Exploring the Parameter Space of Cosmic-Ray Propagation with Machine Learning

18aC310-13

JPS annual meeting
Sapporo, 2024/9/18

Holger Motz, Elias Hanser[1]
Waseda University, GCSE
Technical University of Munich[1]

Introduction

- Context: Study of cosmic-propagation based on the measured nuclei spectra

- Motivation: Suitable propagation conditions needed to use electron/positron cosmic rays for studying astrophysical sources and dark matter signatures.

- Hypothesis: Structures/differences in nuclei spectra caused by propagation

→ Goal: Find propagation conditions explaining the current nuclei spectra measurements assuming a common source spectrum for all primary nuclei

Model and Calculation Method

- Common source spectrum: power law with index γ_l below, and γ_h above the break at R_{bi} with softness s_{bi} , and with an exponential cut-off at R_{cut}

- Diffusion coefficient depending on position and with two breaks in rigidity dependence:

$$D(r,z,D)=D_0 max\left(e^{(r-r_n)/r_s},1\right) max\left(e^{(z-z_n)/z_s},1\right)\left(\frac{R}{4\,GV}\right)^{\delta_l}\left(1+\left(\frac{R}{R_{bl}}\right)^{\frac{\delta-\delta_l}{s_l}}\right)^{s_l}\left(1+\left(\frac{R}{R_{bh}}\right)^{\frac{\delta-\delta_h}{s_h}}\right)^{-s_h}$$

- Diffusive re-acceleration with Alven speed v_a and convection with speed v_c

- Calculations done with DRAGON [D. Gaggero et al., Phys.Rev.Lett. 111(2), 021102 (2013)]
- Modifications: Soft breaks in source spectrum and diffusion coefficient function, double exponential spatial dependence of the diffusion coefficient

18-Dimensional Parameter Space

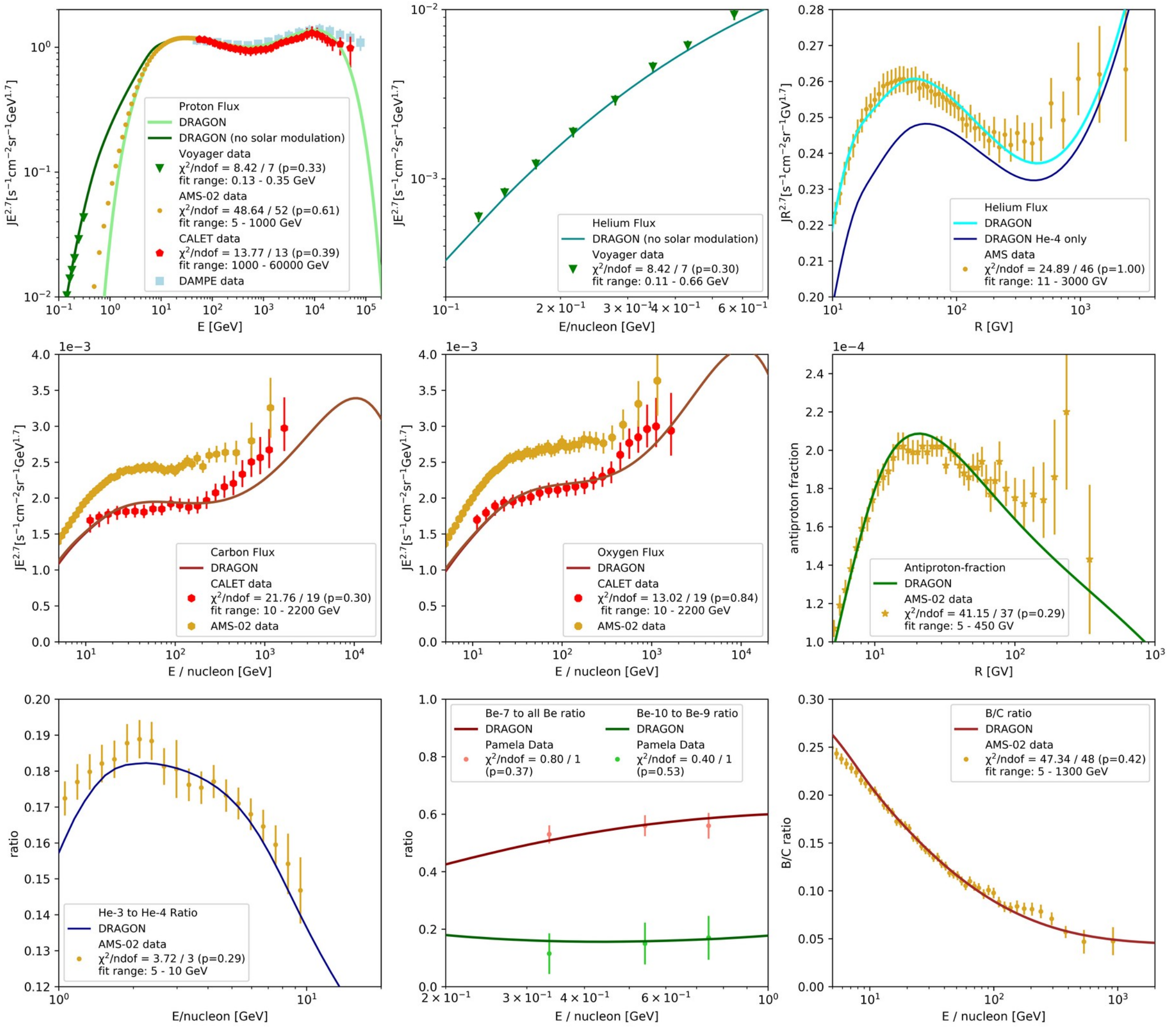
Sources	Low rigidity index	Break rigidity	Break softness	High rigididty index	Cut-off rigidity	Spiral arm width
	γ_l	R_{bi}	s_{bi}	γ_h	R_{cut}	w_{sa}
Diffusion	Diffusion coefficient normalization	Radial scale distance	Exponential scale height	Low rigidity index	Low break rigidity	Low break softness
	D_0	r_s	z_s	δ_l	R_{bl}	s_l
	Mid rigidity index	High break rigidity	High break softness	High rigidity index	Alven velocity	**Convection Velocity**
	δ	R_{bh}	s_h	δ_h	v_a	$\mathbf{v_c}$

Previous Best-fit Model

Reference: PoS ICRC2023 068

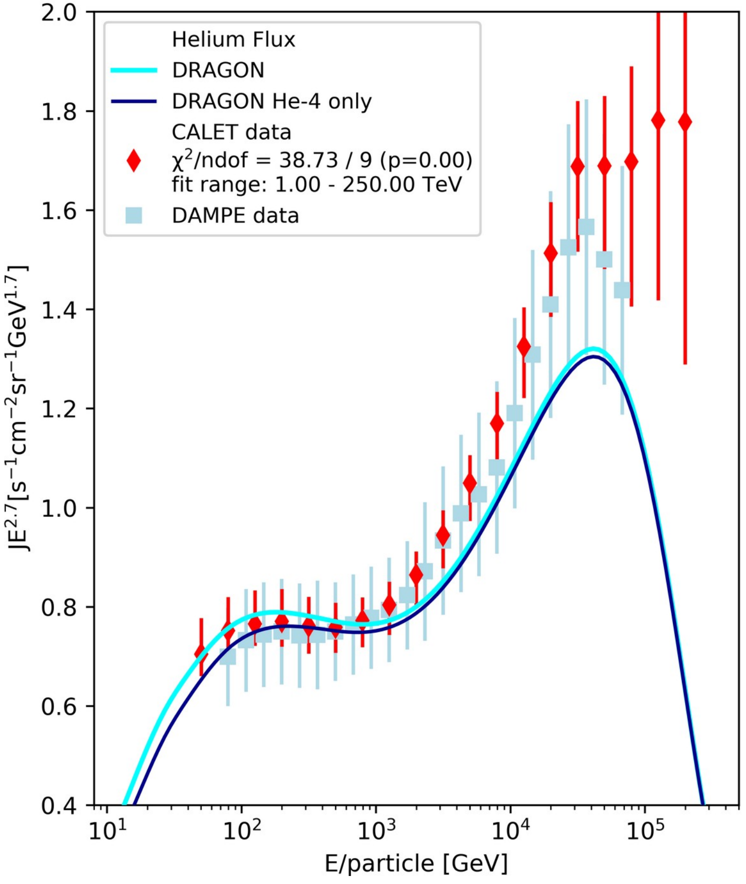
γ_l	R_{bi}	s_{bi}	γ_h	R_{cut}	w_{sa}
2.030	13.45 GV	0.229	2.345	30.12 TV	0.648 kpc
D_0	r_s	z_s	δ_l	R_{bl}	s_l
3.066 $\times 10^{28}$ cm^2/s	10.70 kpc	4.50 kpc	0.211	8.437 GV	0.053
δ	R_{bh}	s_h	δ_h	v_a	v_c
0.514	913.5 GV	0.369	0.002	14.74 m/s	0 m/s

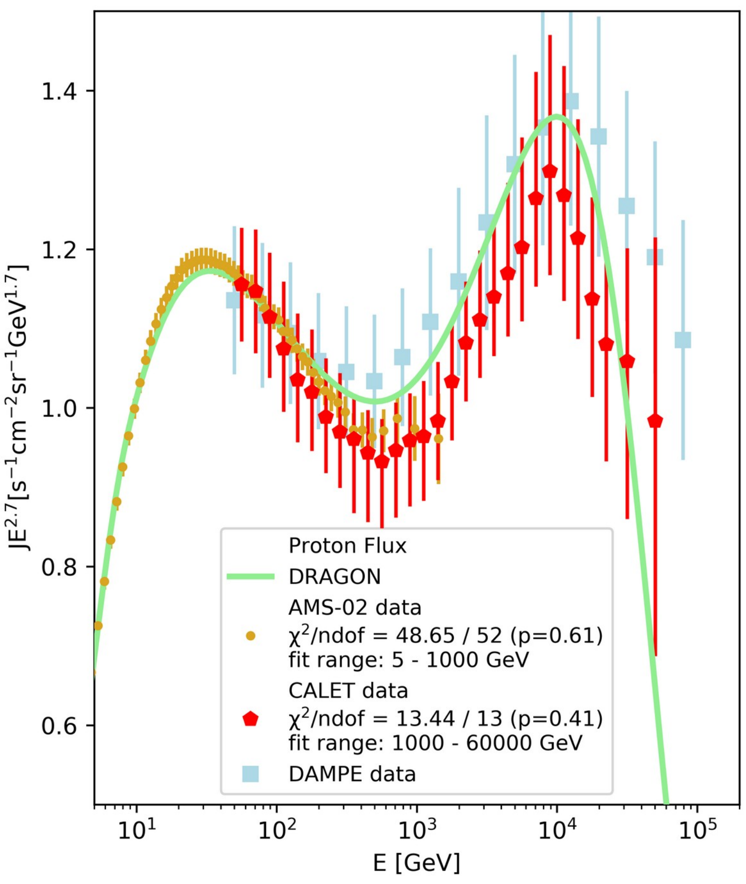
total χ^2/ndof:= 230/290



Impact of new Helium Data

CALET Helium data [Phys. Rev. Lett. 130, 171002 (2023)] shows a steeper hardening than in the proton spectrum
→ harder spectrum in conflict with TeV proton data (already upper side of errors)
→ challenge to the assumption of a common injection spectrum
→ search parameter space w. convection if there is a solution





Optimizing DRAGON Parameters

- Model quality parameters extracted from fit of normalization and solar modulation parameters to data. Examples:
  - Total χ^2 of all experimental data
    → not suitable, experiments with many data-points dominate
  - Likelihood (p-value of single exp.), neg. log. sum over experiments (Sum-NLL)
    → all experiments have equal weight
  - Negative log of p-value for single worst fitting experiment (Max-NLL)
    → best to see if model can agree to each experiment, but not smooth function

- Parameter space so far probed by “random” walk, combination of different methods used to select next model to calculate. Examples:
  - randomly within a given step size
  - by interpolating/extrapolating parameters of already calculated models

- Calculation of full Nickel-proton spallation network very time consuming
  → to test if Helium-Proton hardening discrepancy can be solved,
  calculating helium-proton (with antiproton) range is sufficient

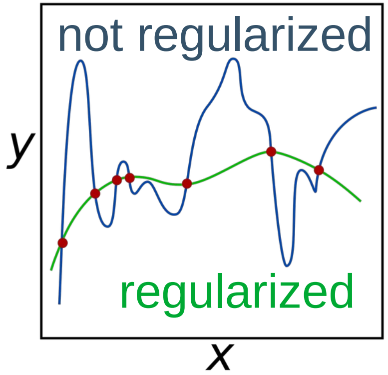
New Approach

- Try to predict the likelihood as a function of the input parameters to have a better chance of finding better parameters in each optimization step.

  → 18 dimensional parameter space → Machine Learning

- Investigated methods:
  - Neural Network (promising but difficult to set up, over-training)
  - Bayesian Optimization (for coarsely exploring parameter space)
  - **Focus of this talk: Gradient Descent on Ridge Model**

Method



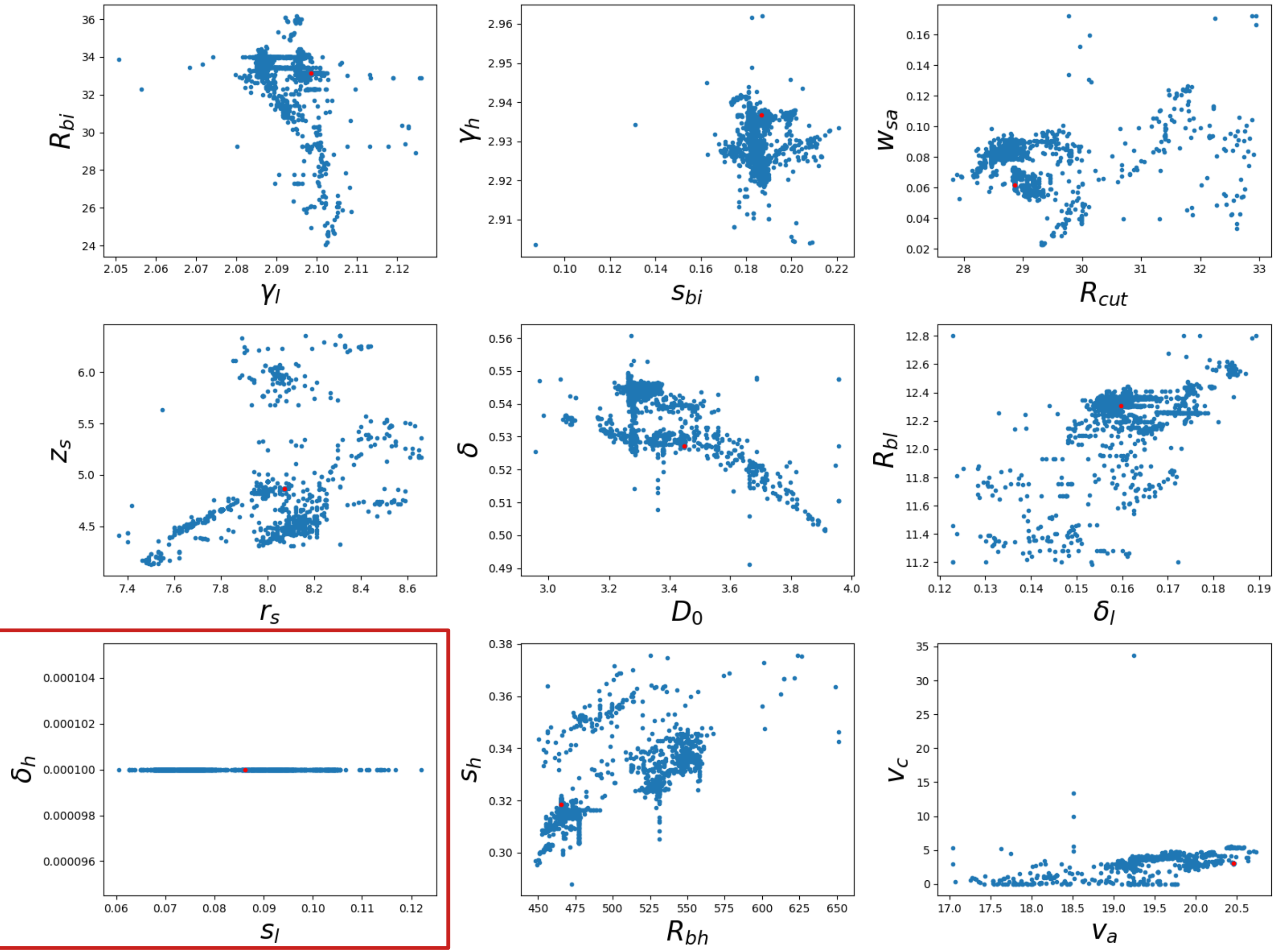
from wikipedia

- Fit a **ridge model** (regularized polynomial function) to the (preprocessed) likelihood (*Y*) results, parameters are the (preprocessed) DRAGON input parameters (*X*)
- Hyper-parameters (polynomial degree, regularization parameter) are determined by splitting the existing data in training (90%) and test (10%) samples and optimizing for best prediction of the test samples by the model

  (polynomial terms are all combinations of the parameters with combined power of the polynomial's degree: 1329 for 18 parameters and 3rd degree)
- Determine gradient of function at a start-point (best likelihood or random) and take a step in negative gradient direction
  → new set of parameters to be calculated by DRAGON
- Using tools from the scikit-learn library

Parameter Space

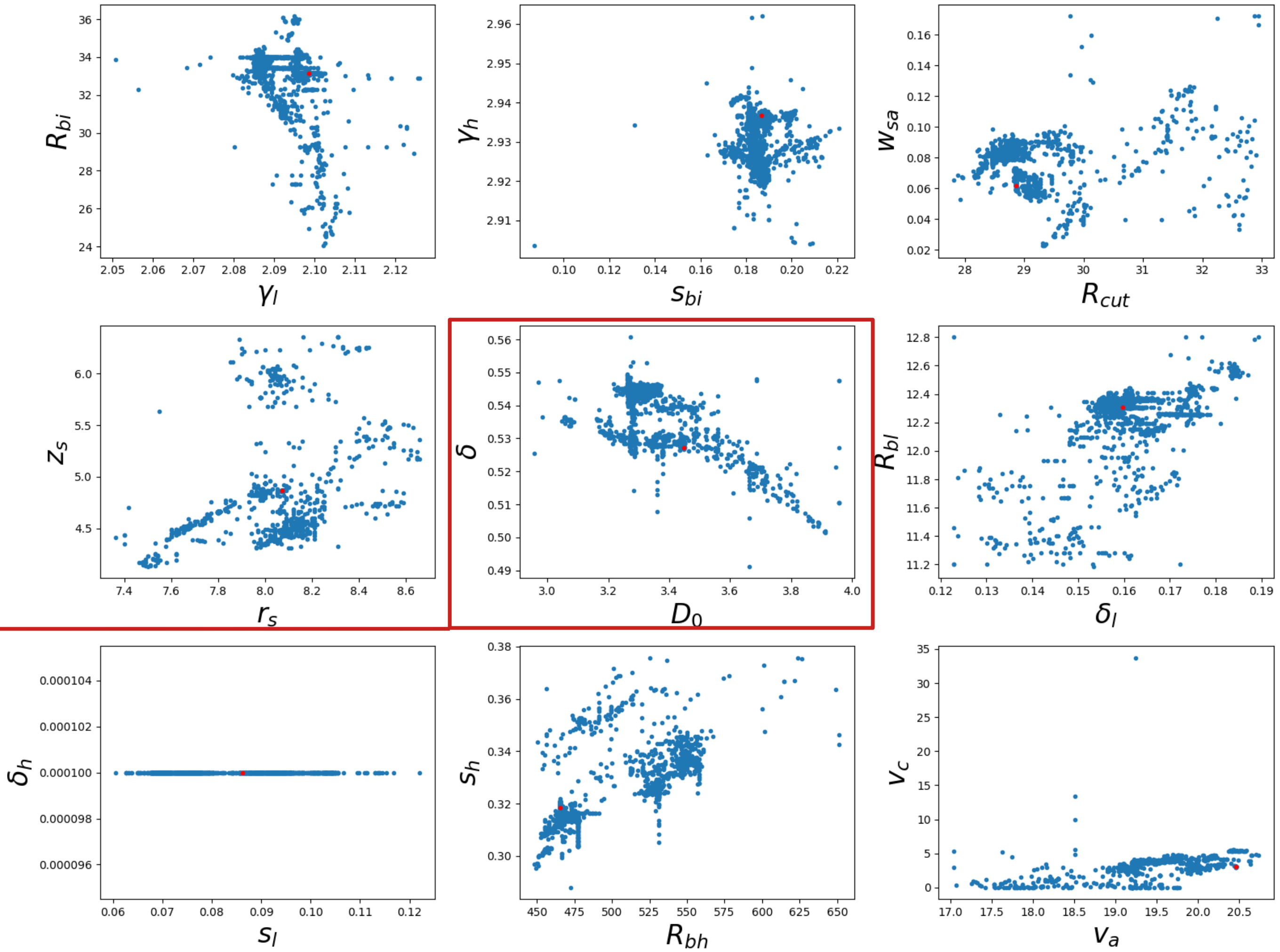
scatter plots of the already calculated points with pairs of parameters as axes



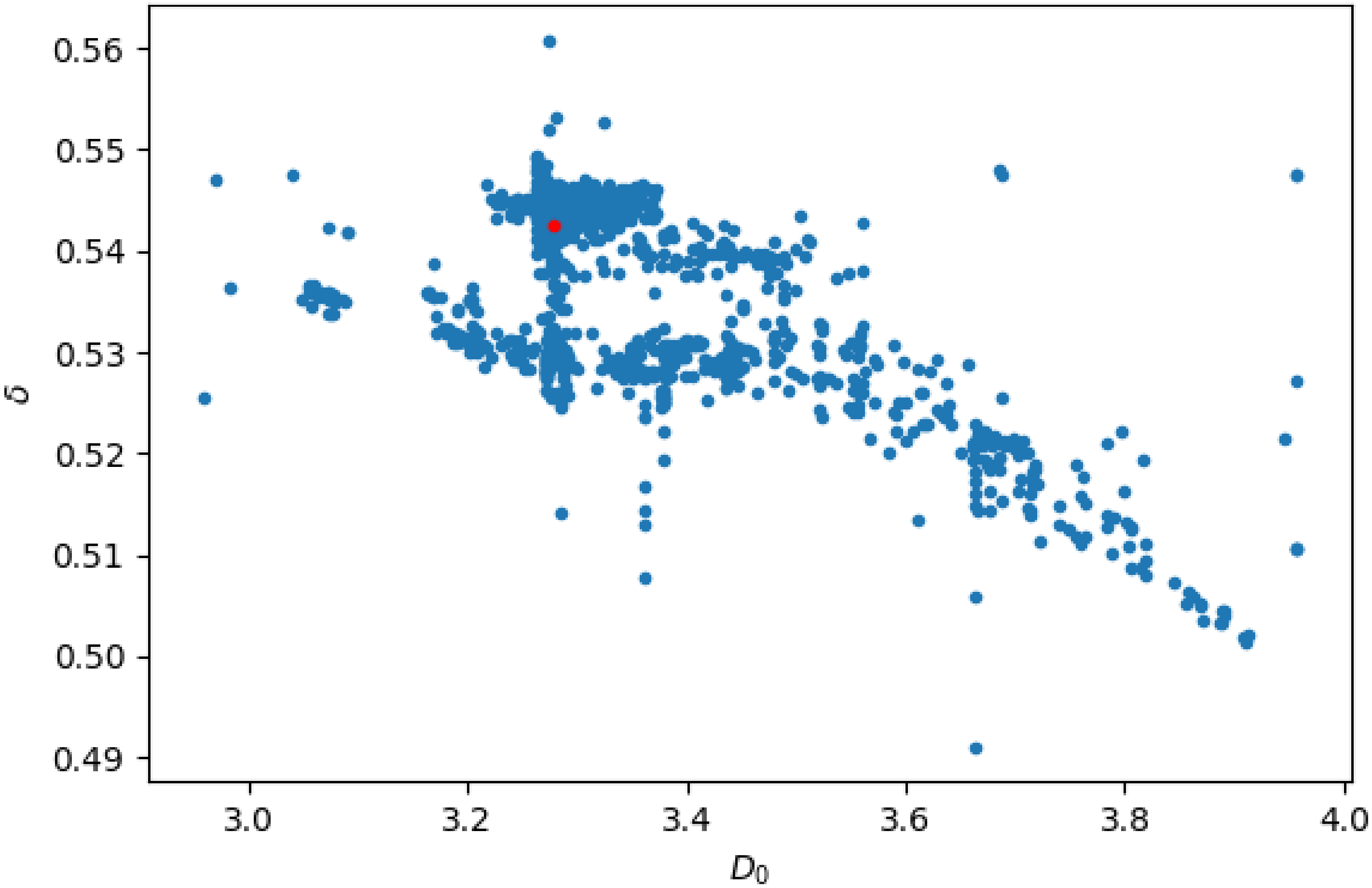
High energy diffusion coefficient power law index fixed to 0.0001 (equivalent to 0) since optimization converged to this lower boundary

Parameter Space

Take diffusion coefficient normalization vs. power law index as an example to demonstrate data pre-processing

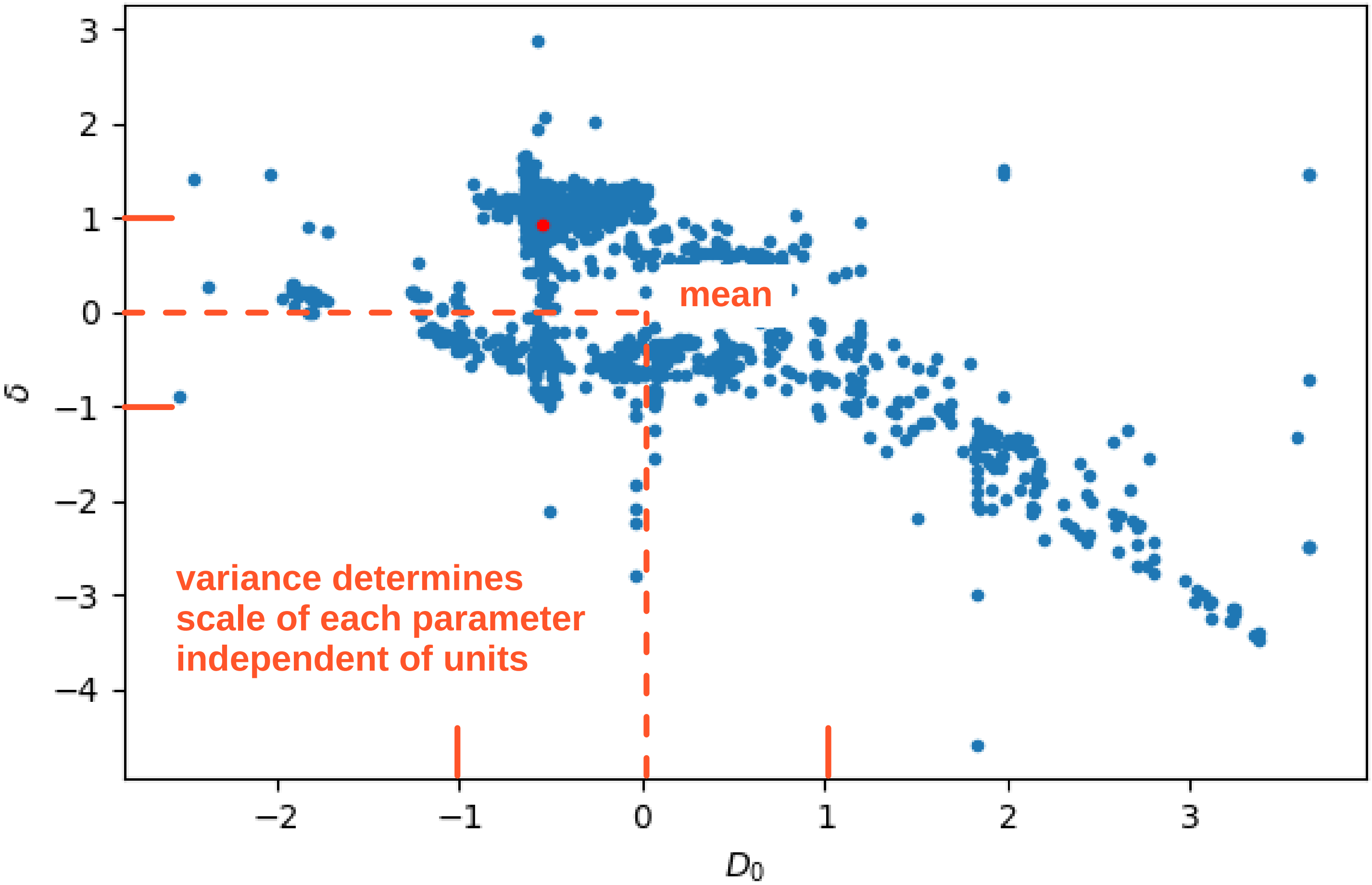


Pre-Processing



Step 1:
subtract mean and
scale to variance
(sklearn.preprocessing.
StandardScaler)

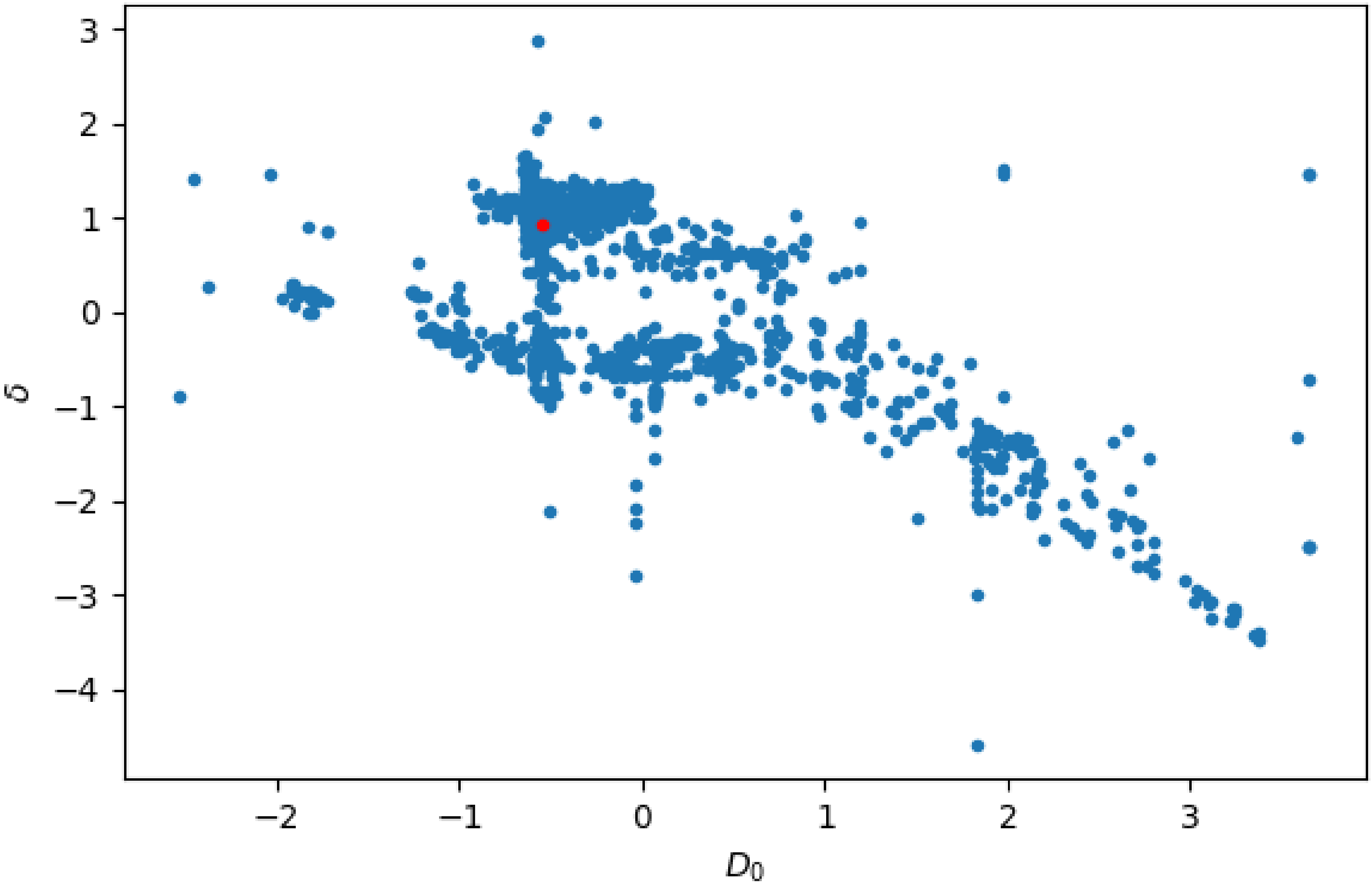
Pre-Processing



Step 1:
subtract mean and scale to variance (sklearn.preprocessing. StandardScaler)

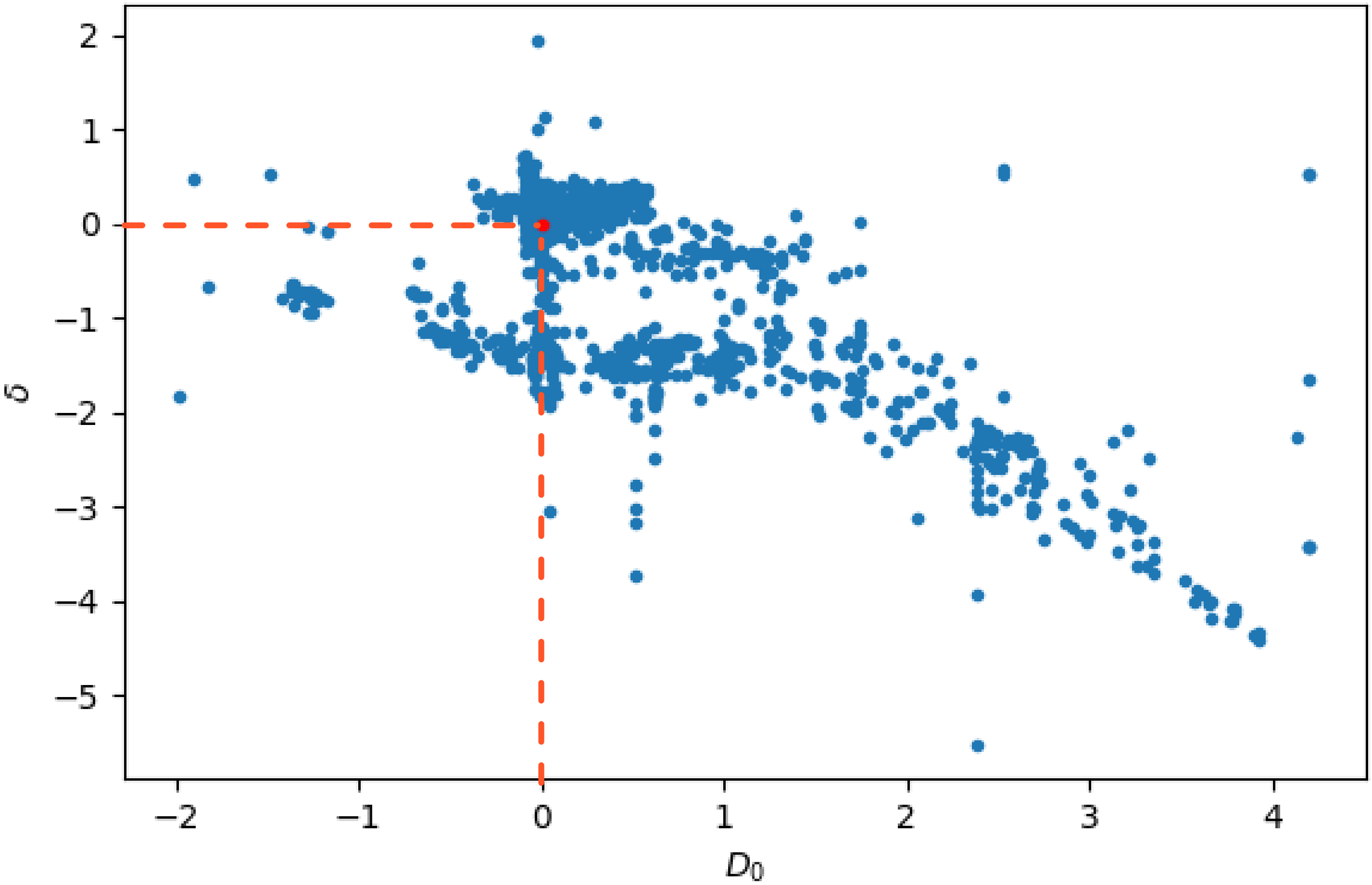
→ this allows for calculating distances between models using an Euclidean metric
→ used for weighting of points in fitting of the ridge model

Pre-Processing



Step 2:
subtract position of the start point (best point)

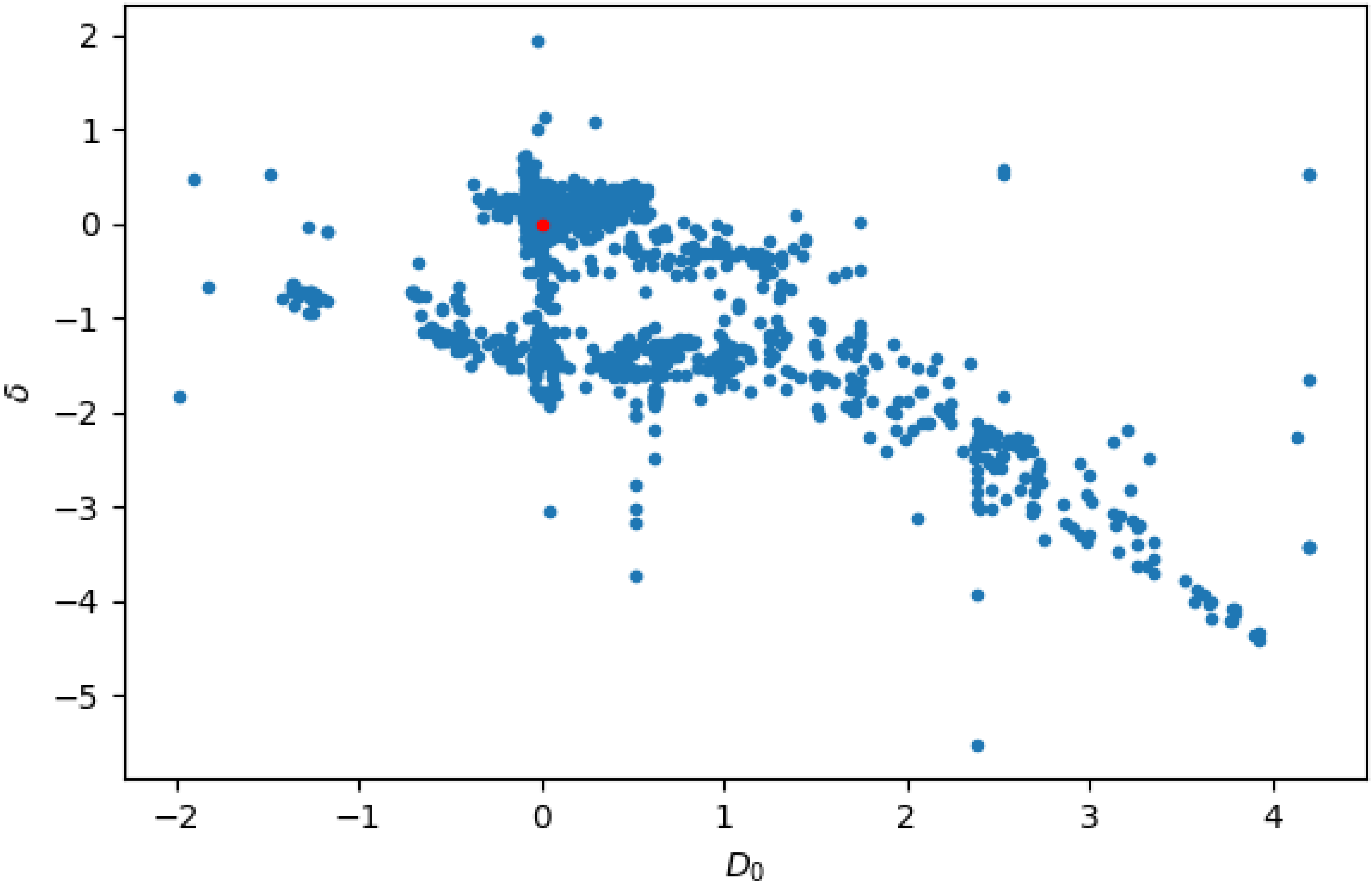
Pre-Processing



Step 2:
subtract position of the start point (best point)

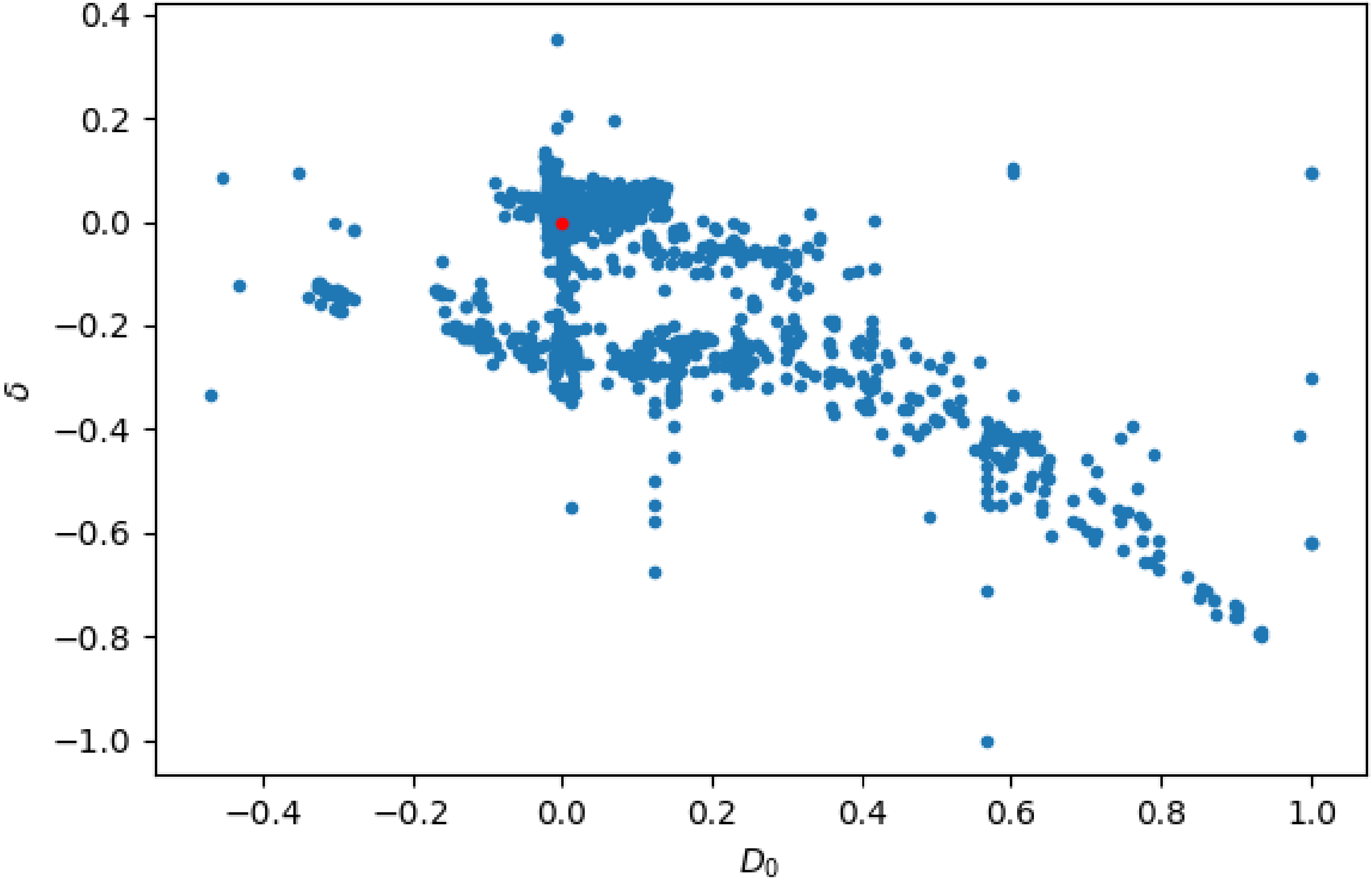
→ the intercept of the polynomial function is set to zero
→ likelihood is also scaled and shifted
→ the start point becomes the origin and the function is forced to be correct there

Pre-Processing



Step 3:
scale values to fit in the range between -1 and 1 (sklearn.preprocessing. MaxAbsScaler)

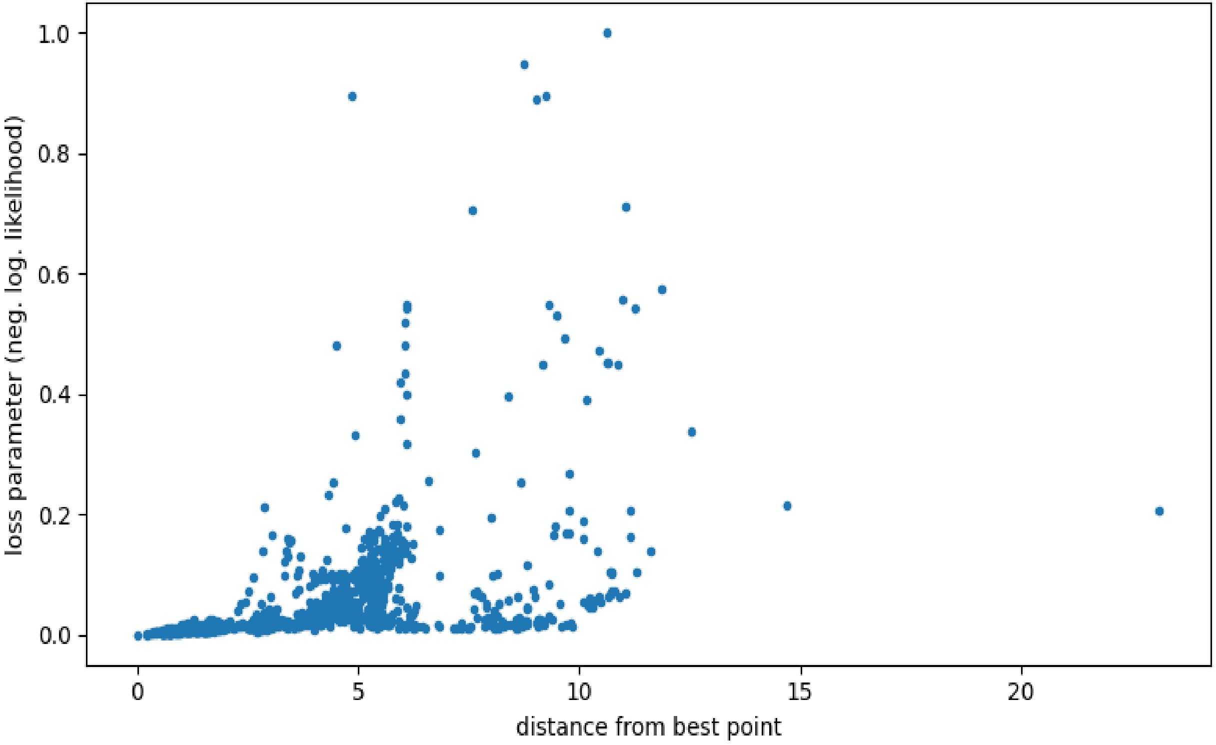
Pre-Processing



Step 3:
scale values to fit in the range between -1 and 1 (sklearn.preprocessing.
MaxAbsScaler)

→ fitting procedure requires normalized data

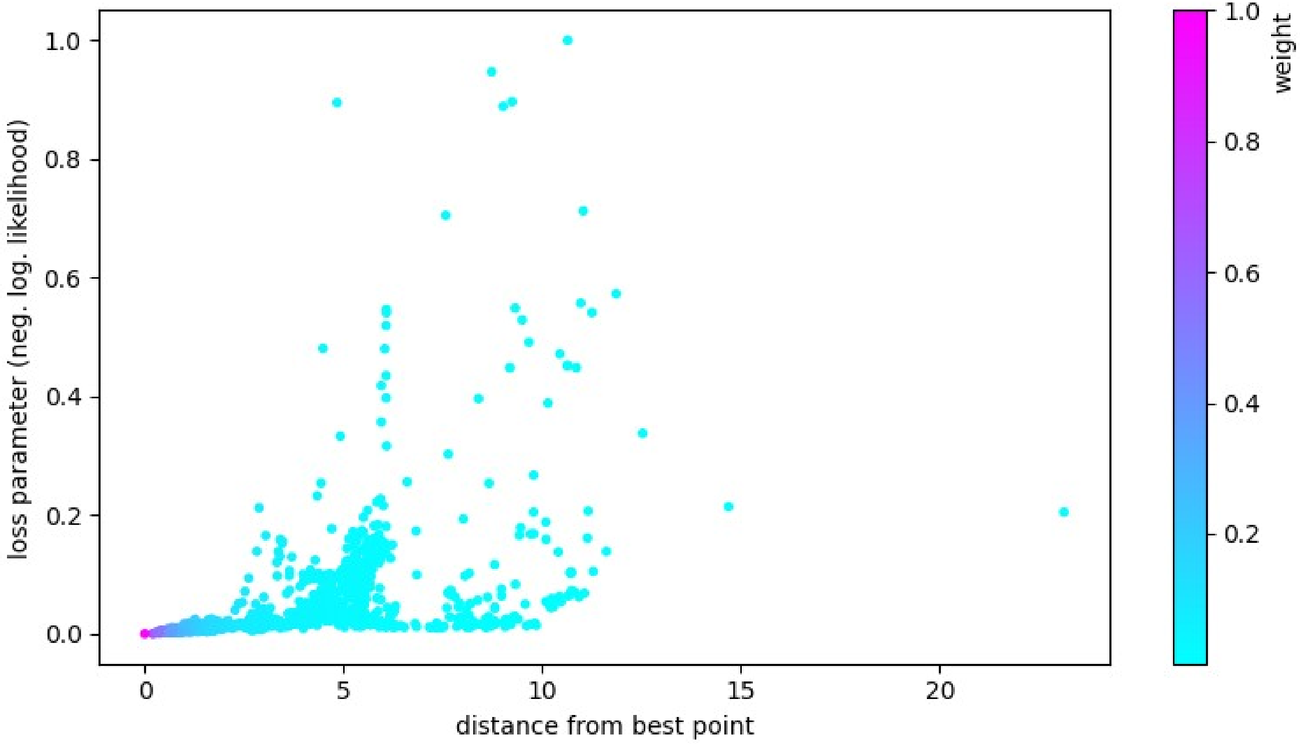
Weighting in Ridge Model Fit



No Weights:

All points are considered equally in the fitting

Weighting in Ridge Model Fit

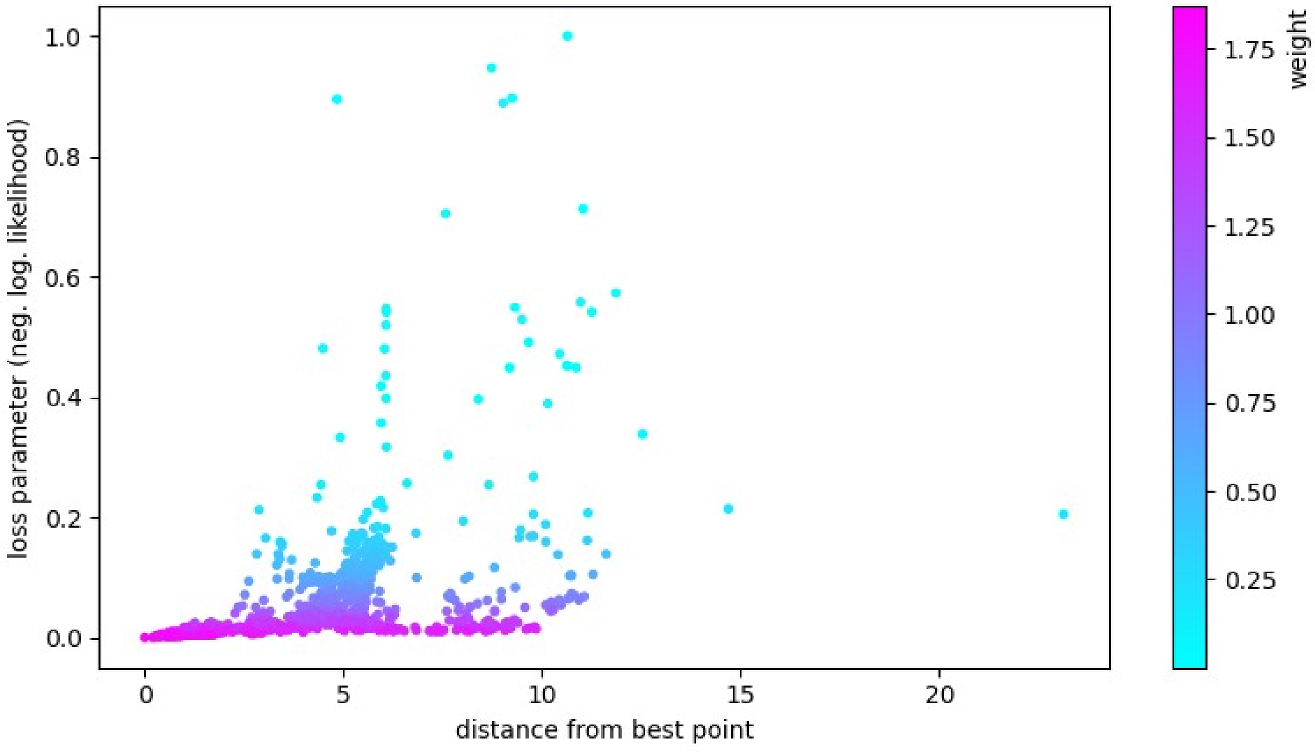


Distance Weights:

We are most interested in the function being correct near the start point → weight reduced with distance from start point

$$w = \left(1 + \frac{1}{d}\right)^2$$

Weighting in Ridge Model Fit



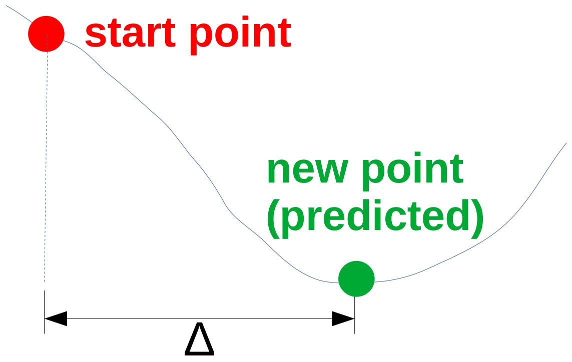
Exponential Weights:

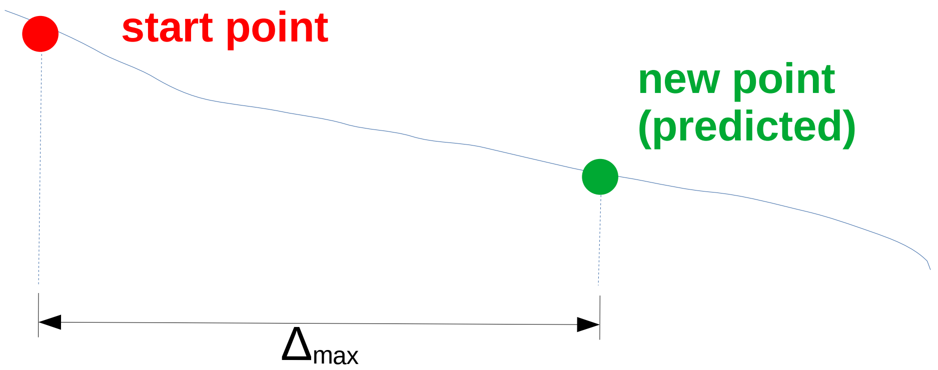
We are most interested in the function being correct for points with good likelihood (reduce influence of outliers with very high neg. log. likelihood)

$$w = e^{-NLL}$$

Gradient Steps

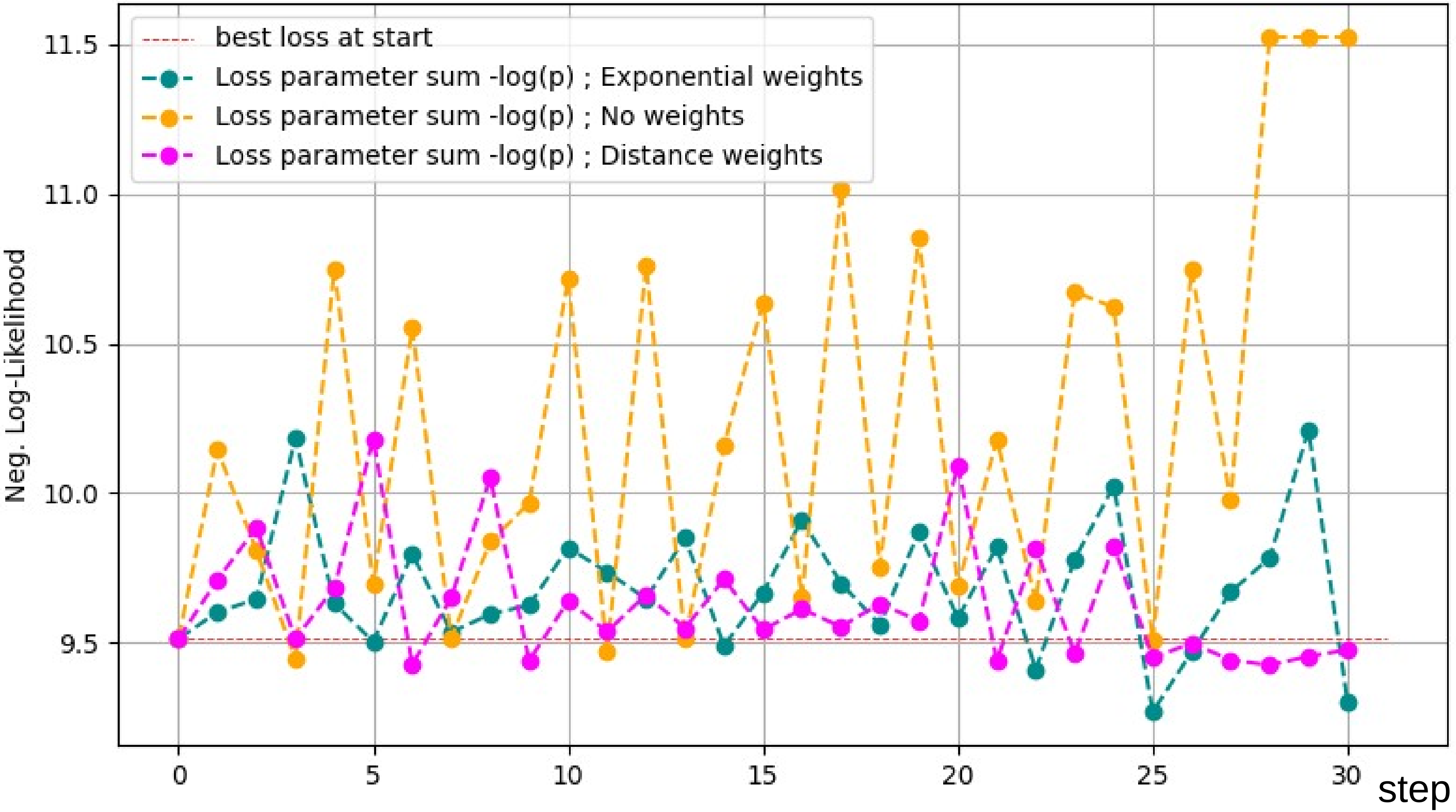
- Gradient is calculated numerically by finite steps in each parameter direction
- Distance Δ to go in direction of negative gradient is calculated by evaluating the polynomial function along the gradient axis and finding the minium.
- A pre-set maximum distance Δ_{max} is taken if no minimum found.



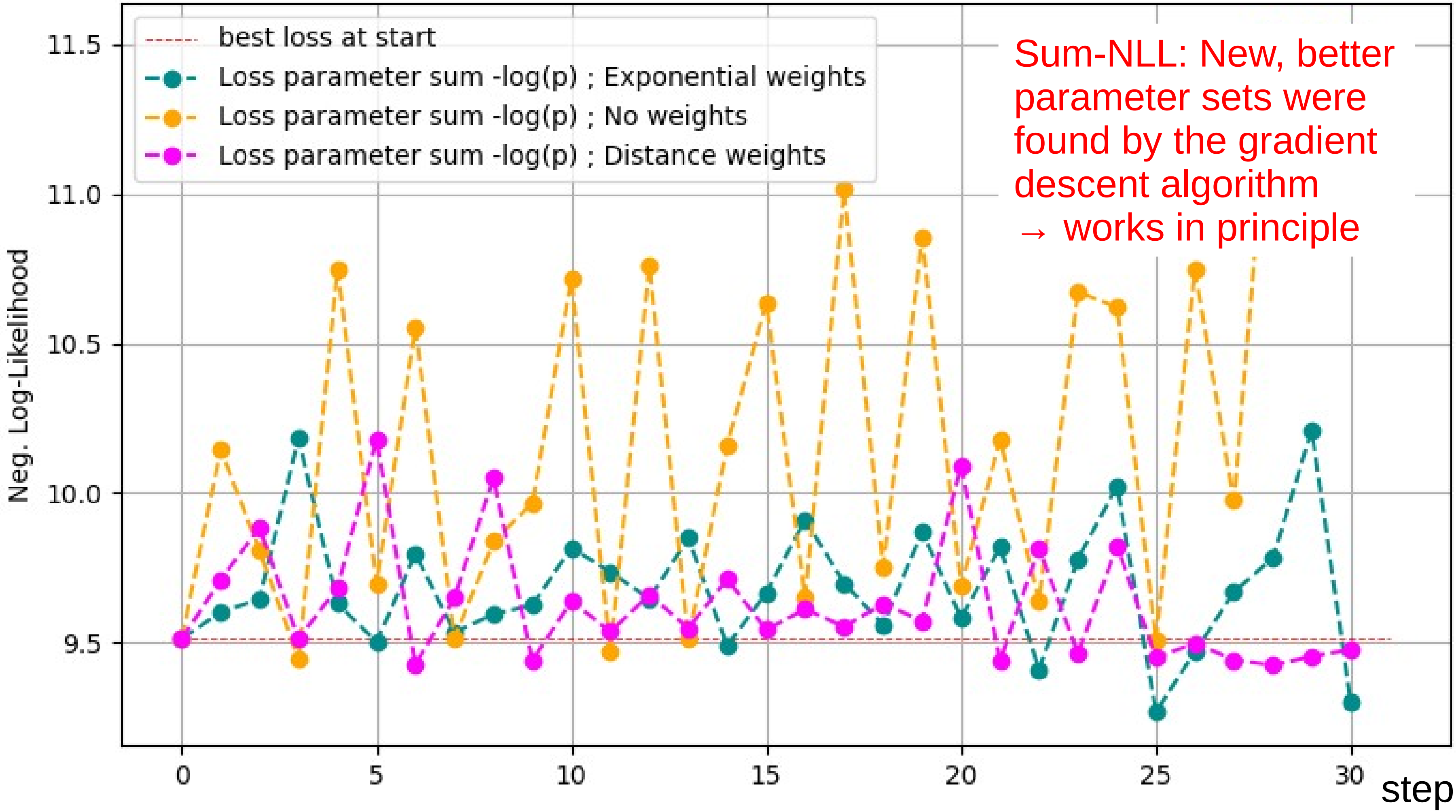


- After performing the DRAGON calculation at the new point and obtaining a likelihood value, it is added to the point sample and it becomes the start-point for the next step

Preliminary Results (Sum-NLL)

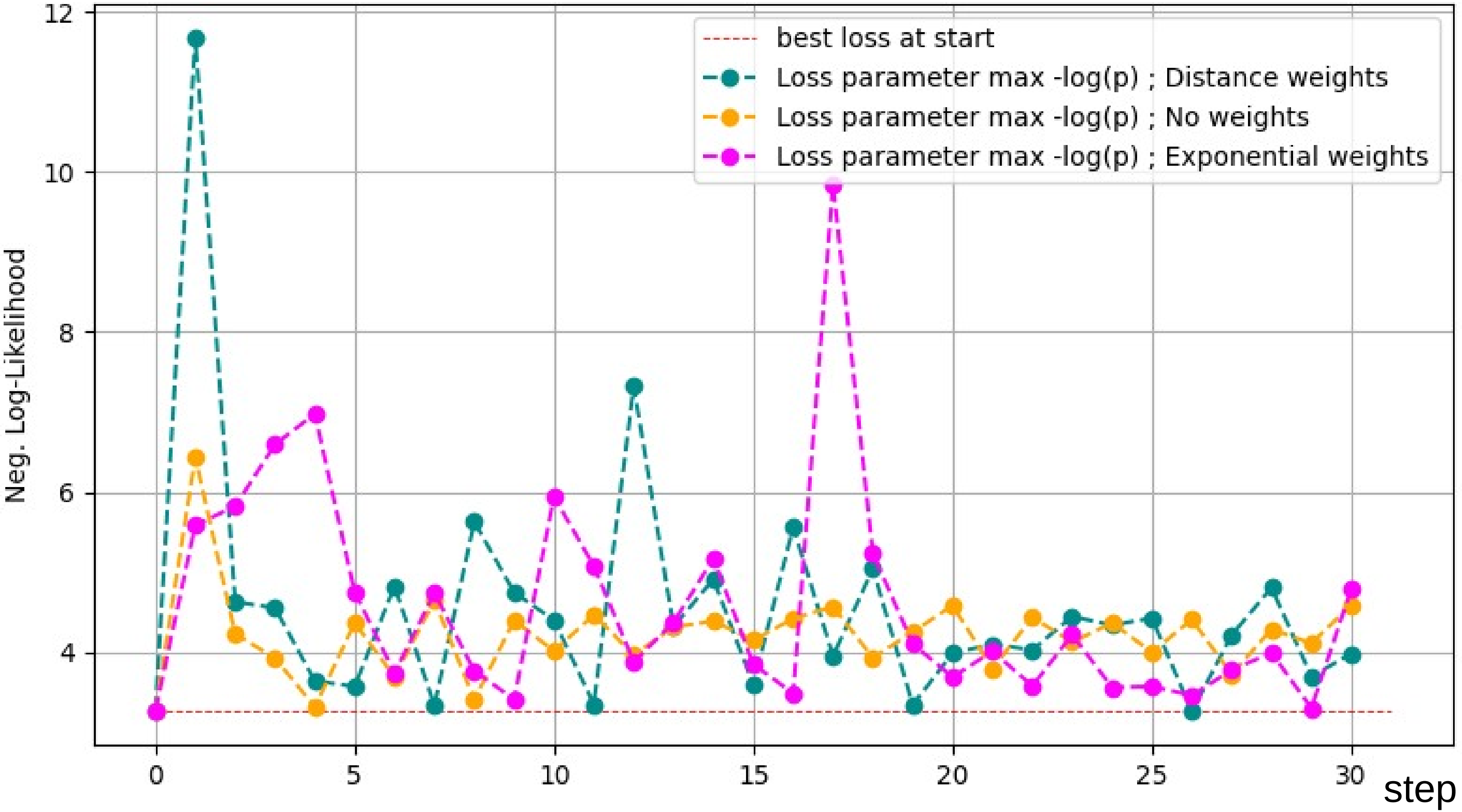


Preliminary Results (Sum-NLL)

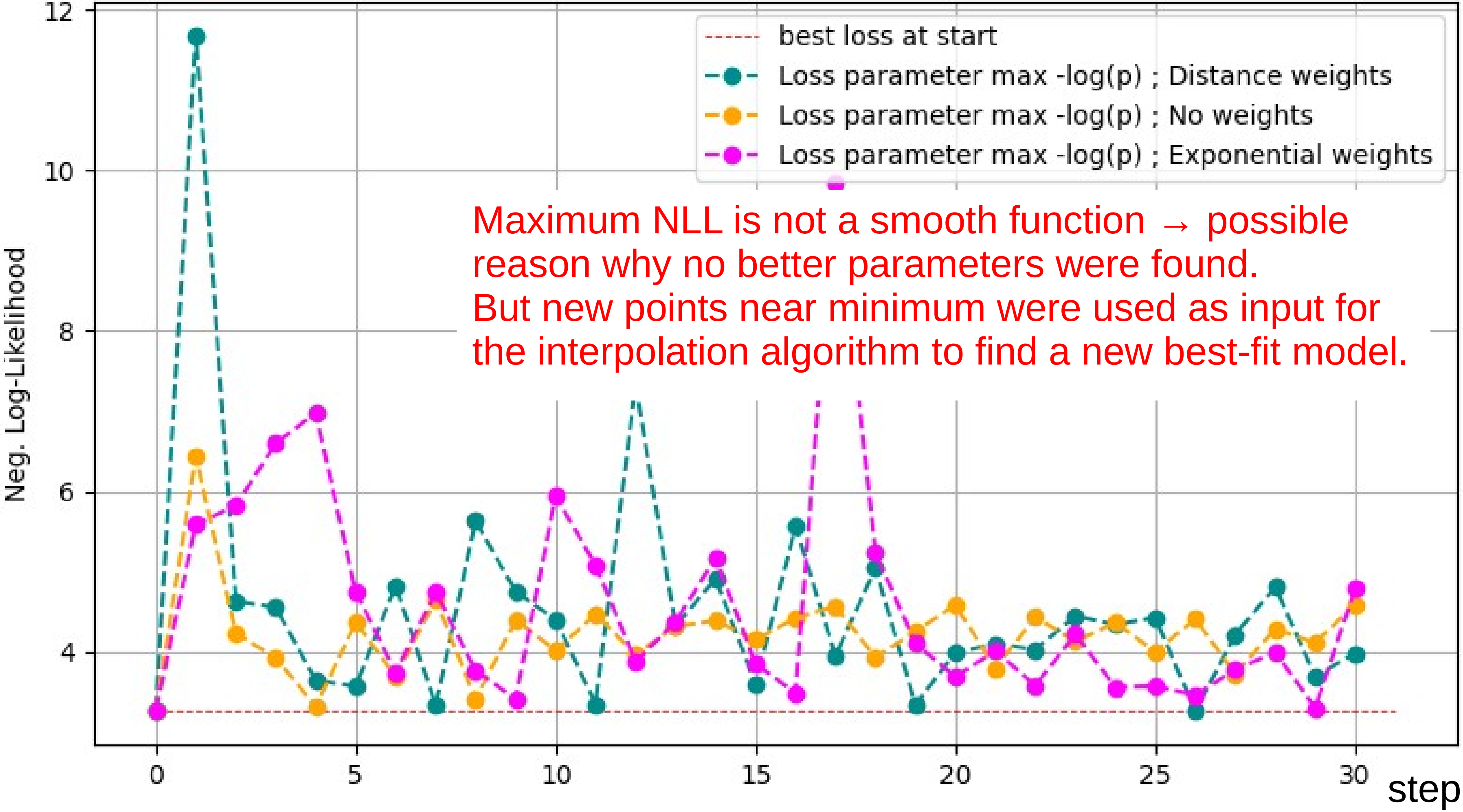


Sum-NLL: New, better parameter sets were found by the gradient descent algorithm
→ works in principle

Preliminary Results (Max-NLL)

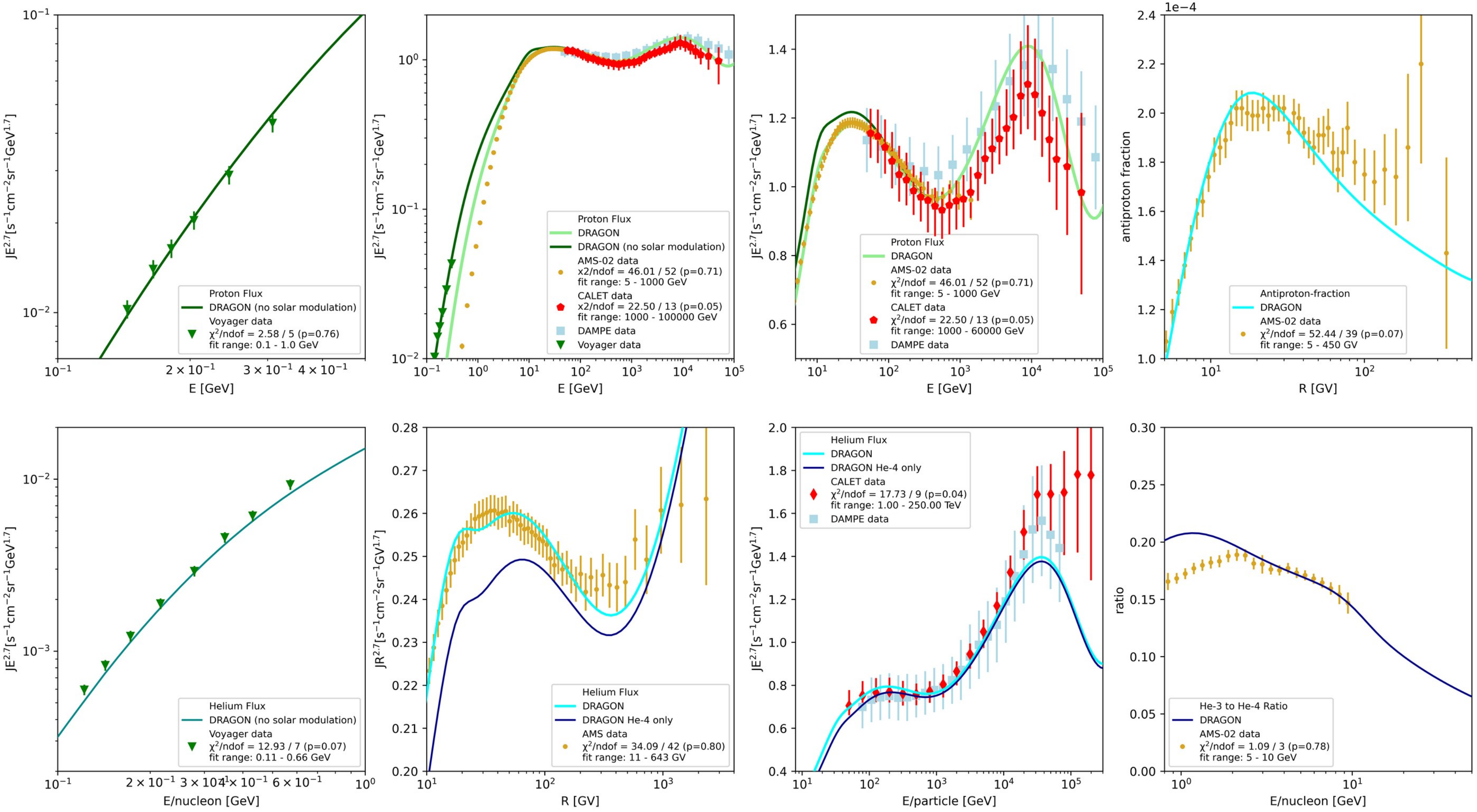


Preliminary Results (Max-NLL)



Maximum NLL is not a smooth function → possible reason why no better parameters were found.
But new points near minimum were used as input for the interpolation algorithm to find a new best-fit model.

Current Best Fit Model



minimum p-value is 0.04 → 96 %CL exclusion → progress, but need further improvement

Conclusions / Outlook

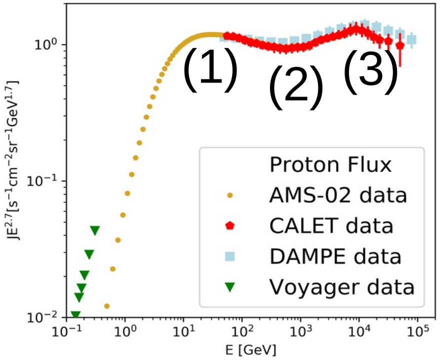
- The ridge model gradient descent algorithm is a first implementation of a machine learning algorithm to understand the parameter space of cosmic ray propagation and support the numerical calculation with DRAGON.
- The stronger hardening of the Helium spectrum compared to proton is difficult to explain with a common injection spectrum, but the found best-fit model suggests an explanation based on propagation effects may be possible.
- Further evaluation and tuning of the algorithm ongoing.
- Next goal: Prediction of likelihood by neural network aiming for better performance also with not smooth loss functions (i.e. Max-NLL).

This work is supported by JSPS KAKENHI Grant No. JP24K00665 and Waseda University Grant for Special Research Project 2024Q-016

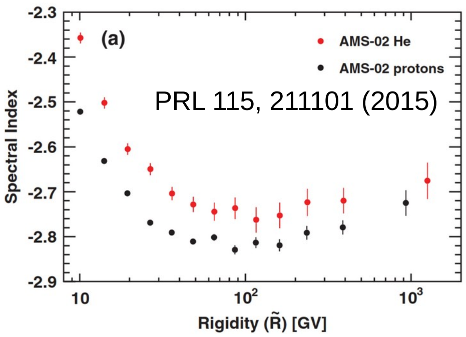
Backup Slides

Hypothesis: Differences in Nuclei Spectra Caused by Propagation

- Observed spectra are power laws but the index changes with rigidity at several points: (1) Softening @ O 10 GV, (2) Hardening @ O 100 GV – 1 TV, (3) Softening @ O 10 TV



- Indices and break positions different between proton and He (and other nuclei, but less significant)



- Possible Explanations:
  - Source spectrum different for each nuclei species
  - **Propagation causes differences in spectral shape**

→ Assumed common source spectrum: power law with index γ_l below, and γ_h above the break at R_{bi} with softness s_{bi} , and with an exponential cut-off at R_{cut}

Diffusion Coefficient Structure

- Further spectral changes of the nuclei spectra are modeled by two breaks in the rigidity dependence of the diffusion coefficient, softening from δ_l to δ at R_{bl} with softness s_l , then hardening again to δ_h at R_{bh} with softness s_h
- Diffusion coefficient depends on position – exponential increase with galactic radius r, distance from galactic plane z – constant central zones: galactic center r_n = 2 kpc, galactic disc z_n = 0.15 kpc

$$D(r,z,D)=D_0 max\left(e^{(r-r_n)/r_s},1\right) max\left(e^{(z-z_n)/z_s},1\right)\left(\frac{R}{4\,GV}\right)^{\delta_l}\left(1+\left(\frac{R}{R_{bl}}\right)^{\frac{\delta-\delta_l}{s_l}}\right)^{s_l}\left(1+\left(\frac{R}{R_{bh}}\right)^{\frac{\delta-\delta_h}{s_h}}\right)^{-s_h}$$

- Motivation: Sources concentrated in galactic center and disk cause magnetic field turbulence, influence decreasing with distance – different propagation conditions for nuclei species depending on how far they propagate out into the halo and back based on nuclei mass and A/Z

Experimental Data Used (Spectra)

- **Proton Flux**

0.13 – 0.35 GeV: Voyager APJ 831(1), 18 (2016)

5 GeV – 1 TeV: AMS-02 PRL 114, 171103 (2015)

1 – 60 TeV : CALET PRL 129, 101102 (2022)

- **Helium Flux**

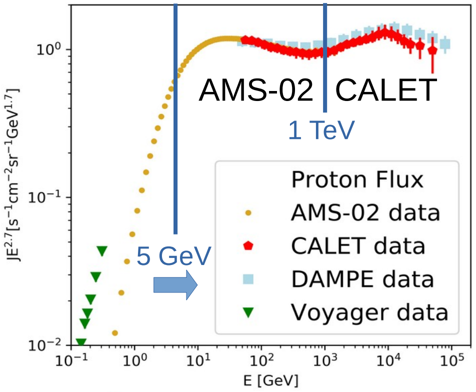
0.11 – 0.66 GeV: Voyager APJ 831(1), 18 (2016)

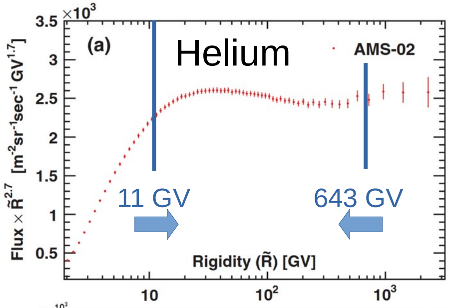
11 GV – 643 GV: AMS-02 PRL 115, 211101 (2015)

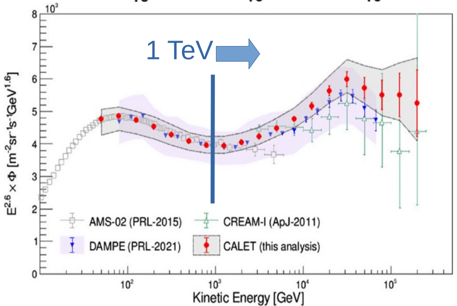
1 TeV – 250 TeV: CALET PRL 130, 171002 (2023)

- **Carbon & Oxygen Flux**

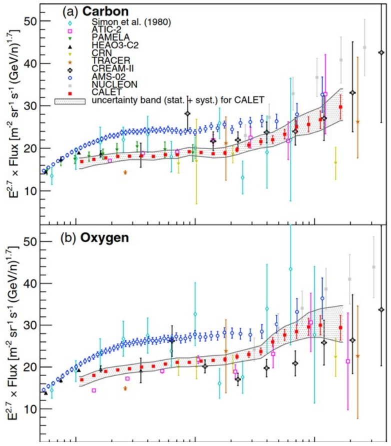
10 GeV – 2.2 TeV: CALET PRL 125, 251102 (2020)







Using data above 5 GeV/nucleon or equivalent rigidity – solar modulation difficult to model below this energy → Voyager data for lower energy range



Experimental Data Used (Ratios)

- **Antiproton fraction**

  5 – 450 GV: AMS-02 PRL 117, 091103 (2016)

- **^{3}He/^{4}He ratio**

  5 – 10 GeV: AMS-02 PRL 123, 181102 (2019)

- **B/C ratio**

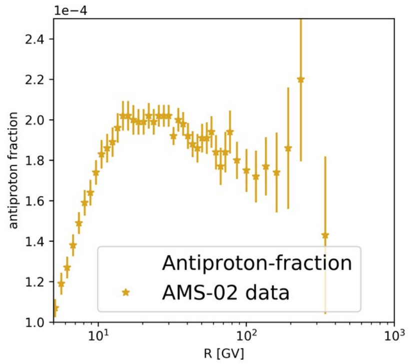
  5 GeV – 1.3 TeV: AMS-02 PRL 117, 231102 (2016)

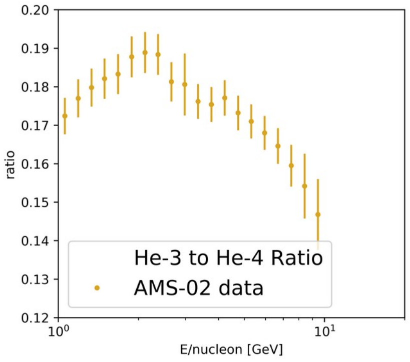
- **^{7}Be/Be ratio**

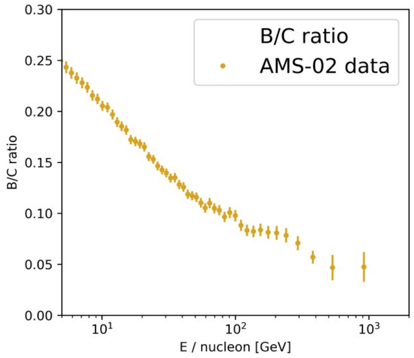
  0.25 – 0.85 GeV: PAMELA Universe 7 (2021) 6, 183

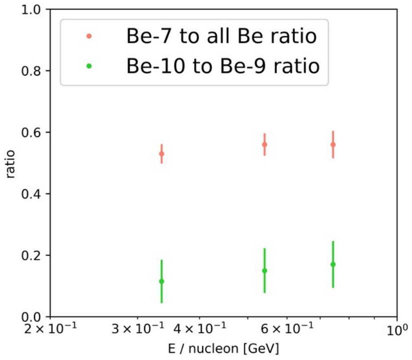
- **^{10}Be/^{9}Be ratio**

  0.25 – 0.85 GeV: PAMELA Universe 7 (2021) 6, 183





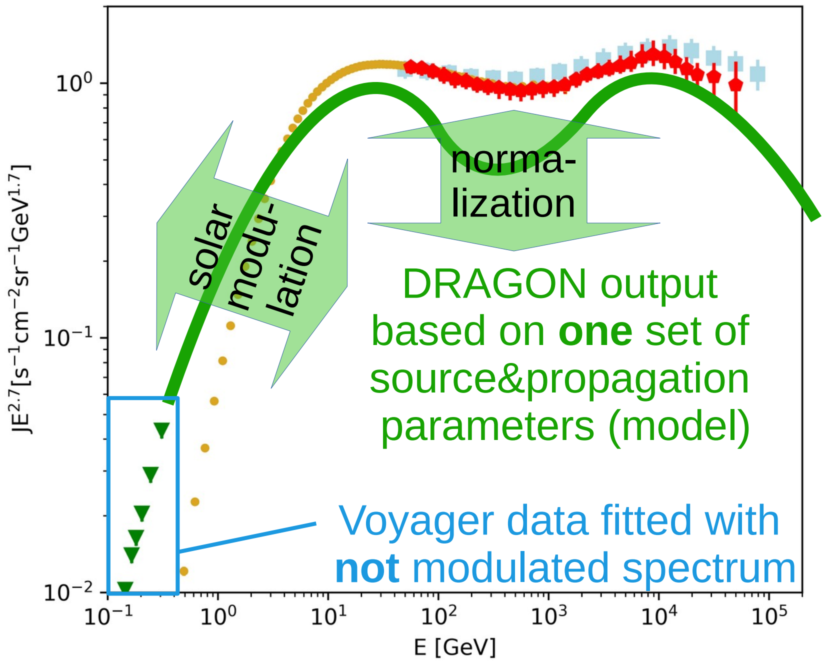




Universe 7 (2021) 6, 183
Beryllium Radioactive Isotopes as a Probe to Measure the Residence Time of Cosmic Rays in the Galaxy and Halo Thickness: A “Data-Driven” Approach, Francesco Nozzoli, Cinzia Cernetti

Fitting the Spectra to the Data

- Parameters fitted by minimizing total χ^2 of all experimental data
  - Normalization correction factors
    - Proton
    - Helium
    - Carbon
    - Oxygen
  - Solar modulation potential parameters
    - Φ_0
    - Φ_{1+} (positive charge)
    - Φ_{1-} (negative charge)
    - R_0



Charge sign and rigidity dependent solar modulation potential:

$$\Phi = \Phi_0 + \Phi_{1\pm}\left(\frac{1+(R/R_0)^2}{((R/R_0)^3)}\right)$$

based on
Ilias Cholis, Dan Hooper, Tim Linden
Phys. Rev. D 93, 043016 (2016)
“A Predictive Analytic Model for the Solar Modulation of Cosmic Rays”