NeuralHMC: Accelerated Hamiltonian Monte Carlo with a Neural Network Surrogate Likelihood

Linnea M. Wolniewicz
Information and Computer Science Department
University of Hawai'i at Mānoa
1680 East-West Road, Honolulu HI, USA
linneamw@hawaii.edu

Peter Sadowski
Information and Computer Science Department
University of Hawai'i at Mānoa
1680 East-West Road, Honolulu HI, USA
peter.sadowski@hawaii.edu

Claudio Corti
Physics and Astronomy Department
University of Hawai'i at Mānoa
2505 Correa Road, Honolulu HI, USA;
NASA Goddard Space Flight Center
8800 Greenbelt Road, Greenbelt MD, USA
corti@hawaii.edu

Abstract

Bayesian Inference with Markov Chain Monte Carlo requires the ability to efficiently compute the likelihood function. In some scientific applications, the likelihood can only be computed by a numerical PDE solver, which can be prohibitively expensive. We demonstrate that some such problems can be made tractable by amortizing the computation with a surrogate likelihood function implemented by a neural network. This can have the added benefits of reducing noise in the likelihood evaluations and providing fast gradient calculations. We demonstrate these advantages in a model of heliospheric transport of galactic cosmic rays, where our approach enables us to estimate the posterior of five latent parameters of the Parker equation.

1 Introduction

Markov Chain Monte Carlo (MCMC) methods are widely used to perform Bayesian inference in the sciences, but they are computationally expensive. Hamiltonian Monte Carlo (HMC) is an approach to accelerate MCMC sampling by using gradients of the likelihood function [Duane et al., 1987, Neal, 1993] that requires the likelihood function to be both tractable and differentiable. HMC provides faster convergence to the target distribution than traditional MCMC methods such as Random-Walk Metropolis-Hastings (RWMH), and consecutive samples have much lower autocorrelation than samples drawn using RWMH. In this work, we address a scenario in which the likelihood function is calculated by a numerical solver for a partial differential equation (PDE) and is incompatible with HMC for three reasons: it is non-differentiable, it is too slow, and it suffers from numerical instabilities. Our solution is to train a neural network (NN) surrogate likelihood that solves all three of these problems.

Previous work has used machine learning models to accelerate HMC sampling [Foreman et al., 2021a,b, Levy et al., 2018, Dhulipala et al., 2022, Li et al., 2019, Zhang et al., 2017]. Surrogate models of the log-likelihood function using Gaussian Processes were proposed by Rasmussen [2003], and Zhang et al. [2017] used shallow NNs as surrogate models in order to scale better to large datasets. Our setup differs in that the original likelihood function can only be queried imperfectly

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

by a numerical solver, and is thus never used for the HMC rejection step. The surrogate model is assumed to provide a more reliable prediction of the likelihood function, and is thus used throughout the HMC sampling.

We demonstrate this approach on the problem of modeling the heliospheric transport of galactic cosmic rays (GCRs) [Rankin et al., 2022, Engelbrecht et al., 2022], a component of space weather that will influence the scheduling of future manned space missions. The likelihood function computed by the numerical solver fails a fraction of the time due to numerical instabilities, resulting in a noisy estimate of the true likelihood. A NN surrogate model — with an inductive bias towards smoothness — is trained on a large dataset of pre-computed PDE solutions. This provides significant improvement in both accuracy and computational cost per likelihood evaluation during MCMC. This surrogate likelihood is then combined with HMC to demonstrate state-of-the-art constraints on the global heliospheric transport parameters.

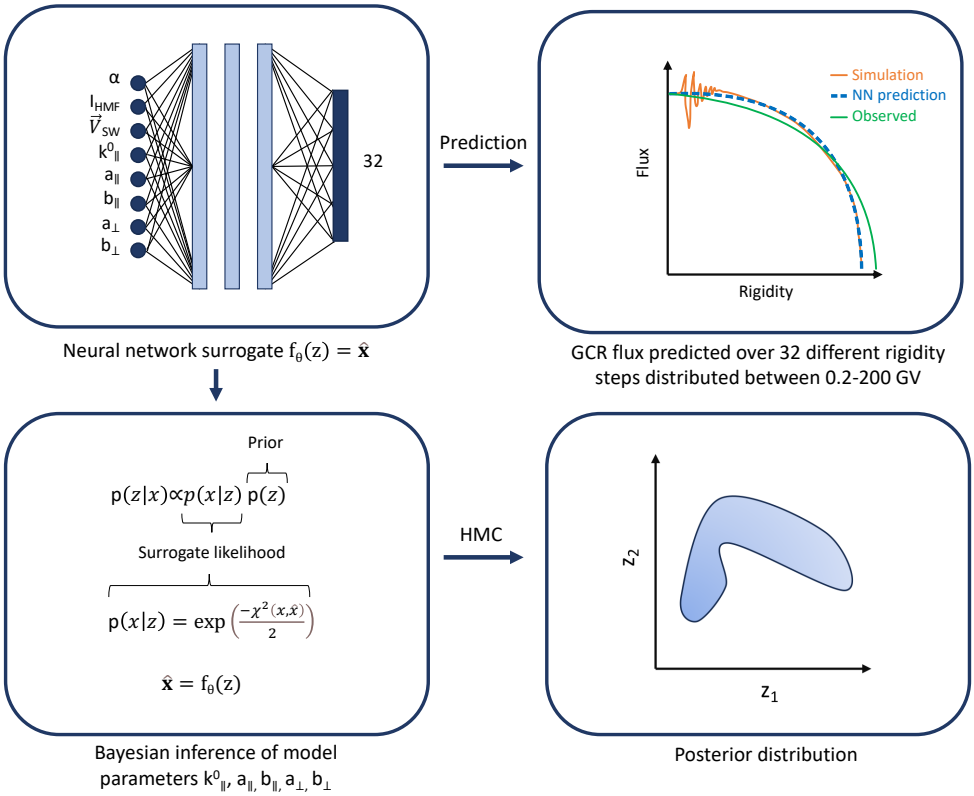

Figure 1: The NN (top left) takes as input 8 latent parameters of the heliosphere, and predicts the GCR flux at 1 astronomic unit (AU) for 32 rigidity steps. The NN is trained with targets from a numerical solver that suffers from instabilities (top right), and a surrogate likelihood is computed by comparing the NN outputs with observed fluxes (bottom left). This surrogate likelihood is used to sample from the posterior (bottom right) using HMC.

2 Related work

MCMC uses a Markov chain to stochastically explore the typical set of a target distribution and sample from an unknown multi-dimensional PDF [Neal, 1993]. The first MCMC algorithm, Metropolis-Hastings, was introduced by Metropolis et al. [2004]. Hamiltonian dynamics were integrated with MCMC to inform its steps towards high-likelihood samples and accelerate convergence to the typical set [Betancourt, 2018]. Hoffman and Gelman [2011] improved the convergence of HMC further with the introduction of the No-U-Turn-Sampling (NUTS) algorithm. The need for HMC to compute gradient and likelihood function calculations at each step has since inspired various methods to

accelerate MCMC sampling accelerate MCMC sampling [Cranmer et al., 2020, Magris and Iosifidis, 2023].

Rasmussen [2003] used Gaussian Processes to construct a surrogate likelihood function to reduce computational costs. Zhang et al. [2017] carried this work forward by applying random nonlinear bases along with efficient learning algorithms to construct a surrogate likelihood function to improve on the computational cost of Gaussian Processes. Later work applied NNs to accelerate HMC, including Li et al. [2019] who proposed an NN to approximate the gradients themselves, as opposed to a surrogate likelihood function. Levy et al. [2018] used NNs to predict the HMC transition kernel, and normalizing flow models were also used as a trainable kernel within the dynamic update of the HMC by Foreman et al. [2021a]. Foreman et al. [2021b] proposed the use of NNs to replace consecutive leapfrog steps. Finally, Dhulipala et al. [2022] applied Hamiltonian NNs to learn the Hamiltonian dynamics of an HMC sampler and continue sampling without gradient information.

Thus, NNs have been used to accelerate many of the calculations required for HMC, including the gradients, kernel updates, and leapfrog integrator calculations. However, to the best of our knowledge, no work has explored the use of NNs as a surrogate likelihood function.

3 Constraining the global heliospheric transport of GCRs

We demonstrate our method by using it to constrain five parameters characterizing the transport of GCRs within the heliosphere based on observations by detectors in orbit around Earth (1 AU). GCRs constitute a major radiation hazard for deep-space human exploration, and understanding their behavior will be critical for manned missions to Mars and beyond.

3.1 Problem set-up

The transport of GCRs within the heliosphere is described by the Parker equation [Parker, 1965]:

$$\frac{\partial J}{\partial t} = -\mathbf{V}_{sw} \cdot \boldsymbol{\nabla} J + \boldsymbol{\nabla} \cdot (\mathbf{K} \boldsymbol{\nabla} J) + \frac{\boldsymbol{\nabla} \cdot \mathbf{V}_{sw}}{3} \beta R^3 \frac{\partial}{\partial R} \left(\frac{J}{\beta R^2} \right) \tag{1}$$

where $J(\mathbf{r}, R)$ is the measured GCR flux at a given position, $\mathbf{r}$, and rigidity, R = (particle momentum) / (particle charge), while β is the particle speed divided by the speed of light. The various terms represent the interactions of GCRs with the solar wind, a plasma ejected by the Sun moving with velocity $\mathbf{V}_{sw}$, and the heliospheric magnetic field (HMF). The HMF is characterized by the intensity at 1 AU, I_{HMF}, the solar dipole tilt angle, α, and by the direction (positive or negative polarity). The diffusion tensor, $\mathbf{K}$, describes the GCR scattering and drifting due to HMF small and large-scale structures. It is characterized by a normalization constant, $k_{\|}^{0}$, and by the rigidity slopes of the diffusion coefficient (DC) in the directions parallel ($a_{\|}, b_{\|}$) and perpendicular ($a_{\perp}, b_{\perp}$) to the HMF.

Observations come from the data listed in Table A.1, and our model parameters are those used to describe the heliospheric magnetic field (HMF) conditions in Corti et al. [2019]. Our full methodology is visualized in Figure 1.

3.2 Surrogate neural network

Evaluations of model-predicted fluxes are slow, and solving the Parker equation for slightly different parameters is redundant as we expect the solution to vary smoothly. A surrogate NN has an inductive bias towards smoothness, can be evaluated quickly at inference time, and is differentiable. This conveniently solves all three obstacles to the use of HMC for Bayesian inference.

In order to train a NN surrogate, numerical solutions were computed for 6 million parameter values, similar to what was done in Corti et al. [2019]. This large initial computational cost is parallelizable. Solutions that demonstrated numerical instability were removed, resulting in 2,088,385 positive HMF polarity and 1,987,658 negative HMF polarity samples. Two surrogate NNs are trained separately on the negative and positive HMF polarity data. The resulting negative and positive polarity datasets are split into training sets (90%) and test sets (10%), with the latter used for early stopping and hyperparameter optimization.

The surrogate NNs consist of fully connected networks with two hidden layers of 256 units each, 8 inputs, and 32 outputs. They predict the modulated flux at 32 rigidity steps, uniformly distributed in logspace between 0.2 and 200 GV, from the 8 input parameters (tilt angle α, HMF intensity at Earth I_{HMF}, solar wind speed $\vec{V}_{SW}$, and parameters related to the parallel and perpendicular diffusion coefficients, DCs, $k_{\parallel}^{0}$, $a_{\parallel}, b_{\parallel}, a_{\perp}$, and $b_{\perp}$) for both positive and negative polarity periods. They were trained tominimize the mean squared error (MSE) using the Adam optimizer. Further details can be found in Table A.2.

Using the numerically stable model solutions as ground truth, the relative error of the NN predictions was found to be very low — between 1% and 2% at all rigidities and for both NN models. Thus, the NNs serve as highly accurate surrogates for the numerical model within the domain of parameter space used for training and testing. Figure A.1 shows three examples of the positive polarity NN predictions vs. model solutions from [Corti et al., 2019] with numerical instabilities of varying severity.

3.3 HMC

The dataset used to train our negative and positive polarity NNs (detailed in section 3.2) comprises model parameter data over 210 time intervals of interest. To infer the model parameters for each of these 210-time intervals, we perform HMC to generate samples from the posterior for each time interval. The five model parameters are $k_{\parallel}^{0}$, $a_{\parallel}, b_{\parallel}, a_{\perp}$, and $b_{\perp}$. Tilt angle, HMF intensity at Earth, and solar wind speed (α, I_{HMF}, and $\vec{V}_{SW}$) are fixed to their 1-year backward average for each interval, using data from OMNIWeb[1] and the Wilcox Solar Observatory[2]. The likelihood of a sampled parameter is defined as

$$p(x|z) = \exp(\frac{-\chi^2(x, \hat{x})}{2}) \tag{2}$$

where χ^2 is the standard chi-squared between the GCR fluxes predicted by the model ($\hat{x}$) and the observed GCR flux (x). This likelihood function is chosen as it gives a high likelihood of low χ^2 values and a low likelihood of high χ^2 values. It should be noted that the use of a surrogate NN gives a strong bias on likelihood smoothness, which is appropriate for our application but may not be universally applicable.

Since our NNs are trained on samples from a limited domain of parameter space, they should not be expected to generalize well outside this domain. Thus, we prevent the HMC from sampling outside the "trusted" domain with a prior distribution, $p(z)$. This prior is uniform in the domain of the training data and rapidly decays in every direction outside the domain. The strength of this penalty and other hyperparameters are detailed in A.2. This effectively prevents the HMC from accepting samples beyond the trusted region.

3.4 Results

Figure 2 shows an example of the probability distribution functions (PDFs) of the free parameters obtained with the HMC. The PDF is very narrow for the normalization of the DC ($k_{\parallel}^{0}$) and the slopes of the perpendicular DC ($a_{\perp}$ and $b_{\perp}$), meaning that these parameters are well constrained by the data, while it is wider for the slopes of the parallel DC ($a_{\parallel}$ and $b_{\parallel}$), meaning that these parameters are not well constrained by the data. This is expected since perpendicular diffusion dominates the transport processes in the majority of the heliosphere. These results are in agreement with what was found in Corti et al. [2019] using an ordinary least-square minimization procedure on the same AMS-02 data and numerical model. Our method improves on the results of Corti et al. [2019] by smoothing the numerical instabilities of the model and calculating posterior PDFs for all free parameters of the numerical model, as well as computing posterior PDFs on the predicted GCR fluxes.

[1] `https://omniweb.gsfc.nasa.gov/`

[2] `http://wso.stanford.edu/Tilts.html`

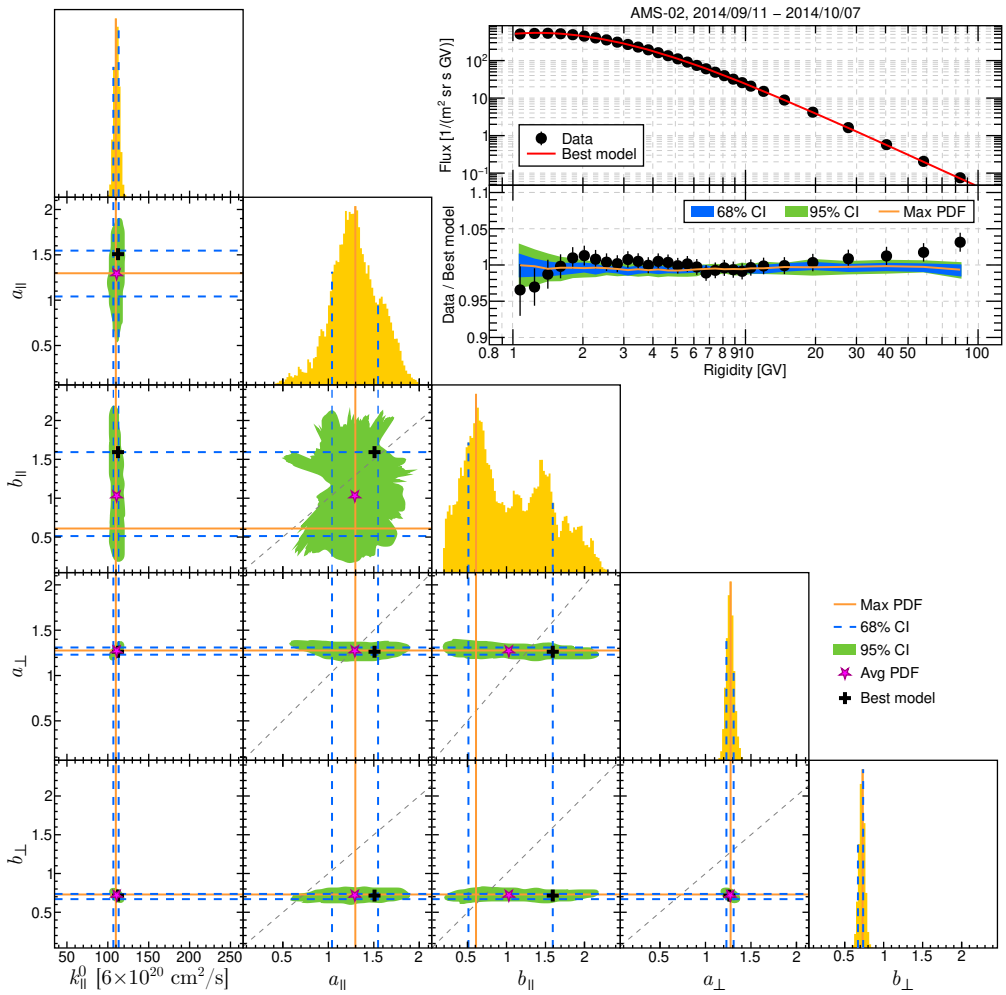

Figure 2: 2D and 1D PDFs of the diffusion coefficient parameters obtained from the HMC for GCR flux measured by AMS-02 in the positive polarity time interval 2014/09/11-2014/10/07. The top right panel shows a comparison of the maximum likelihood NN model (red line) with observations, together with the 68% and 95% credible intervals.

4 Discussion

To the best of our knowledge, this is the first demonstration of an NN surrogate likelihood with HMC. We have shown that this is a practical method that can be used to solve previously intractable problems. A potential problem of this approach was identified — that the NN surrogate should not be trusted outside the domain of the training data — and we proposed and tested a solution. The method was used to provide state-of-the-art constraints on the heliospheric transport of galactic cosmic rays. The resulting PDFs characterize the uncertainty on each transport parameter, which will enable better uncertainty estimates for GCR flux forecasts when planning future space missions. Nevertheless, we recognize that our method's applicability may not extend to all problem domains, given its pronounced bias toward likelihood smoothness with the use of a surrogate NN and the reliance on the availability of simulation data to train the NN.

5 Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 2236415, and the National Aeronautics and Space Administration under Grant Living with the Star No. LWS80NSSC20K1819. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or National Aeronautics and Space Administration.

The authors would also like to thank Arianna Bunnell, Yannik Glaser, Yusuke Hatanaka, and Amila Indika for their helpful conversations while preparing this manuscript. The technical support and advanced computing resources from the University of Hawai'i Information Technology Services Cyberinfrastructure are also gratefully acknowledged.

References

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

Radford M Neal. Probabilistic inference using markov chain monte carlo methods. 1993.

Sam Foreman, Taku Izubuchi, Luchang Jin, Xiao-Yong Jin, James C. Osborn, and Akio Tomiya. Hmc with normalizing flows, 2021a.

Sam Foreman, Xiao-Yong Jin, and James C. Osborn. Deep learning hamiltonian monte carlo, 2021b.

Daniel Levy, Matthew D. Hoffman, and Jascha Sohl-Dickstein. Generalizing Hamiltonian Monte Carlo with neural networks, 2018.

Somayajulu L. N. Dhulipala, Yifeng Che, and Michael D. Shields. Bayesian inference with latent hamiltonian neural networks, 2022.

Lingge Li, Andrew Holbrook, Babak Shahbaba, and Pierre Baldi. Neural network gradient hamiltonian monte carlo. *Computational statistics*, 34:281–299, 2019.

Cheng Zhang, Babak Shahbaba, and Hongkai Zhao. Hamiltonian monte carlo acceleration using surrogate functions with random bases. *Statistics and computing*, 27:1473–1490, 2017.

Carl Edward Rasmussen. Gaussian processes to speed up hybrid monte carlo for expensive bayesian integrals. In *Seventh Valencia international meeting, dedicated to Dennis V. Lindley*, pages 651–659. Oxford University Press, 2003.

Jamie S. Rankin, Veronica Bindi, Andrei M. Bykov, Alan C. Cummings, Stefano Della Torre, Vladimir Florinski, Bernd Heber, Marius S. Potgieter, Edward C. Stone, and Ming Zhang. Galactic cosmic rays throughout the heliosphere and in the very local interstellar medium. *Space Science Reviews*, 218(5):42, 2022. doi: 10.1007/s11214-022-00912-4.

N. Eugene Engelbrecht, F. Effenberger, V. Florinski, M. S. Potgieter, D. Ruffolo, R. Chhiber, A. V. Usmanov, J. S. Rankin, and P. L. Els. Theory of cosmic ray transport in the heliosphere. *Space Science Reviews*, 218(4):33, 2022. doi: 10.1007/s11214-022-00896-1.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 12 2004. ISSN 0021-9606. doi: 10.1063/1.1699114. URL `https://doi.org/10.1063/1.1699114`.

Michael Betancourt. A conceptual introduction to hamiltonian monte carlo, 2018.

Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo, 2011.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. doi: 10.1073/pnas.1912789117. URL `https://www.pnas.org/doi/abs/10.1073/pnas.1912789117`.

Martin Magris and Alexandros Iosifidis. Bayesian learning for neural networks: an algorithmic survey, 2023.

E. N. Parker. The passage of energetic charged particles through interplanetary space. 13(1):9–49, 1965. doi: 10.1016/0032-0633(65)90131-5.

Claudio Corti, Marius S. Potgieter, Veronica Bindi, Cristina Consolandi, Christopher Light, Matteo Palermo, and Alexis Popkow. Numerical modeling of galactic cosmic-ray proton and helium observed by AMS-02 during the solar maximum of solar cycle 24. *The Astrophysical Journal*, 871 (2):253, 2019. doi: 10.3847/1538-4357/aafac4.

6 Appendix

Table A.1: Dataset information

Experiment	Period	Position (r, θ)	Measurements
PAMELA [b]	2006 – 2014	Earth LEO	H: 0.4 – 50 GV
AMS-02 [c]	2011 – 2019	Earth LEO	H, He: 1 – 100 GV

[b] PAMELA monthly protons are available at the ASI Cosmic Ray Database: `https://tools.ssdc.asi.it/CosmicRays/`.
[c] AMS-02 daily protons are available at : `https://ams02.space/sites/default/files/publication/202105/table-s1-s2824.csv`.

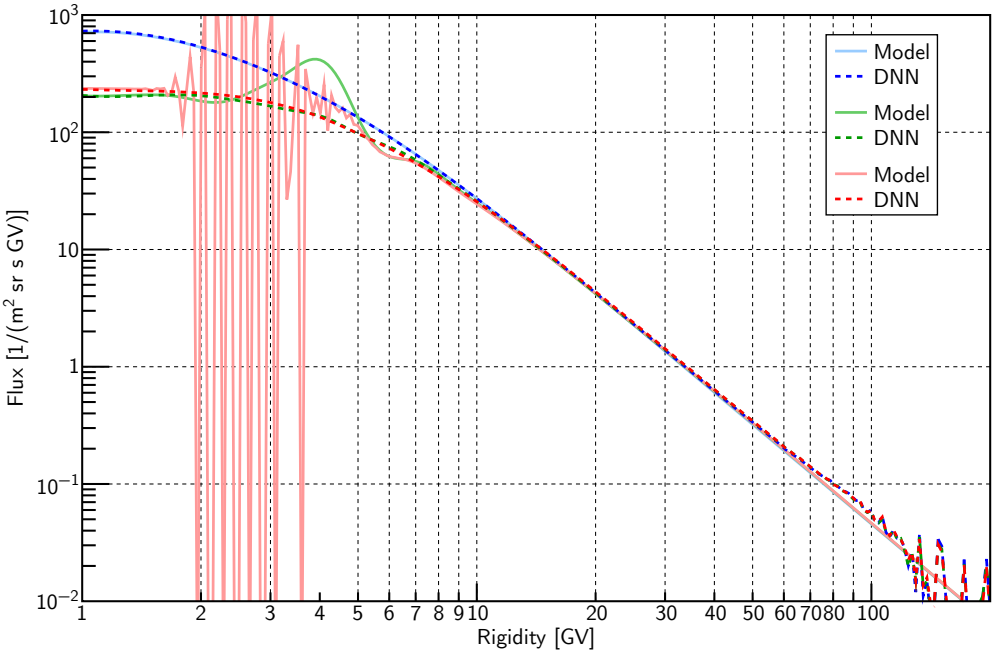

Figure A.1: Modulated proton flux as a function of rigidity computed via the model from Corti et al. [2019] and our NN for three typical examples with varying levels of numerical instability. The NN incorporates our intuition that the flux varies slowly with rigidity and predicts a smooth curve even when the model solution exhibits numerical instabilities. The three examples shown are test samples not used for training the NN.

Table A.2: Hyperparameter choices for the NN and HMC. Optimal hyperparameters were selected by hand.

Neural Network	Initial Learning Rate	$1 * 10^{-6}$
	Max Epochs	100
	Optimizer	Adam
	LR Schedule	ReduceLROnPlateau
	Early Stopping	Monitored Validation loss
	Regularization	L2, weight $1 * 10^{-6}$
	Loss Function	MSE
	Batch Size	128
	Number of Layers	3
	Number of Neurons	256, 256, 32
	Activation Functions	SeLU, SeLU, Linear
	Dataset size	2,088,385 positive polarity 1,987,658 negative polarity
	Train/Test Split	90/10
	Data Normalization	Inputs: Min-Max scaling Outputs: Log scaling
Hamiltonian Monte Carlo	Number of Results	110,000
	Kernels	NUTS DualAveragingStepSizeAdaptation
	Number of Burn-in steps	1,000
	Number of Adaption Steps	800
	Step Size	$1 * 10^{-3}$
	Max Tree Depth	10
	Max Energy Difference	1,000
	Unrolled Leapfrog Steps	100
	Number of Time Intervals	210
	Penalty for out-of-bound samples	$1 * 10^{6}$