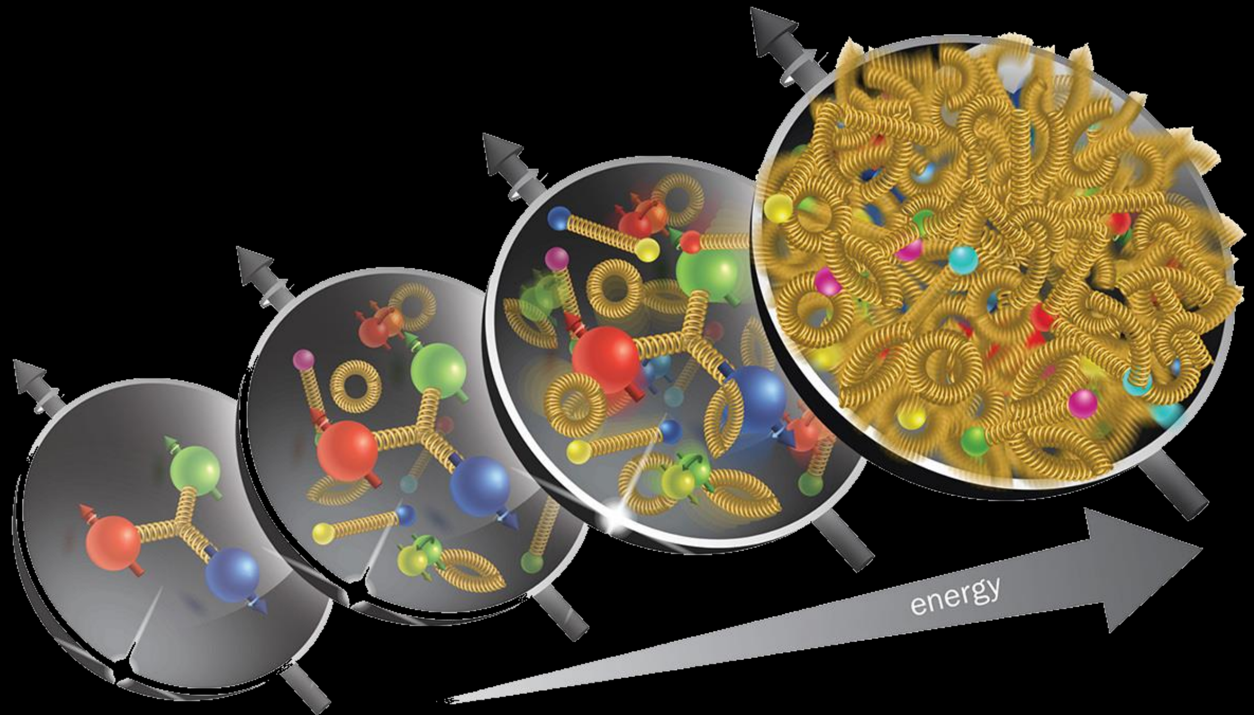
Signatures of gluon saturation from structure-function measurements

Nestor Armesto, Tuomas Lappi, Heikki Mäntysaari, Hannu Paukkunen and Mirja Tevio

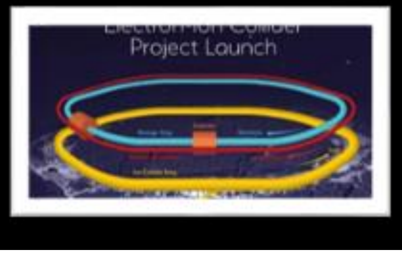
arXiv:2203.05846v2 [hep-ph] 16 Jun 2022

N. Armesto et al. [Signatures of gluon saturation],
Phys. Rev. D 105, 1114017 (2022)





Gluon density increases at low x
(Nonlinear effects)

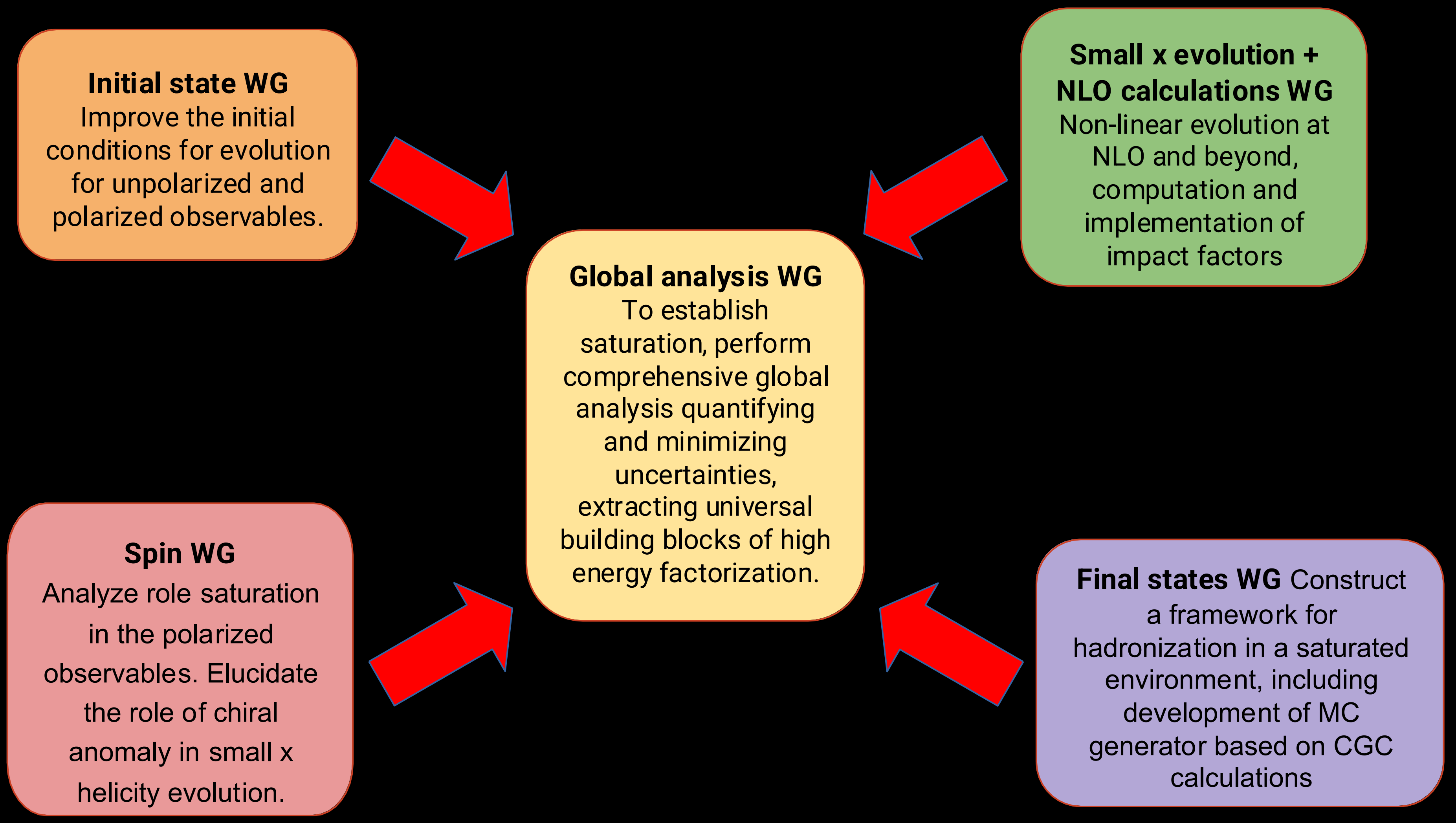


To understand this
we need new data and ideas

The Saturated Glue (SURGE) Collaboration

- **Discover and explore the gluon saturation regime of QCD**

GOAL: Develop the global analysis framework
The numerical calculation must be FAST
as this will go inside a fitting loop



Initial state WG
Improve the initial conditions for evolution for unpolarized and polarized observables.

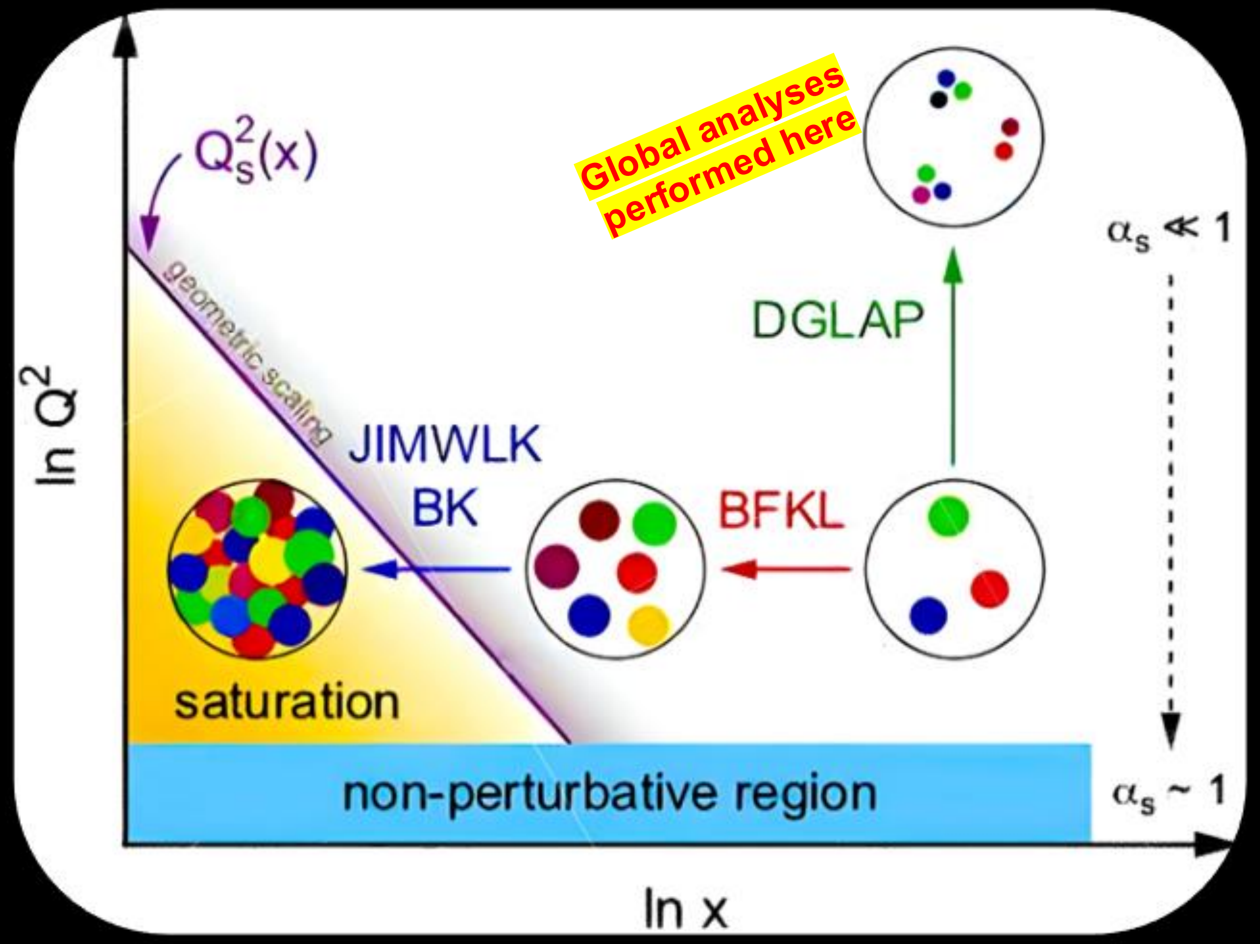
Small x evolution + NLO calculations WG
Non-linear evolution at NLO and beyond, computation and implementation of impact factors

Global analysis WG
To establish saturation, perform comprehensive global analysis quantifying and minimizing uncertainties, extracting universal building blocks of high energy factorization.

Spin WG
Analyze role saturation in the polarized observables. Elucidate the role of chiral anomaly in small x helicity evolution.

Final states WG Construct a framework for hadronization in a saturated environment, including development of MC generator based on CGC calculations

CHALLENGE: Extend global analyses into nonlinear regime



QCD evolution phase diagram

SURGE Global Analysis WG

Global Analysis WG: The SMU Team

Fred
Olness

Brandon
Stevenson
Grad Student

Rebecca
Preston
Grad Student

Junaid
Khan
Grad Student

Peter
Risse
Postdoc

with contributions from many others including

CTEQ

nCTEQ
nuclear parton distribution functions

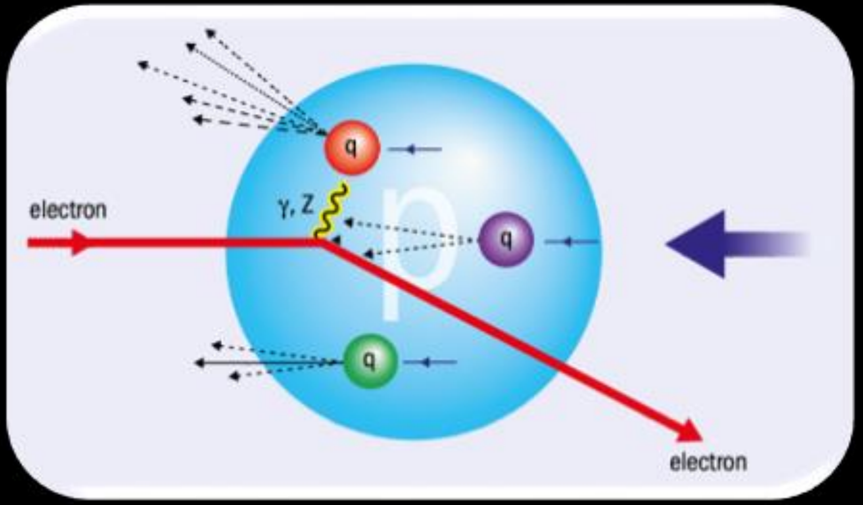
xFitter

SURGE COLLABORATION

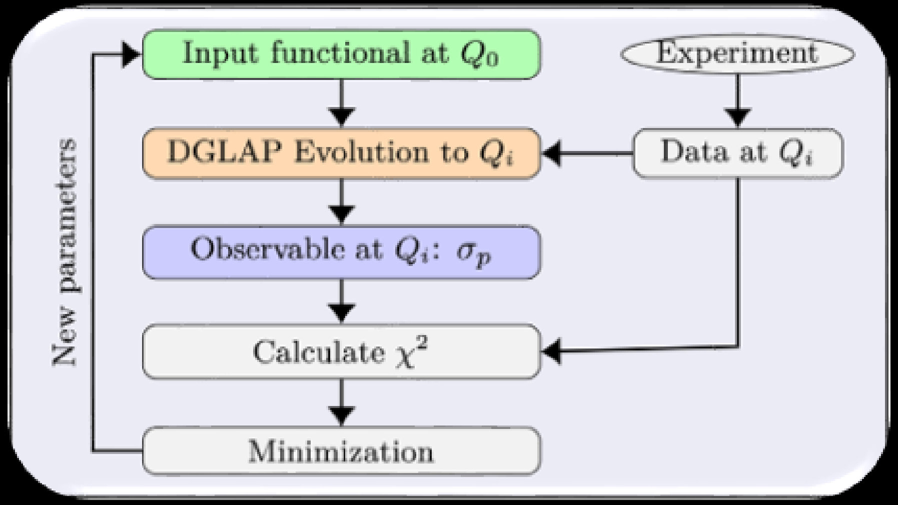
Linear QCD

$$\sigma_p = \left\{ \begin{array}{l} f(x,Q)\otimes\widehat{\sigma}_p \\ \text{non-perturbative part} \otimes \text{perturbative part} \end{array} \right.$$

- **Parton Model**
- **Established global analysis framework**



Schematic of DIS in the parton model

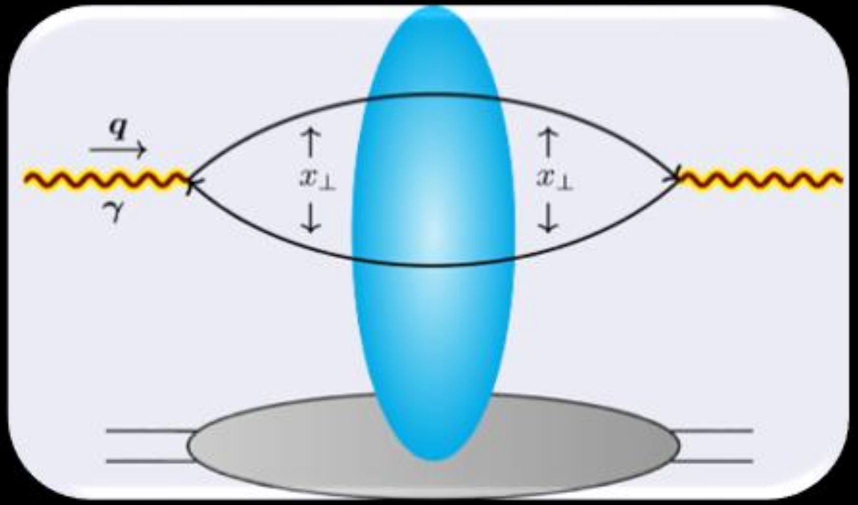


Parton fitting workflow

Nonlinear QCD

$$\sigma_d = \left\{ \begin{array}{l} \left|\Psi\left(\vec{x}_\perp, z\right)\right|^2 \otimes\widehat{\sigma}_d \\ \text{perturbative part} \otimes \text{non-perturbative part} \end{array} \right.$$

- **Dipole Model**
- **Global analysis framework is needed**



Schematic of DIS in the dipole model

Dipole fitting workflow

Step#1: Dipole Amplitude Models

- Compute dipole target scattering amplitude
- Various models can be used
- Each model depends on the set of different initial conditions
- Impact of the model parameters is also crucial

$$\sigma_d = \begin{cases} \left|\Psi\left(\vec{x}_\perp, z\right)\right|^2 \otimes \widehat{\sigma}_d \\ \text{perturbative part} \otimes \text{non-perturbative part} \end{cases}$$

MV$^\gamma$ Model

$$x = \frac{(r^2 Q_{s0}^2)^\gamma}{4} \ln\left(\frac{1}{r\Lambda_{\rm QCD}} + e\right)$$

$$N^{\rm MV^\gamma}(r) = \begin{cases} 1 - \exp(-x), & x > 10^{-6}, \\ x, & \text{otherwise.} \end{cases}$$

rcMV$^\gamma$ Model

$$x = \frac{r^2 Q_{s0}^2}{4} \alpha_s(r) \frac{N_c}{\pi} \left(1 + \ln \frac{\alpha_s^{\rm freeze}}{\alpha_s(r)}\right)$$

$$N^{\rm rcMV}(r) = \begin{cases} \left[1 - \exp\left(-x^\gamma\right)\right]^{1/\gamma}, & x > 10^{-4}, \\ x, & \text{otherwise.} \end{cases}$$

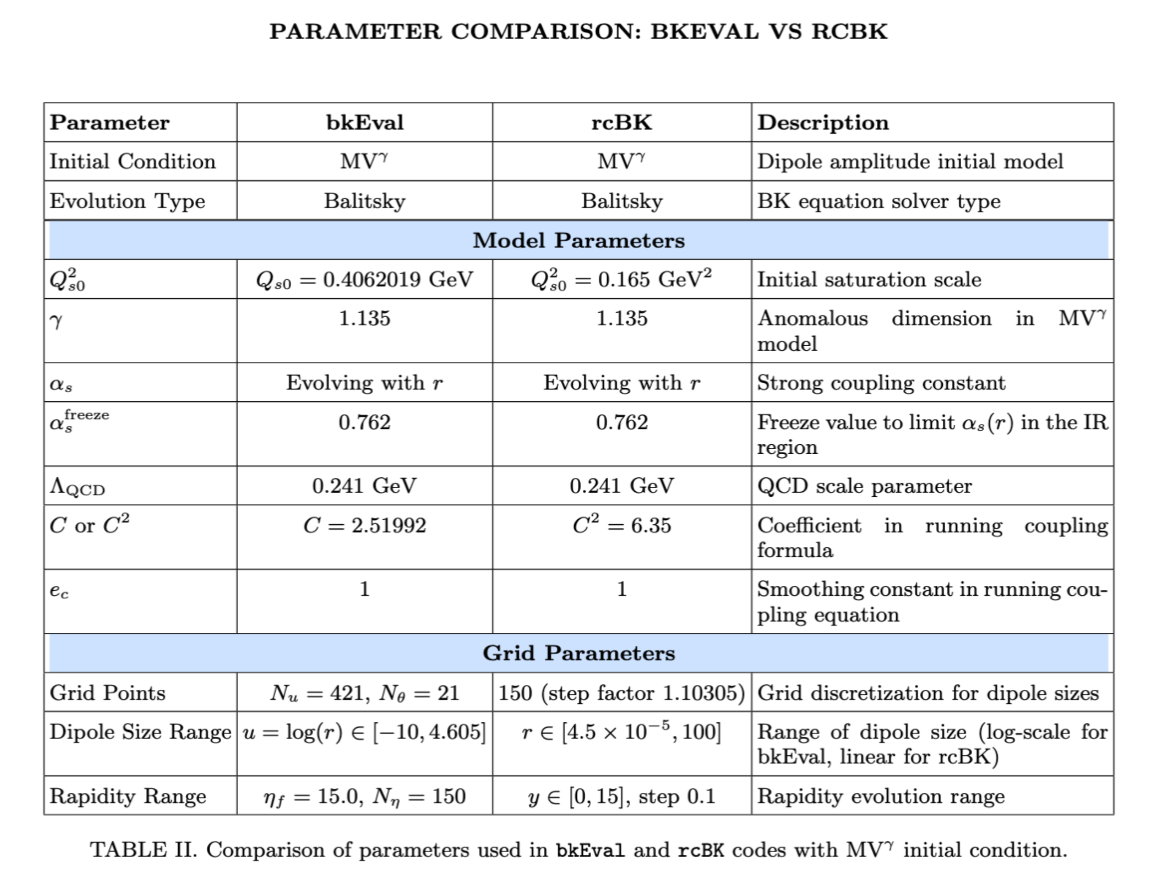
GBW Model

$$N^{\rm GBW}(r) = 1 - \exp\left(-\frac{r^2 Q_{s0}^2}{4}\right)$$

GBW$^\gamma$ Model

$$x = \left(\frac{r^2 Q_{s0}^2}{4}\right)^\gamma$$

$$N_0^{\rm GBW^\gamma}(r) = \begin{cases} \left[1 - \exp(-x)\right]^{1/\gamma}, & \frac{r^2 Q_{s0}^2}{4} > 10^{-4}, \\ \frac{r^2 Q_{s0}^2}{4}, & \text{otherwise.} \end{cases}$$



PARAMETER COMPARISON: BKEVAL VS RCBK

Parameter	bkEval	rcBK	Description
Initial Condition	MV$^\gamma$	MV$^\gamma$	Dipole amplitude initial model
Evolution Type	Balitsky	Balitsky	BK equation solver type
Model Parameters			
Q_{s0}^2	$Q_{s0} = 0.4062019$ GeV	$Q_{s0}^2 = 0.165$ GeV^2	Initial saturation scale
γ	1.135	1.135	Anomalous dimension in MV$^\gamma$ model
α_s	Evolving with r	Evolving with r	Strong coupling constant
$\alpha_s^{\rm freeze}$	0.762	0.762	Freeze value to limit $\alpha_s(r)$ in the IR region
$\Lambda_{\rm QCD}$	0.241 GeV	0.241 GeV	QCD scale parameter
C or C^2	$C = 2.51992$	$C^2 = 6.35$	Coefficient in running coupling formula
e_c	1	1	Smoothing constant in running coupling equation
Grid Parameters			
Grid Points	$N_u = 421,\ N_\theta = 21$	150 (step factor 1.10305)	Grid discretization for dipole sizes
Dipole Size Range	$u = \log(r) \in [-10, 4.605]$	$r \in [4.5 \times 10^{-5}, 100]$	Range of dipole size (log-scale for bkEval, linear for rcBK)
Rapidity Range	$\eta_f = 15.0,\ N_\eta = 150$	$y \in [0, 15]$, step 0.1	Rapidity evolution range

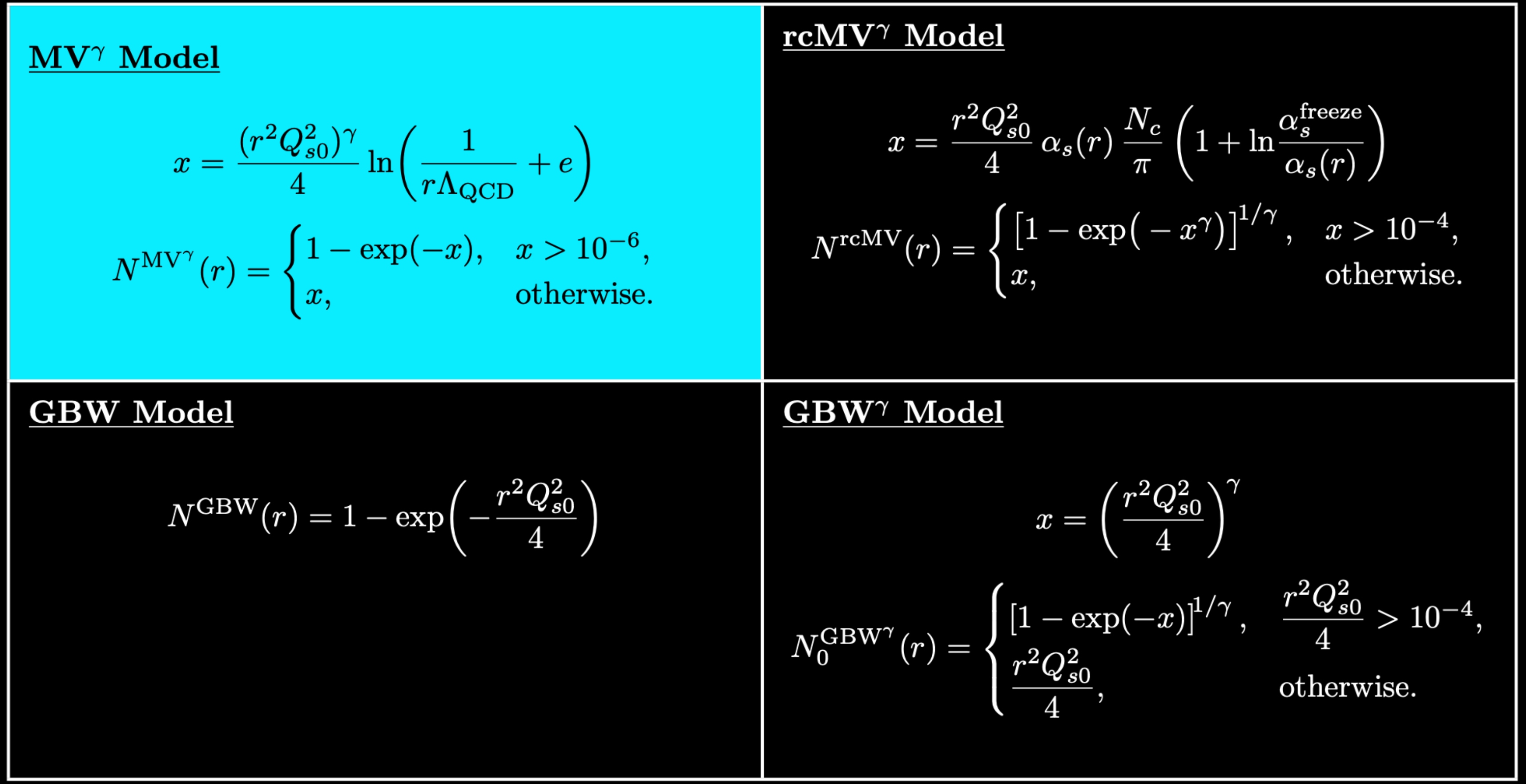
TABLE II. Comparison of parameters used in `bkEval` and `rcBK` codes with MV$^\gamma$ initial condition.

Golec-Biernat–Wusthöff (GBW)
McLerran-Venugopalan (MV)

Step#1: Dipole Amplitude Models

- Compute dipole target scattering amplitude
- Various models can be used
- Each model depends on the set of different initial conditions
- Impact of the model parameters is also crucial

$$\sigma_d = \begin{cases} \left|\Psi\left(\vec{x}_\perp, z\right)\right|^2 \otimes \boxed{\widehat{\sigma}_d} \\ \text{perturbative part} \otimes \text{non-perturbative part} \end{cases}$$



MV$^\gamma$ Model

$$x = \frac{(r^2 Q_{s0}^2)^\gamma}{4} \ln\left(\frac{1}{r\Lambda_{\mathrm{QCD}}} + e\right)$$

$$N^{\mathrm{MV}^\gamma}(r) = \begin{cases} 1 - \exp(-x), & x > 10^{-6}, \\ x, & \text{otherwise.} \end{cases}$$

rcMV$^\gamma$ Model

$$x = \frac{r^2 Q_{s0}^2}{4}\, \alpha_s(r)\, \frac{N_c}{\pi} \left(1 + \ln \frac{\alpha_s^{\mathrm{freeze}}}{\alpha_s(r)}\right)$$

$$N^{\mathrm{rcMV}}(r) = \begin{cases} \left[1 - \exp\left(-x^\gamma\right)\right]^{1/\gamma}, & x > 10^{-4}, \\ x, & \text{otherwise.} \end{cases}$$

GBW Model

$$N^{\mathrm{GBW}}(r) = 1 - \exp\left(-\frac{r^2 Q_{s0}^2}{4}\right)$$

GBW$^\gamma$ Model

$$x = \left(\frac{r^2 Q_{s0}^2}{4}\right)^\gamma$$

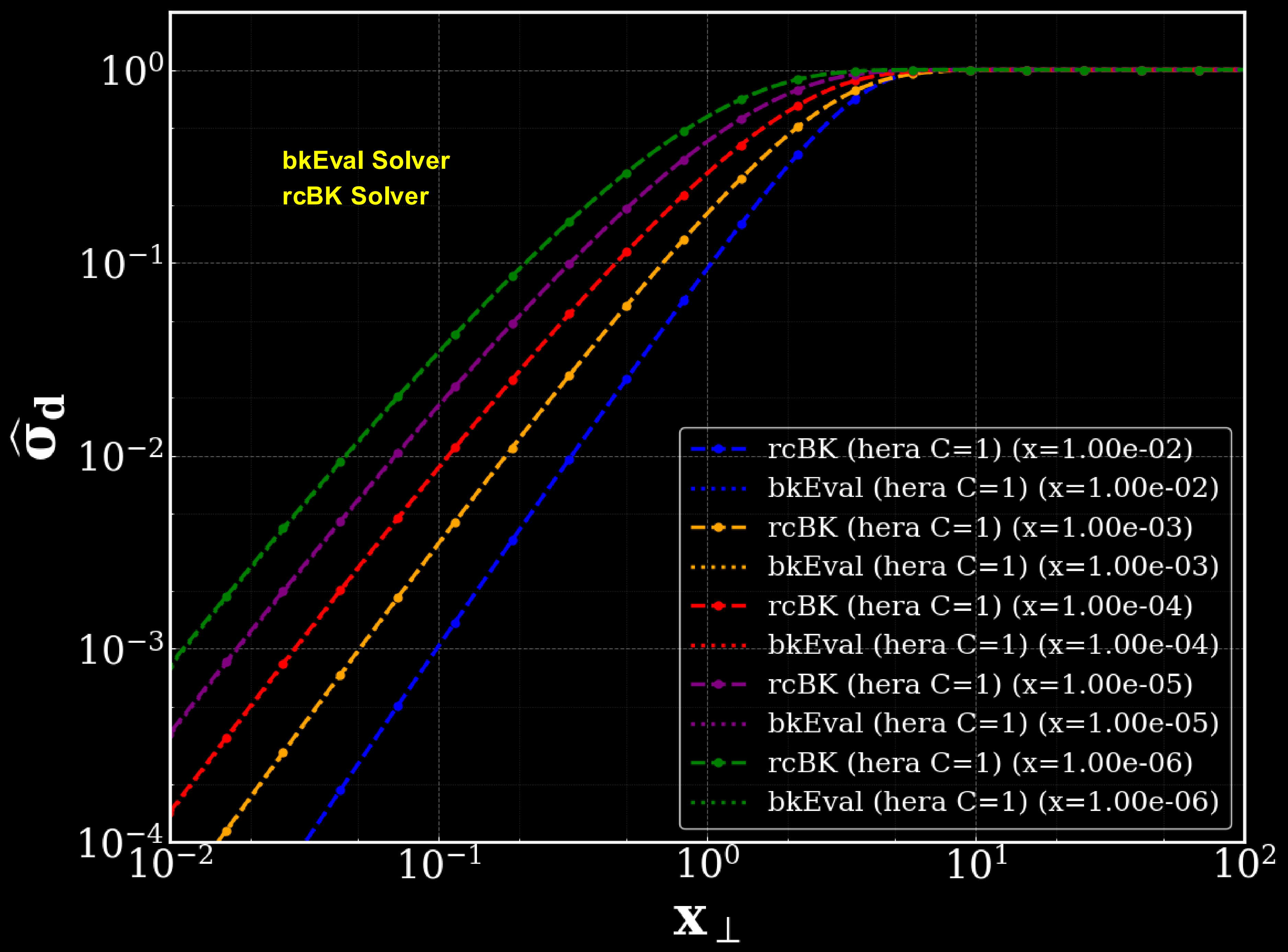
$$N_0^{\mathrm{GBW}^\gamma}(r) = \begin{cases} \left[1 - \exp(-x)\right]^{1/\gamma}, & \frac{r^2 Q_{s0}^2}{4} > 10^{-4}, \\ \frac{r^2 Q_{s0}^2}{4}, & \text{otherwise.} \end{cases}$$

Golec-Biernat–Wusthöff (GBW)
McLerran-Venugopalan (MV)

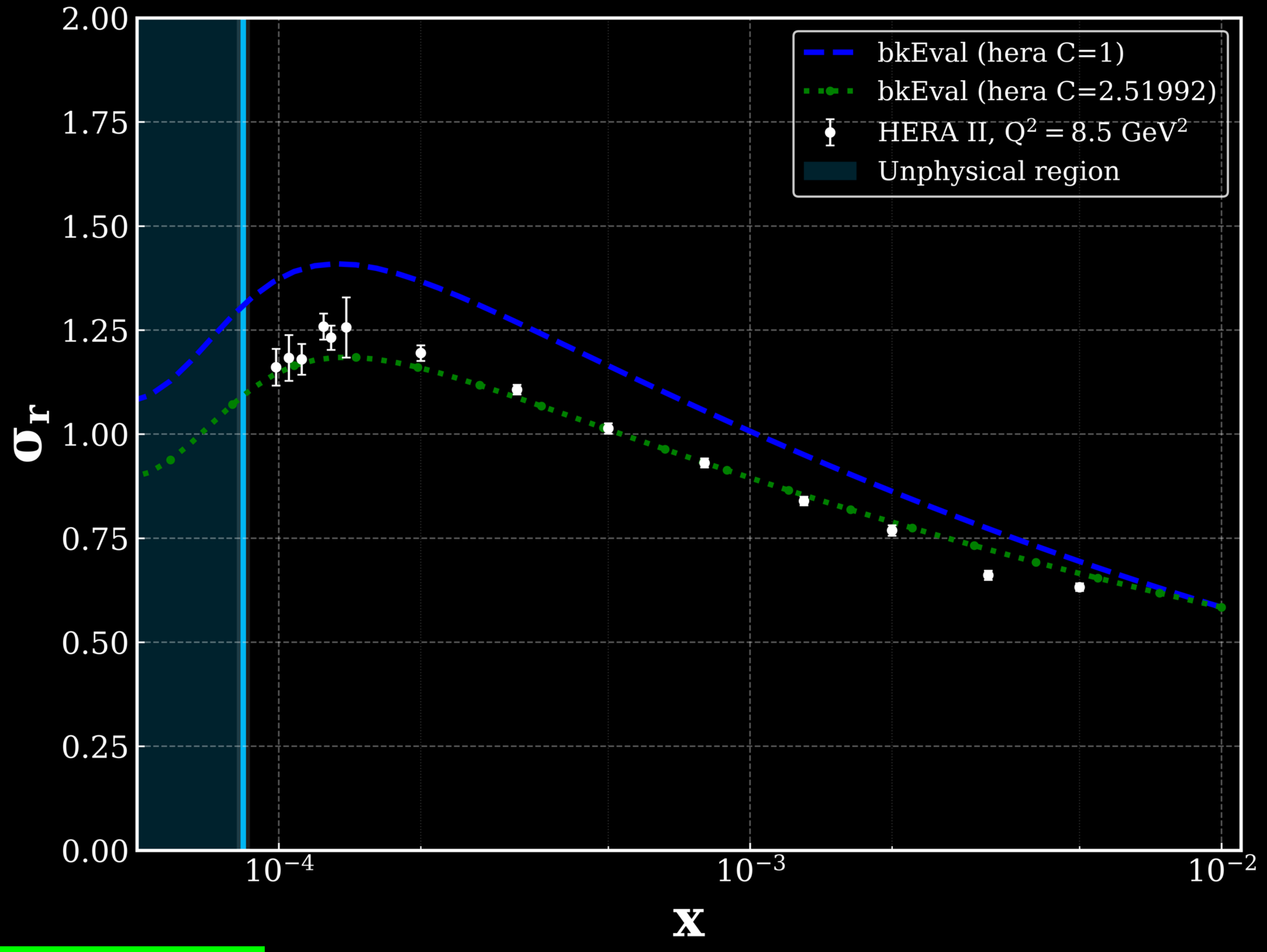
Two different code pipelines

1. bkEval Solver *(Paul Caucal)*	2. rcBK Solver *(Heikki Mäntysaari)*

Step#1: Dipole Amplitude Results



Experimental Validation

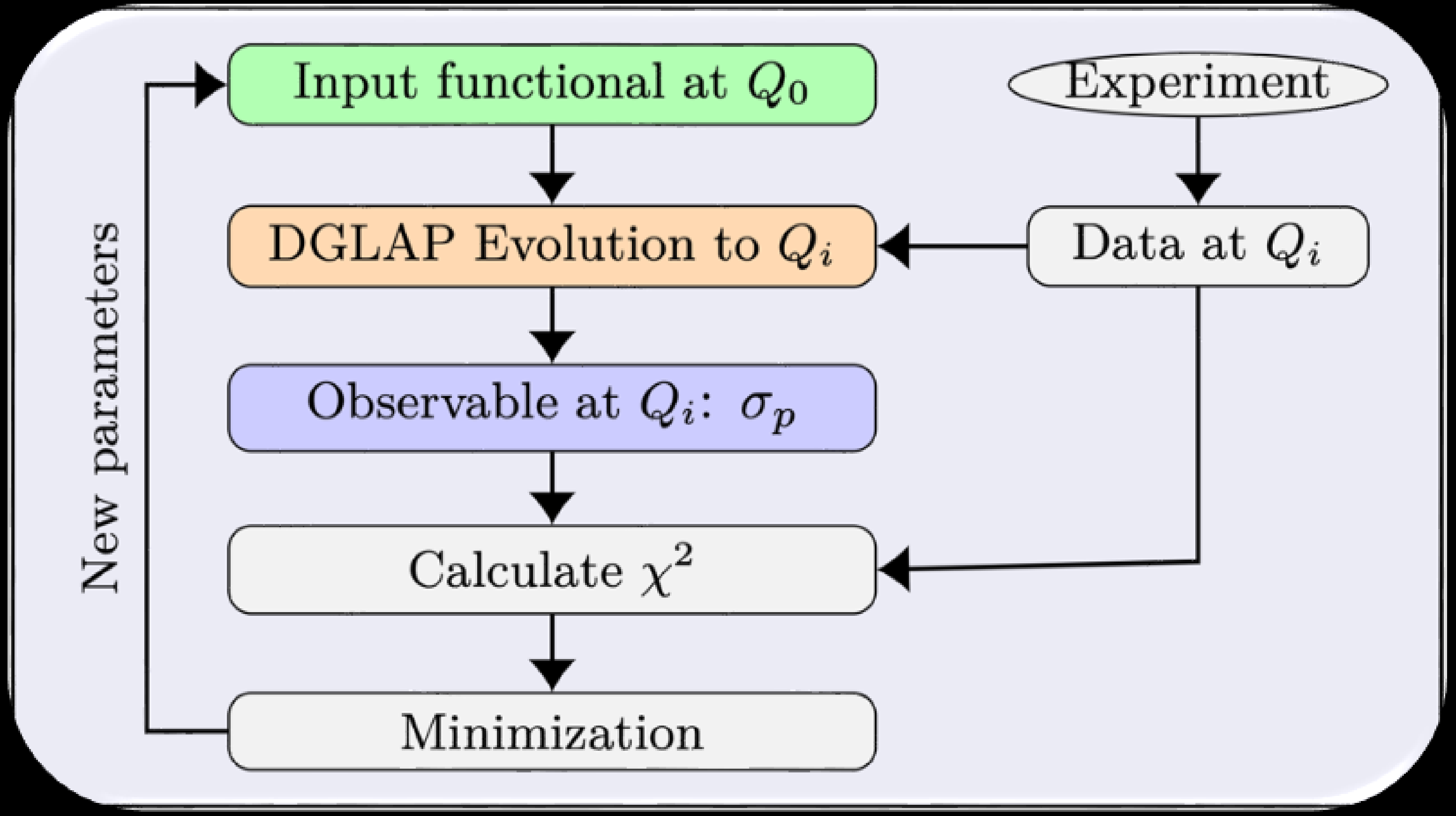


Good agreement

QCD Fitting Workflows

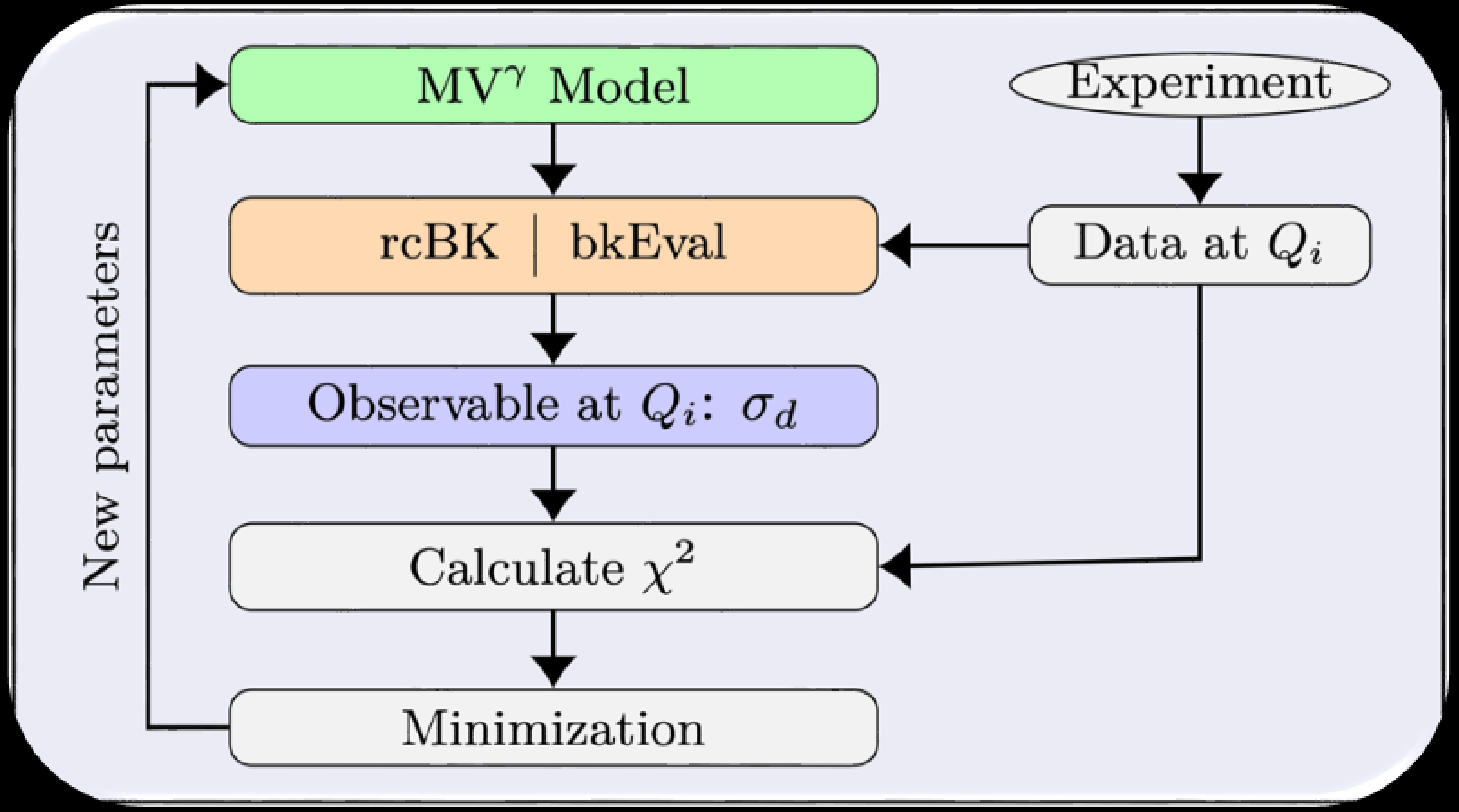
Parton fitting workflow

Established



Dipole fitting workflow

Prototype

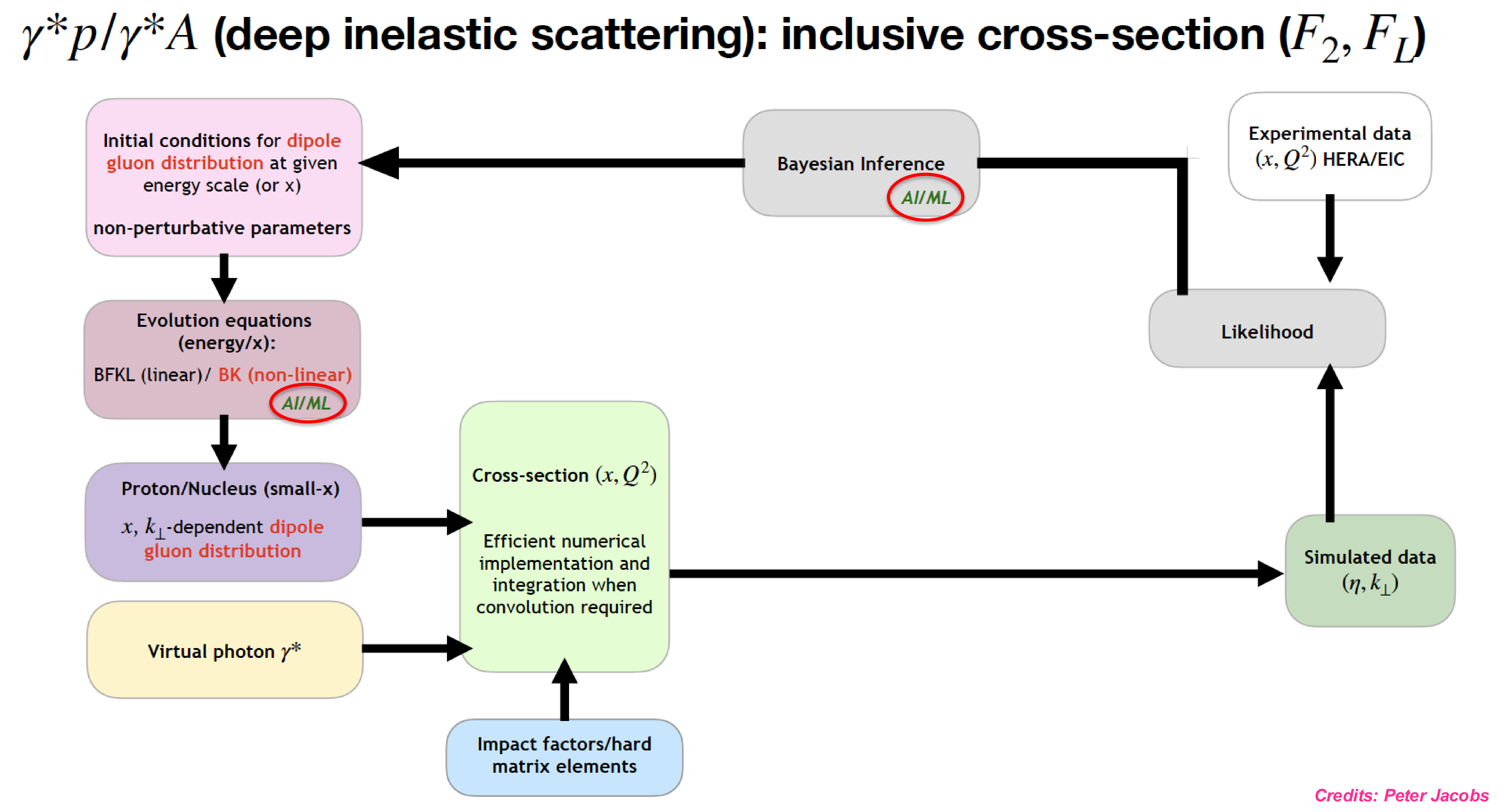


- Good agreement with HERA experimental data
- Not fast enough for global inner fitting loop

Need for Machine Learning Approach

- **Why?**
  - Global fits ≈ 5,000 data points × up to ≈ 3,000 iterations → millions of evaluations.
- **Goal**
  - Fast ML surrogate emulating nonlinear evolution while preserving physics
- **Outcomes**
  - Orders-of-magnitude faster evaluations → timely, iterative comparisons to data.
  - Simplified access to comparison between parton and dipole models.

ML Fits in Global Analysis Workflow



$\gamma^* p / \gamma^* A$ (deep inelastic scattering): inclusive cross-section (F_2, F_L)

Initial conditions for dipole gluon distribution at given energy scale (or x)

non-perturbative parameters

Evolution equations (energy/x):

BFKL (linear)/ BK (non-linear)

AI/ML

Proton/Nucleus (small-x)

$x, k_\perp$-dependent dipole gluon distribution

Virtual photon γ^*

Cross-section (x, Q^2)

Efficient numerical implementation and integration when convolution required

Impact factors/hard matrix elements

Bayesian Inference

AI/ML

Experimental data (x, Q^2) HERA/EIC

Likelihood

Simulated data $(\eta, k_\perp)$

Credits: Peter Jacobs

Machine Learning Surrogate Models

1. **Multi-Layer Perceptron (MLP) Model**
2. **DeepONet Model**

1- Multi-Layer Perceptron (MLP) Model:

Junaid

Neural Network Setup

Input features: x, r, C, Qs0, ɣ

Input model: McLerran-Venugopalan

Target: N(x,r) dipole cross section

Architecture: 5 hidden layers (512, 512, 256, 256, 128 neurons) with GeLU activation

Output layer: Linear

Training Details

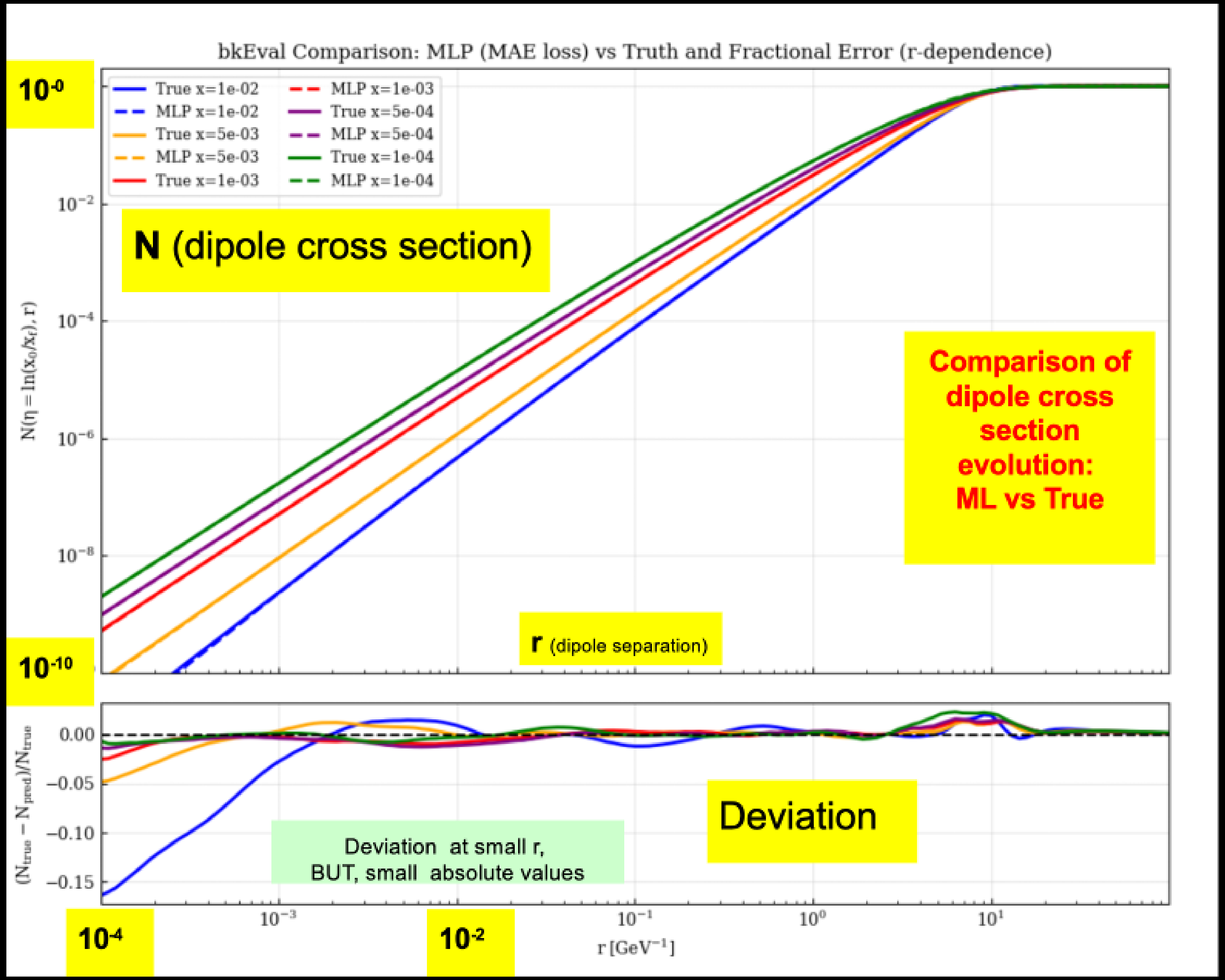
Optimizer: AdamW

AdamW adaptively adjusts the learning rate for each parameter based on estimates of the first and second moments of the gradients. This allows for faster convergence and can handle sparse gradients effectively.

Learning rate: Exponentially decaying

Loss function: MAE

Dataset split: 80/10/10 (train/validation/test)



N (dipole cross section)

Comparison of dipole cross section evolution: ML vs True

r (dipole separation)

Deviation

Deviation at small r, BUT, small absolute values

2- DeepONet Model

Brandon

Neural Network Setup

With test dataset, Varying **C** and **QS0** and 11,998,500 data points:

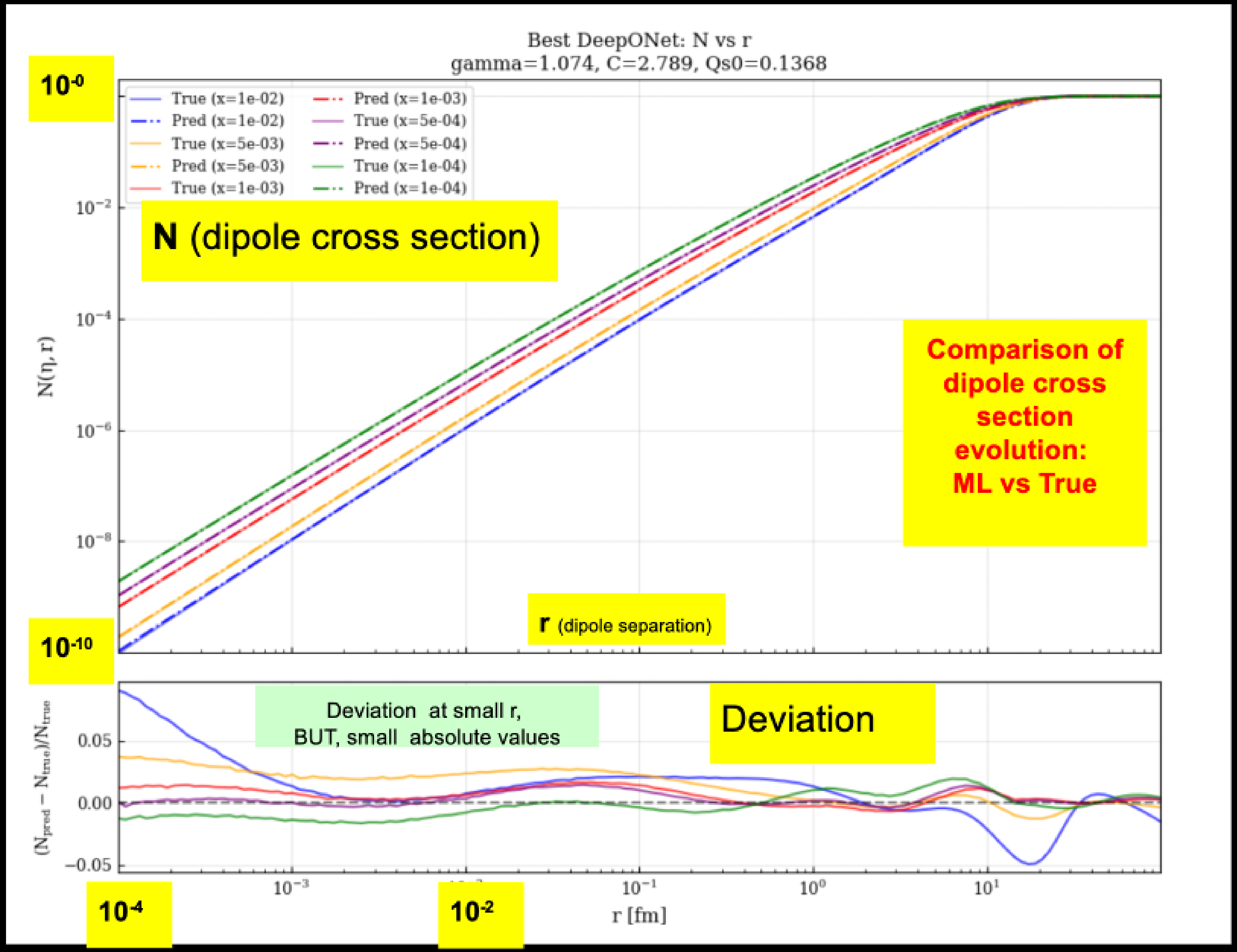
Training time: ~1 hour, 100 epochs

Size of keras model save: 11mb

Inference time: less than 1 second for full grid

"**DeepONet:** Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators" https://arxiv.org/pdf/1910.03193

Two sub-networks a branch and trunk. Branch and Trunk network can be any network, MLP (dense connection) chosen for simplicity. Trunk network also uses a simple fourier features layer to give access to high dimensional information.



10^{-0}

N (dipole cross section)

Comparison of dipole cross section evolution: ML vs True

r (dipole separation)

10^{-10}

Deviation at small r, BUT, small absolute values

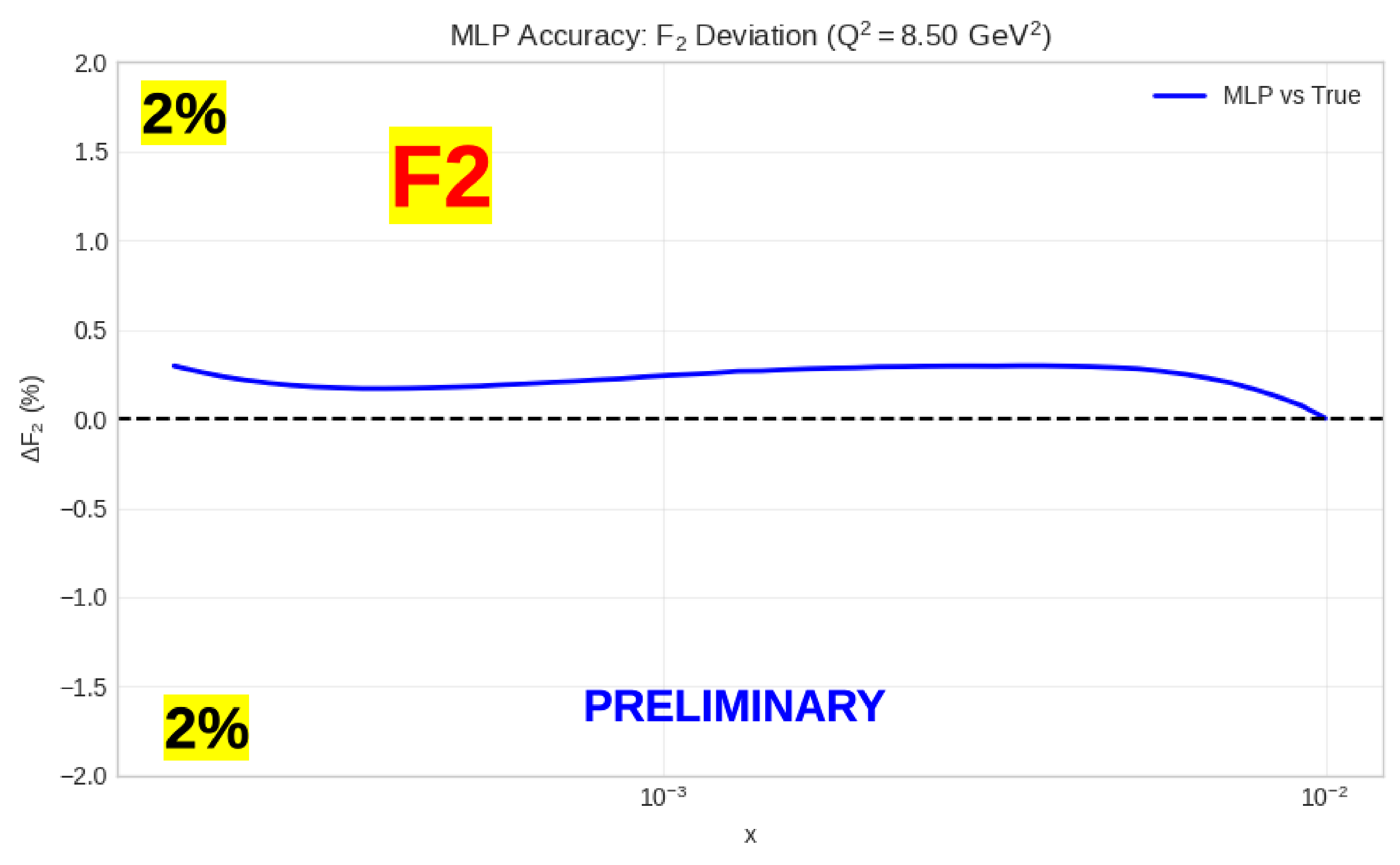
Deviation

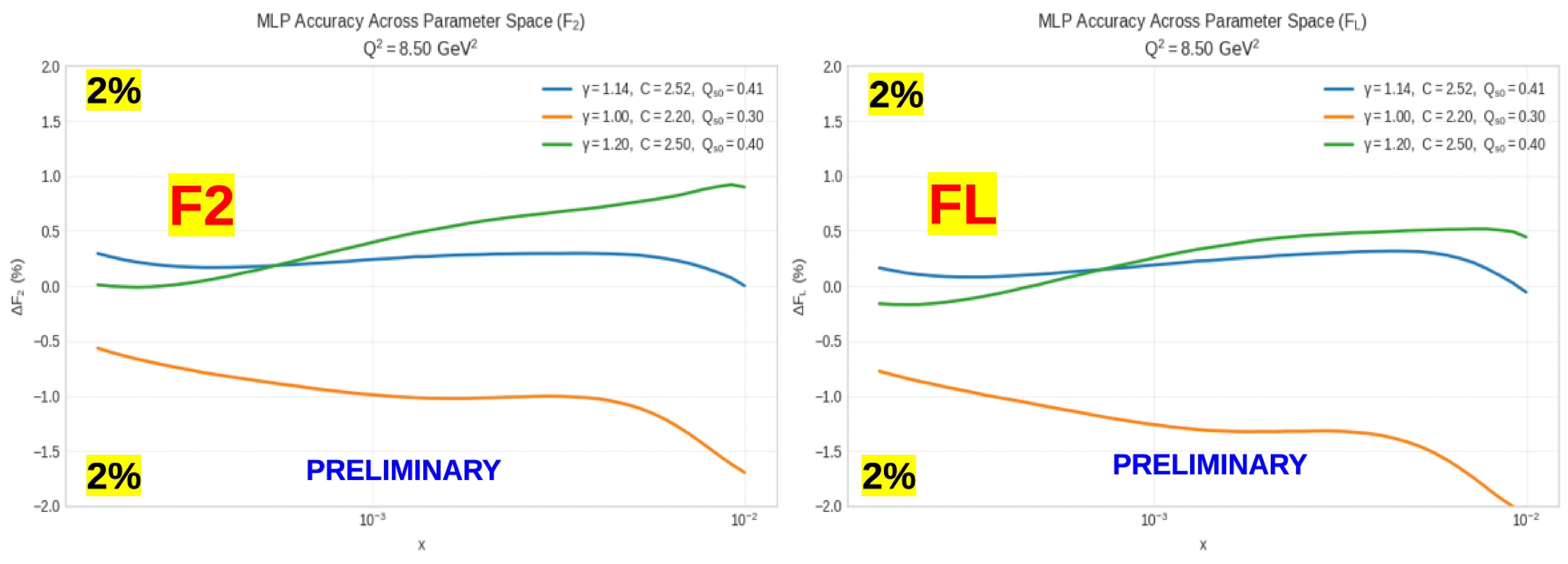
10^{-4}

10^{-2}

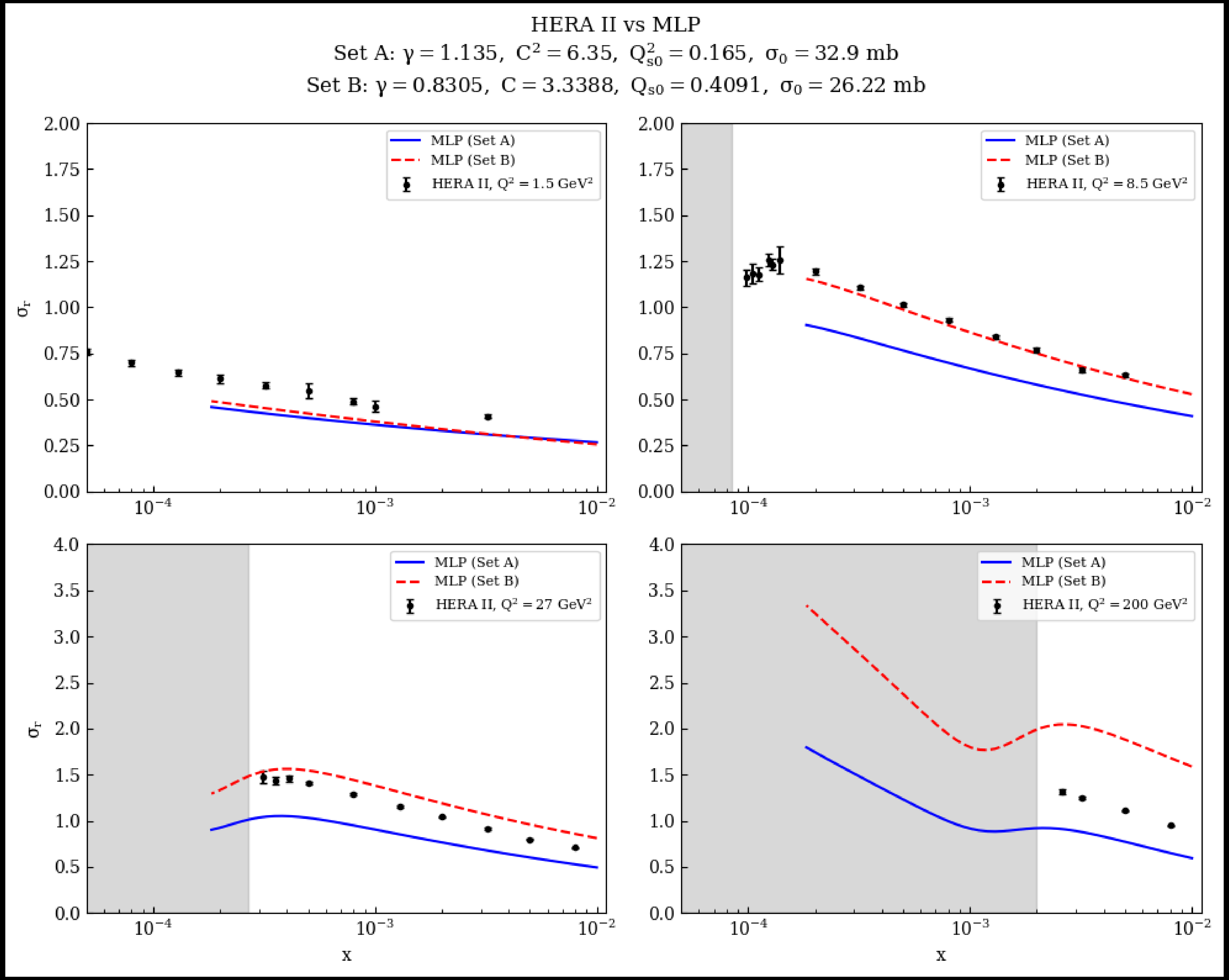
Uncertainties: ML vs. True

Preliminary uncertainties appear to be satisfactory given typical theoretical and experimental uncertainties



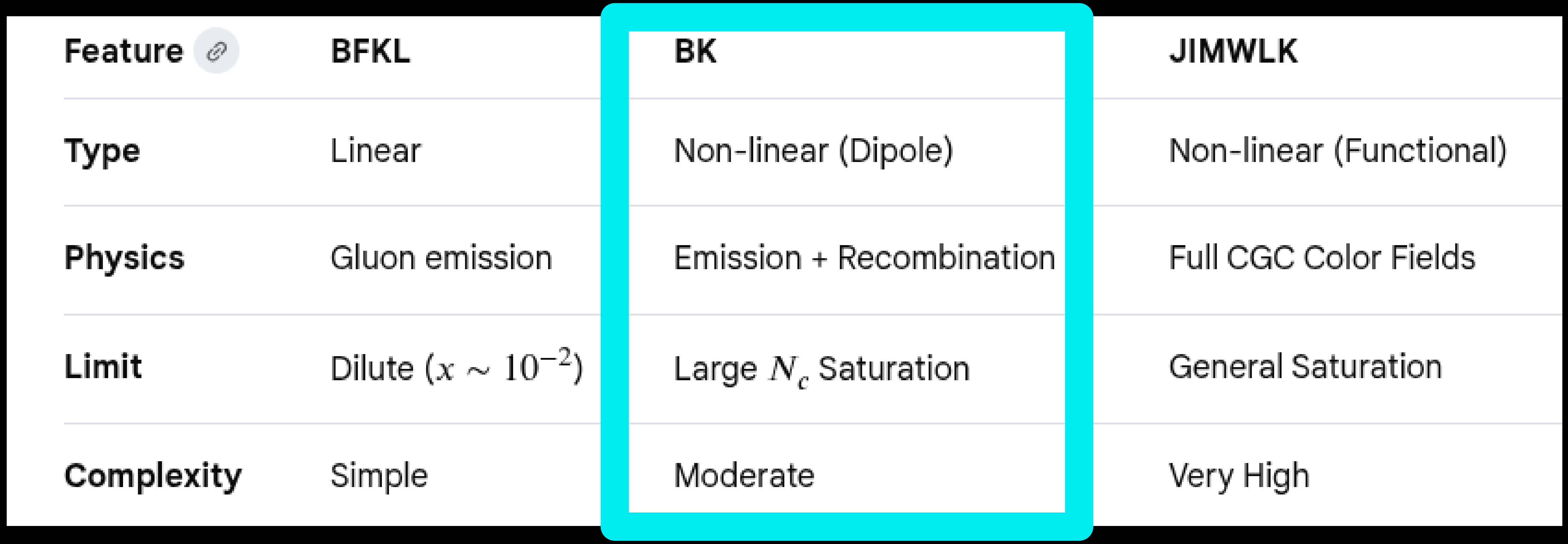


ML Results with HERA



Next Steps:

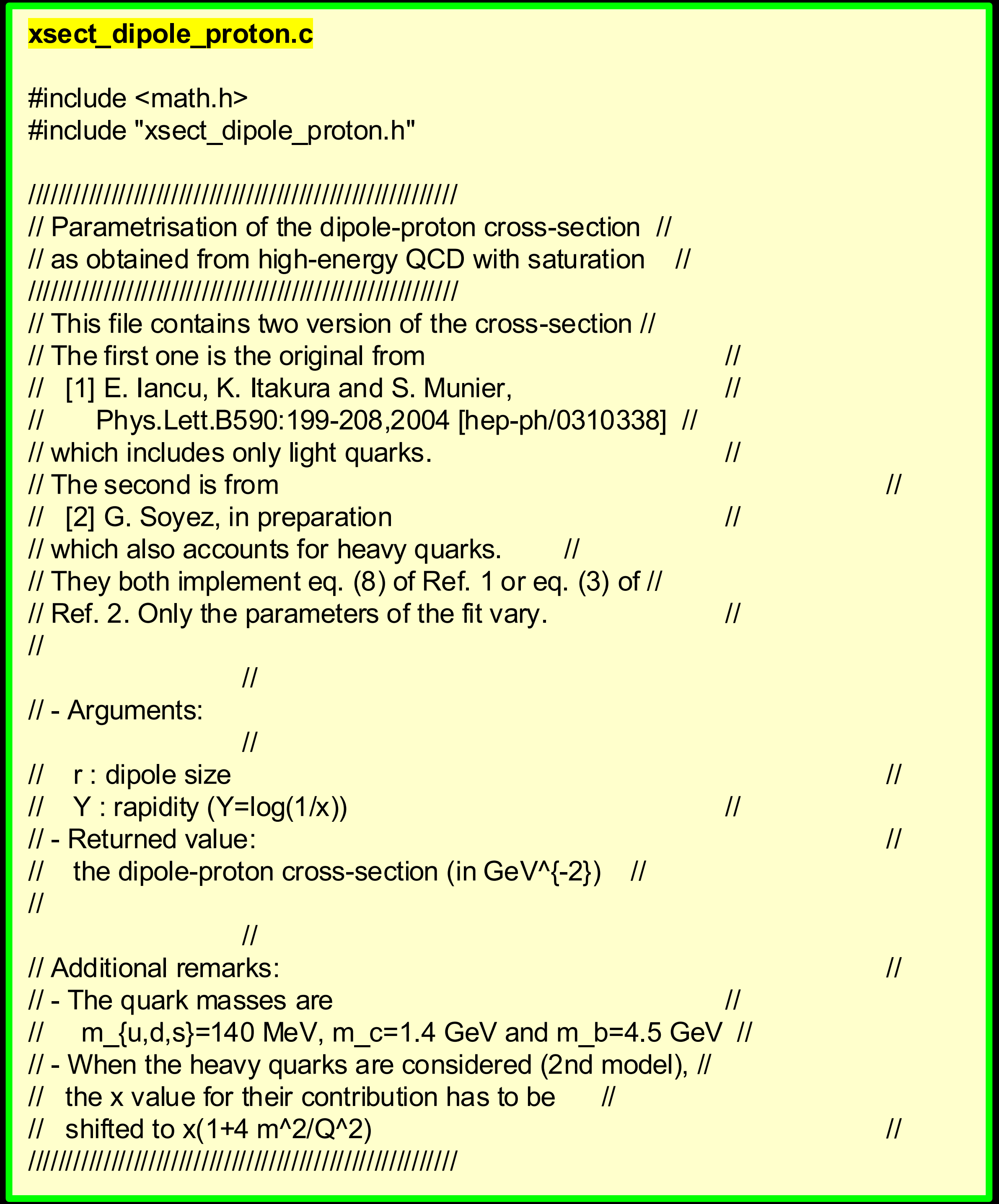
- **Other evolution models:**
    - Could extend to other models: BFKL, JIMWLK, … DGLAP??
    - Could ‘warm-up’ running BK with LO & NLO



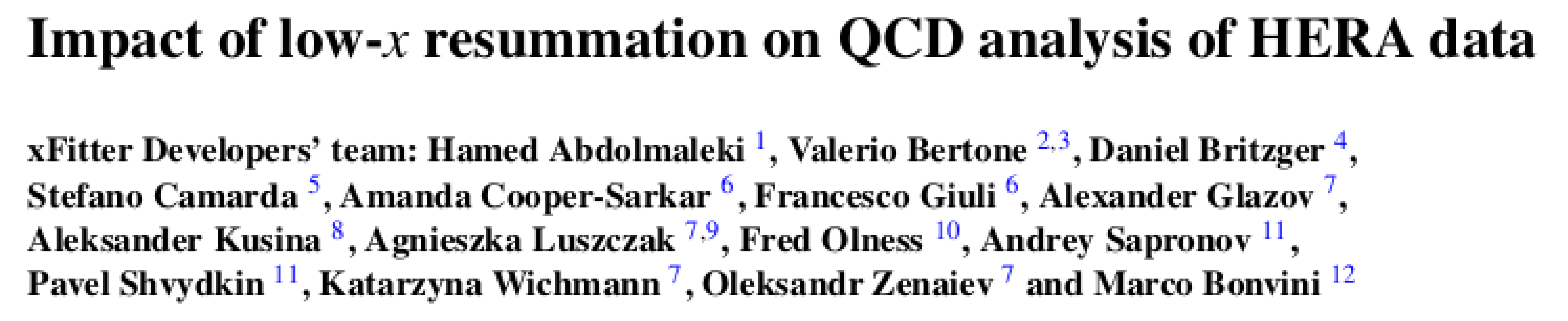
Feature	BFKL	BK	JIMWLK
Type	Linear	Non-linear (Dipole)	Non-linear (Functional)
Physics	Gluon emission	Emission + Recombination	Full CGC Color Fields
Limit	Dilute ($x \sim 10^{-2}$)	Large N_c Saturation	General Saturation
Complexity	Simple	Moderate	Very High

Next Steps:

- **Integration with xFitter:**

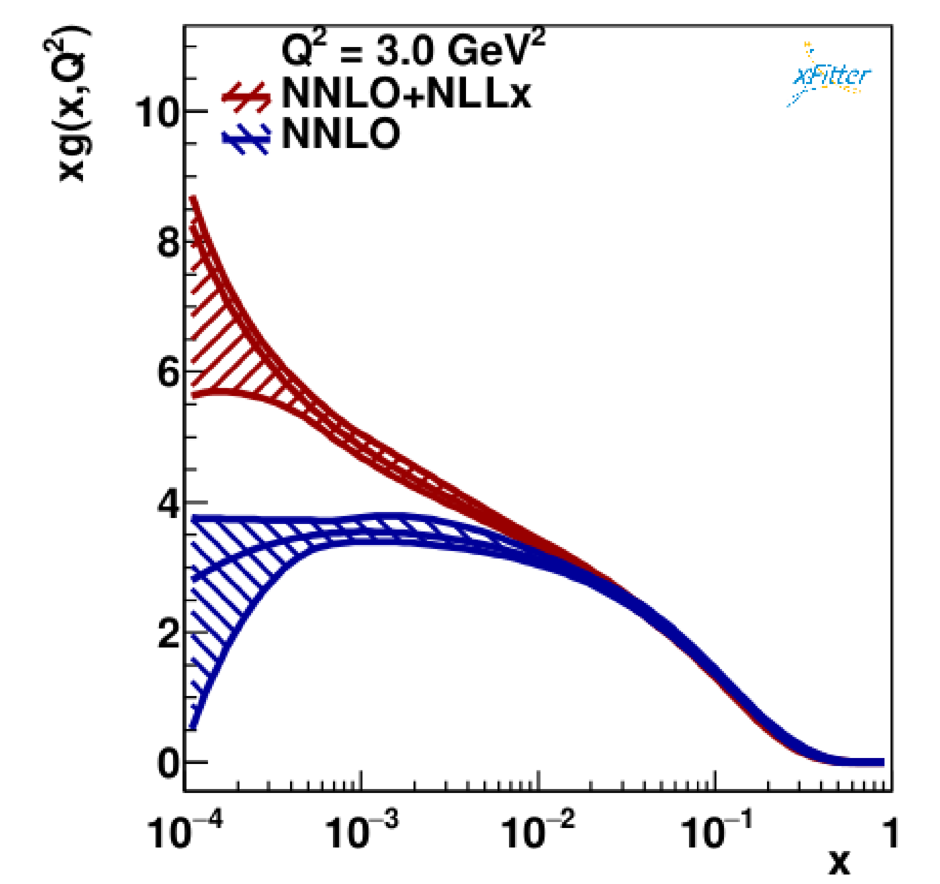


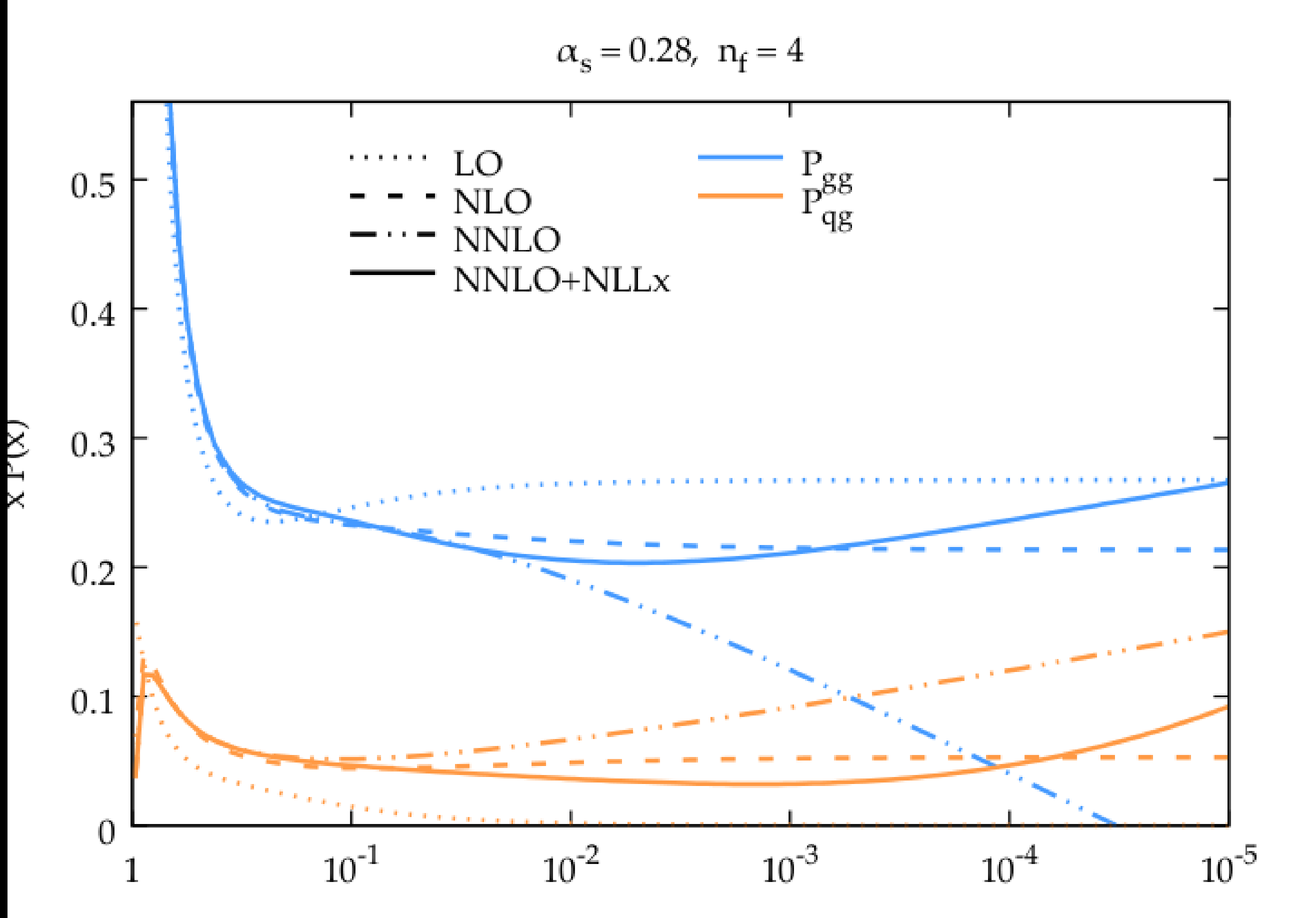
```
xsect_dipole_proton.c

#include <math.h>
#include "xsect_dipole_proton.h"

////////////////////////////////////////////////////////////
// Parametrisation of the dipole-proton cross-section  //
// as obtained from high-energy QCD with saturation    //
////////////////////////////////////////////////////////////
// This file contains two version of the cross-section //
// The first one is the original from                   //
//   [1] E. Iancu, K. Itakura and S. Munier,            //
//       Phys.Lett.B590:199-208,2004 [hep-ph/0310338]  //
// which includes only light quarks.                    //
// The second is from                                           //
//   [2] G. Soyez, in preparation                       //
// which also accounts for heavy quarks.        //
// They both implement eq. (8) of Ref. 1 or eq. (3) of //
// Ref. 2. Only the parameters of the fit vary.         //
//
                        //
// - Arguments:
                        //
//    r : dipole size                                           //
//    Y : rapidity (Y=log(1/x))                         //
// - Returned value:                                            //
//    the dipole-proton cross-section (in GeV^{-2})    //
//
                        //
// Additional remarks:                                          //
// - The quark masses are                               //
//     m_{u,d,s}=140 MeV, m_c=1.4 GeV and m_b=4.5 GeV  //
// - When the heavy quarks are considered (2nd model), //
//   the x value for their contribution has to be      //
//   shifted to x(1+4 m^2/Q^2)                                  //
////////////////////////////////////////////////////////////
```



Impact of low-*x* resummation on QCD analysis of HERA data

xFitter Developers' team: Hamed Abdolmaleki [1], Valerio Bertone [2,3], Daniel Britzger [4], Stefano Camarda [5], Amanda Cooper-Sarkar [6], Francesco Giuli [6], Alexander Glazov [7], Aleksander Kusina [8], Agnieszka Luszczak [7,9], Fred Olness [10], Andrey Sapronov [11], Pavel Shvydkin [11], Katarzyna Wichmann [7], Oleksandr Zenaiev [7] and Marco Bonvini [12]

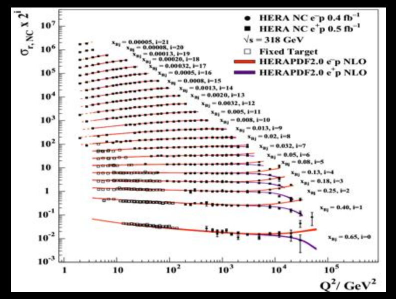




Brief Introduction to xFitter:

xFitter
www.xfitter.org

Experimental Data

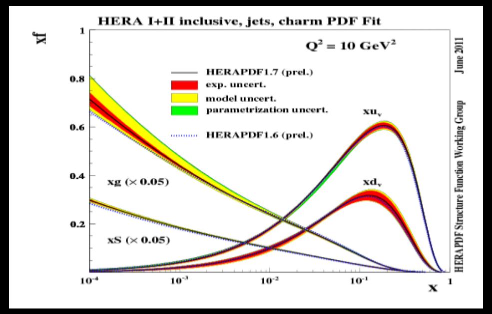


Data: HERA, Tevatron, LHC, fixed target experiments
Processes:
Inclusive DIS, Jets, Drell-Yan, Diffraction, Top production
W and Z production

Theory Calculations

HQ Schemes:	MSTW, NNPDF, ABM, ACOT
Jets, W, Z:	FastNLO, ApplGrid
Top:	Hathor
Evolution:	QCDNUM, APFEL, k_T
Other:	NNPDF reweighting
	TMDs, Dipole Model, ...

xFitter



Parton Distribution Functions

$\alpha_S(M_Z)$, m_c,m_b,m_t ...

Theoretical Cross Sections

Comparisons to other PDFs (LHAPDF)

Features & Recent Updates:

NNLO DGLAP	Heavy Quark Variable Threshold
Photon PDF & **QED**	Improvements in A2 and correlations
Pole & MS-bar masses	**TMDs**, PDFs (uPDFs)
Profiling and Re-Weighting	…and many other
BFKL interface	

extensions include nuclear PDFs

xFitter 2.2.0 Future Freeze

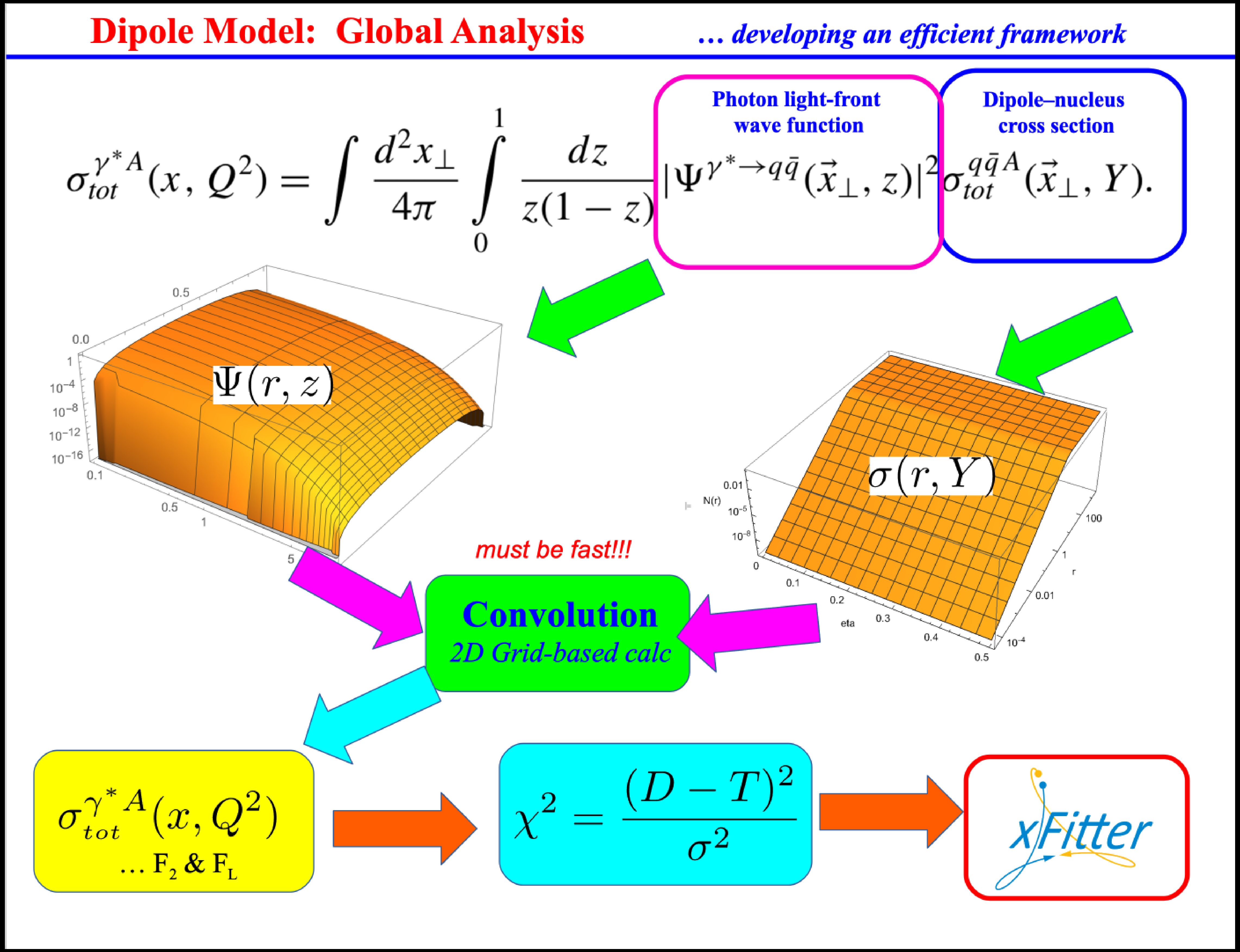
Brief Introduction to xFitter:

www.xfitter.org


xFitter Collaboration Meeting
February 2020, DESY

The xFitter Team

Big Picture:



Dipole Model: Global Analysis

… developing an efficient framework

Photon light-front wave function

Dipole–nucleus cross section

$$\sigma_{tot}^{\gamma^* A}(x, Q^2) = \int \frac{d^2 x_\perp}{4\pi} \int_0^1 \frac{dz}{z(1-z)} \boxed{|\Psi^{\gamma^* \to q\bar{q}}(\vec{x}_\perp, z)|^2} \boxed{\sigma_{tot}^{q\bar{q}A}(\vec{x}_\perp, Y).}$$

$\Psi(r, z)$

$\sigma(r, Y)$

must be fast!!!

Convolution

2D Grid-based calc

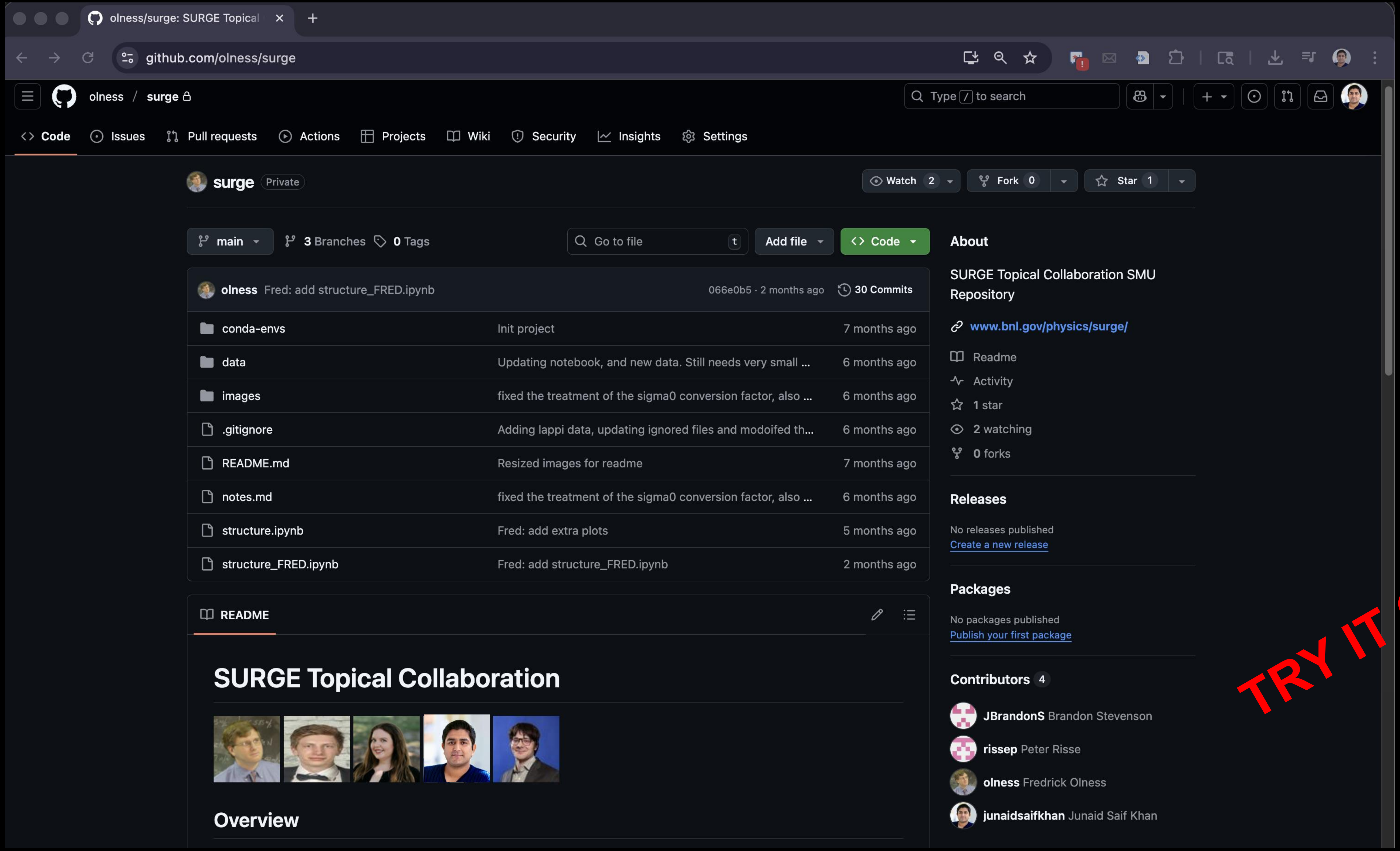
$\sigma_{tot}^{\gamma^* A}(x, Q^2)$

… F_2 & F_L

$$\chi^2 = \frac{(D-T)^2}{\sigma^2}$$

xFitter

Open source, reproduceable workflow



olness Fred: add structure_FRED.ipynb 066e0b5 · 2 months ago 30 Commits

conda-envs	Init project	7 months ago
data	Updating notebook, and new data. Still needs very small ...	6 months ago
images	fixed the treatment of the sigma0 conversion factor, also ...	6 months ago
.gitignore	Adding lappi data, updating ignored files and modoifed th...	6 months ago
README.md	Resized images for readme	7 months ago
notes.md	fixed the treatment of the sigma0 conversion factor, also ...	6 months ago
structure.ipynb	Fred: add extra plots	5 months ago
structure_FRED.ipynb	Fred: add structure_FRED.ipynb	2 months ago

SURGE Topical Collaboration

Overview

About

SURGE Topical Collaboration SMU Repository

www.bnl.gov/physics/surge/

Readme
Activity
1 star
2 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 4

JBrandonS Brandon Stevenson
rissep Peter Risse
olness Fredrick Olness
junaidsaifkhan Junaid Saif Khan

TRY IT OUT!

Paper is in progress

Summary:

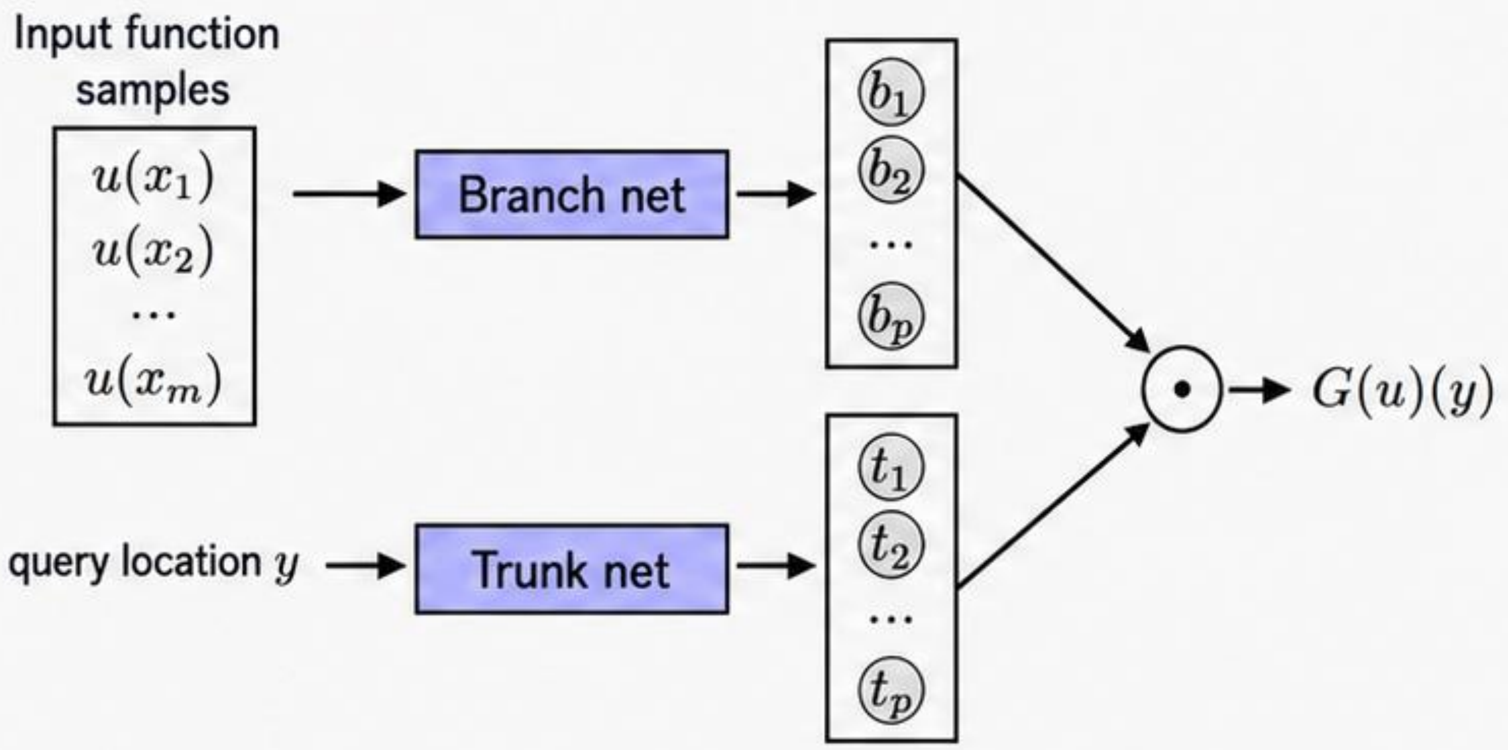
- **Dipole Calculation:**
  - Working Python framework to compute σtot, F2, and FL
  - Reasonable agreement with e.g., HERA data
  - Grid methods enable fast calculation
- **BK Evolution:**
  - Full evolution codes: [BKsolver, rcbk] validated
  - not fast enough for inner fitting loop
- **ML approach to BK Evolution** *(new)*
  - Reasonable approximation across kinematics [preliminary]
  - Fast enough for flexible fits
- **Other evolution models:** *(planned)*
  - Could extend to other models: BFKL, JIMWLK, … DGLAP??
  - Could 'warm-up' running BK with LO & NLO
- **Integration with xFitter:** *(planned)*

Thank You

Backup Slides

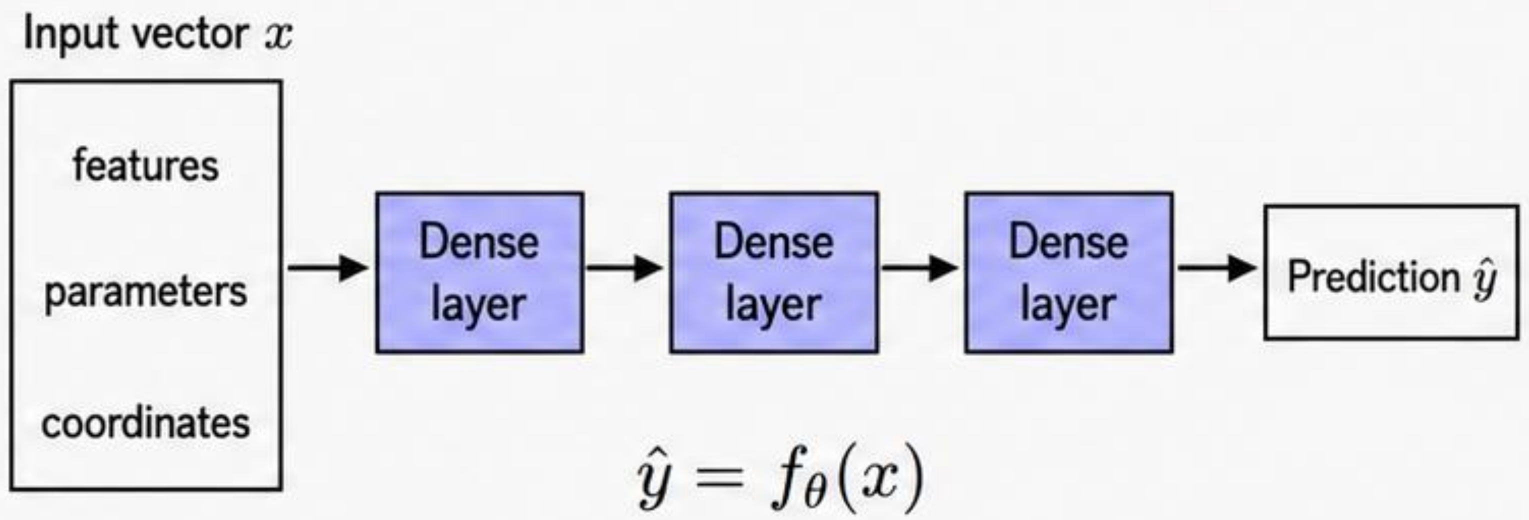
DeepONet vs MLP: Graphical Comparison

DeepONet (Operator Learning)



- Learns operator: function → function
- Uses branch + trunk networks
- Supports off-grid evaluation

MLP (Pointwise Mapping)



$$\hat{y} = f_\theta(x)$$

- Learns mapping: vector → vector/value
- Single feedforward network
- Usually fixed-size, grid-dependent input

Input:
Function samples + query y vs Feature vector x

Architecture:
Branch + trunk vs Stacked dense layers

Best for:
Operator learning / PDE surrogates vs Standard regression / classification

Generalization:
Off-grid evaluation vs Direct pointwise prediction

DeepONet: Operator Learning Framework

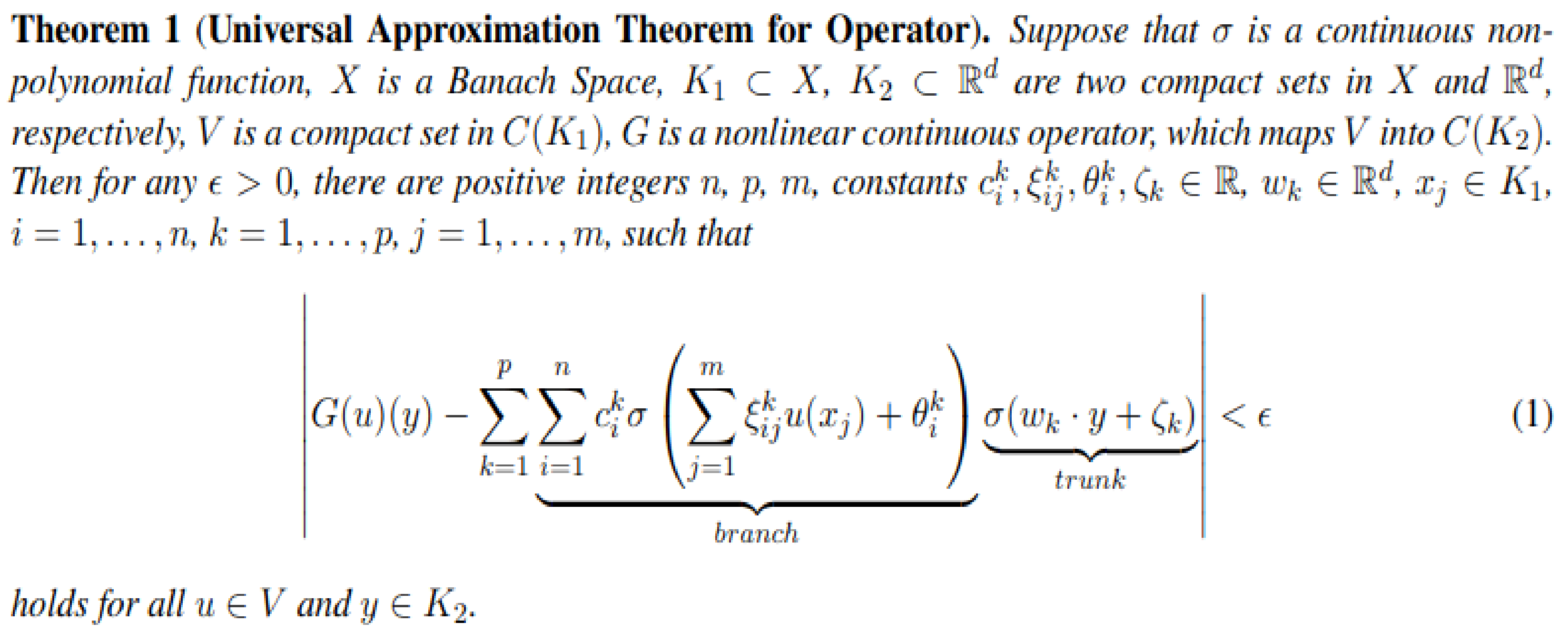
Scientific Context

DeepONet (Lu, Jin, and Karniadakis) learns operators from grids of points and values. Leveraging the universal approximation theorem, it enables "off-grid" learning, a key advantage over traditional ML methods.

Dual-Network Architecture

Branch Network: Holds the input function/grid.

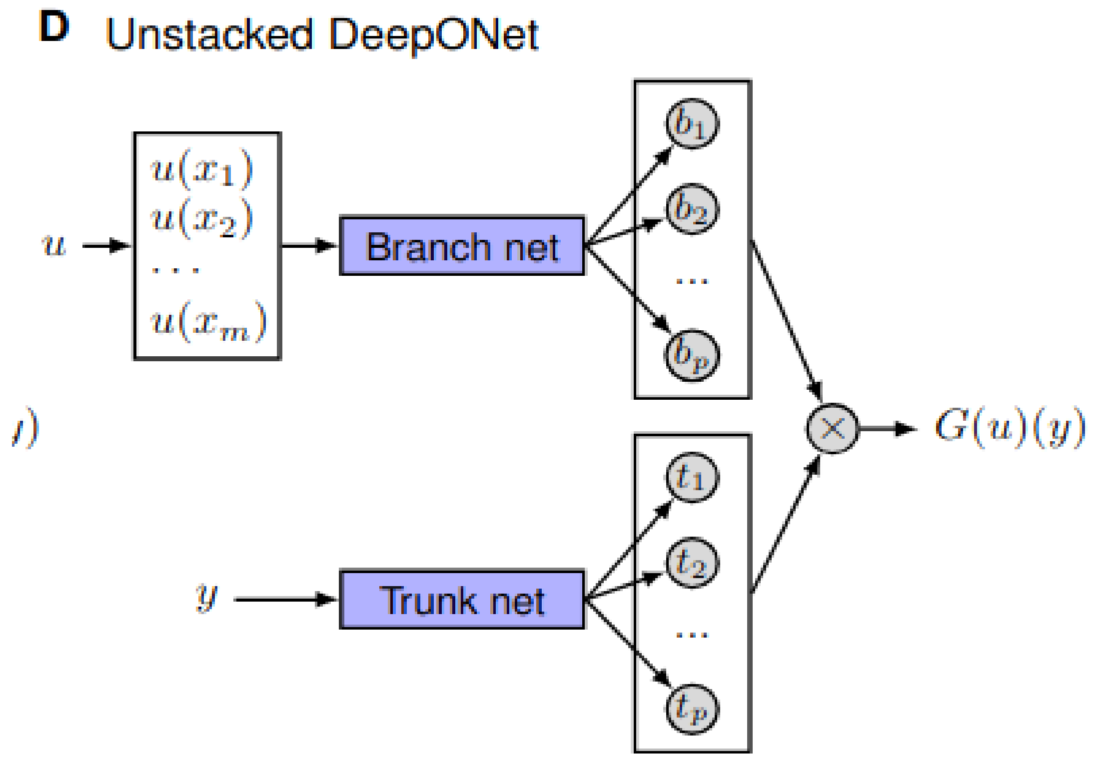
Trunk Network: Processes evaluation parameters and locations.



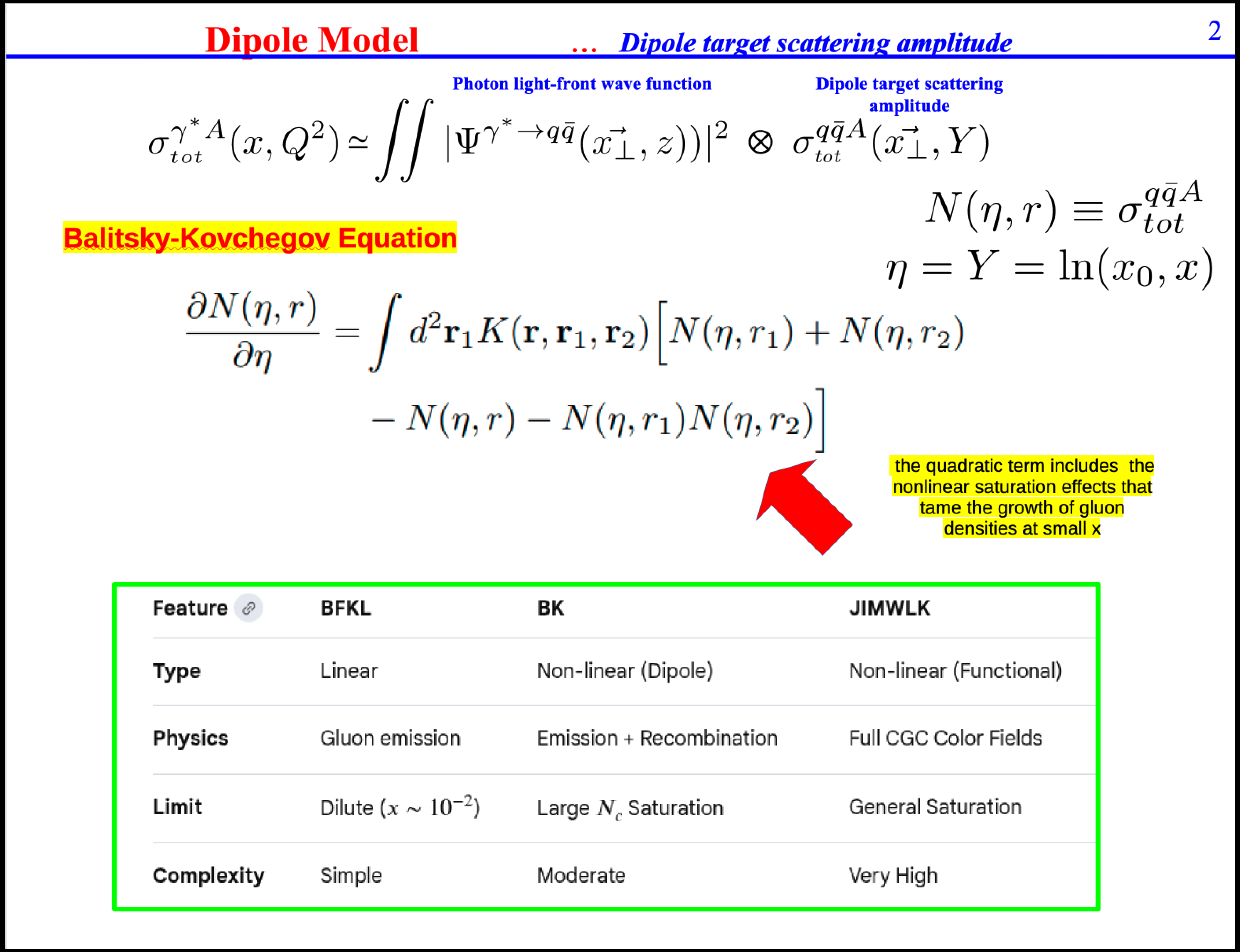
Theorem 1 (Universal Approximation Theorem for Operator). *Suppose that σ is a continuous non-polynomial function, X is a Banach Space, $K_1 \subset X$, $K_2 \subset \mathbb{R}^d$ are two compact sets in X and $\mathbb{R}^d$, respectively, V is a compact set in $C(K_1)$, G is a nonlinear continuous operator, which maps V into $C(K_2)$. Then for any $\epsilon > 0$, there are positive integers n, p, m, constants $c_i^k, \xi_{ij}^k, \theta_i^k, \zeta_k \in \mathbb{R}$, $w_k \in \mathbb{R}^d$, $x_j \in K_1$, $i = 1, \ldots, n$, $k = 1, \ldots, p$, $j = 1, \ldots, m$, such that*

$$\left| G(u)(y) - \sum_{k=1}^{p} \underbrace{\sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} \xi_{ij}^k u(x_j) + \theta_i^k \right)}_{branch} \underbrace{\sigma(w_k \cdot y + \zeta_k)}_{trunk} \right| < \epsilon \qquad (1)$$

holds for all $u \in V$ and $y \in K_2$.



Different Models – Gluon Saturation



Dipole Model … *Dipole target scattering amplitude*

Photon light-front wave function **Dipole target scattering amplitude**

$$\sigma_{tot}^{\gamma^* A}(x, Q^2) \simeq \iint |\Psi^{\gamma^* \to q\bar{q}}(\vec{x_\perp}, z))|^2 \otimes \sigma_{tot}^{q\bar{q}A}(\vec{x_\perp}, Y)$$

$$N(\eta, r) \equiv \sigma_{tot}^{q\bar{q}A}$$

$$\eta = Y = \ln(x_0, x)$$

Balitsky-Kovchegov Equation

$$\frac{\partial N(\eta, r)}{\partial \eta} = \int d^2\mathbf{r}_1 K(\mathbf{r}, \mathbf{r}_1, \mathbf{r}_2)\Big[N(\eta, r_1) + N(\eta, r_2) - N(\eta, r) - N(\eta, r_1)N(\eta, r_2)\Big]$$

the quadratic term includes the nonlinear saturation effects that tame the growth of gluon densities at small x

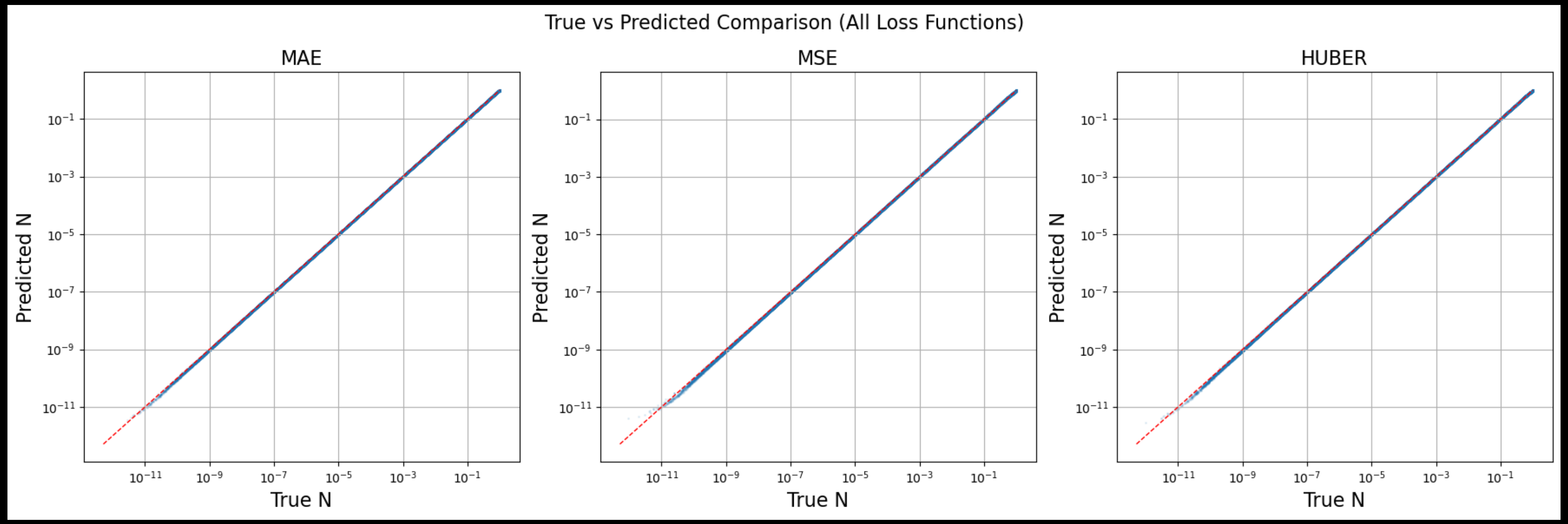
Feature	BFKL	BK	JIMWLK
Type	Linear	Non-linear (Dipole)	Non-linear (Functional)
Physics	Gluon emission	Emission + Recombination	Full CGC Color Fields
Limit	Dilute ($x \sim 10^{-2}$)	Large N_c Saturation	General Saturation
Complexity	Simple	Moderate	Very High

1. Modified MLP – Role of Loss Functions

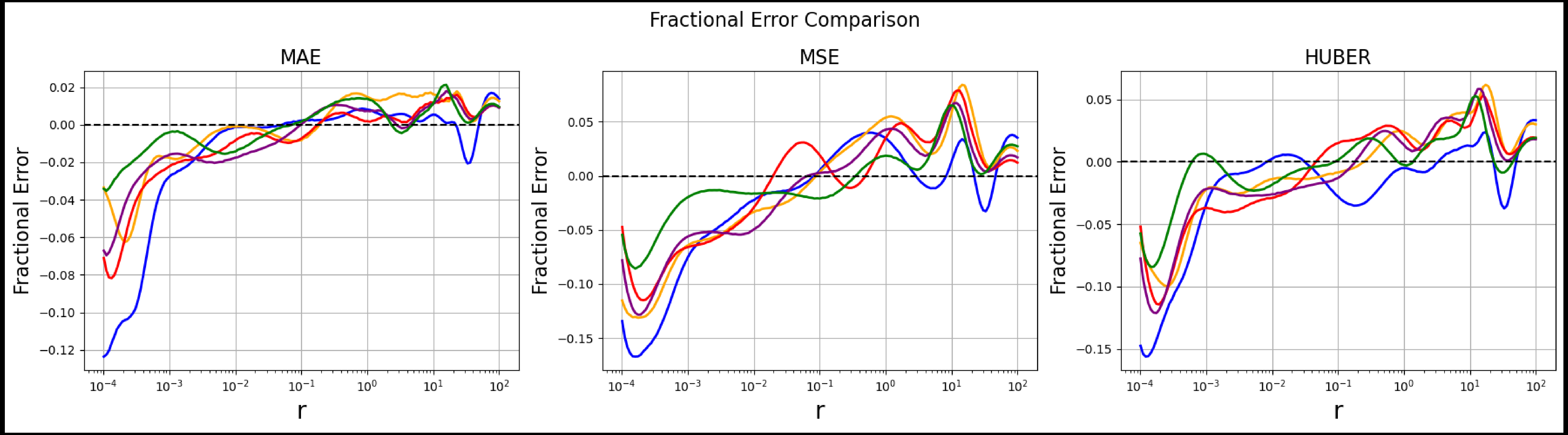
MLP Architecture

Component	Previous → New
Network Width	128 units/layer → **256 units/layer**
Dropout	0.10 → **0.05**
Learning Rate	5e-4 → **3e-4 (decrease)**
LR Decay Rate	0.90 → **0.98**
Max Epochs	Lower limit → **500 (increase)**
Batch Size	Previous setting → **32**

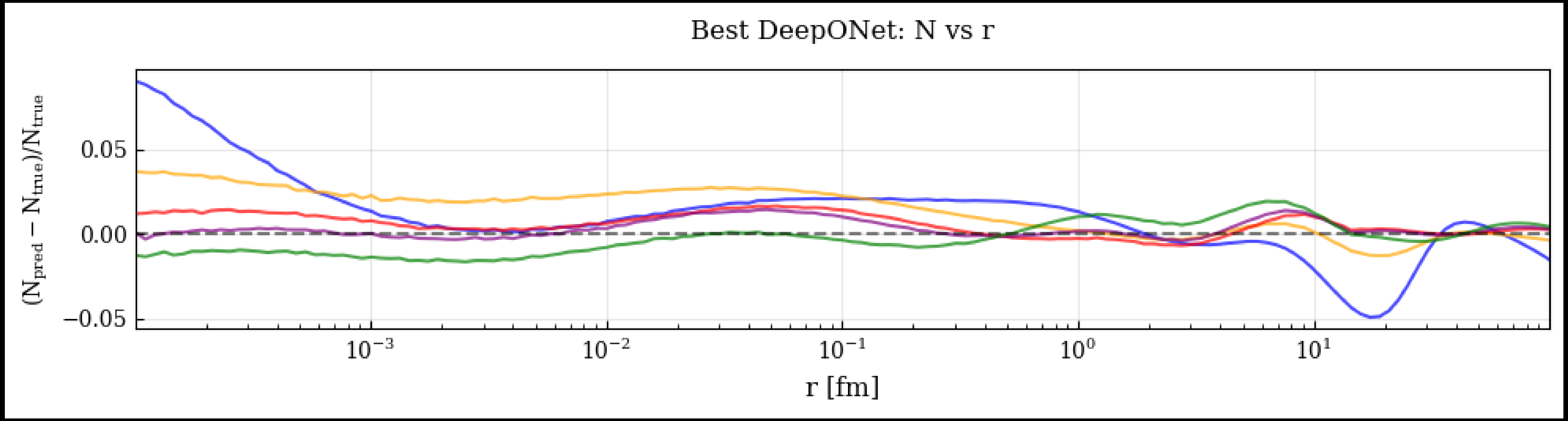
Training with Loss functions



2. Uncertainty difference:



MLP



DeepONet