NNMFit – A DiffuseNuMu Analysis Tool

Erik Ganster, Richard Naab

Physics
Institute III B

RWTH AACHEN
UNIVERSITY

Outline

- Introduction
  - What is NNMFit?
  - Where to find it?
- What does NNMFit do?
- How to use it
- Recent developments
- Summary

NNMFit – What it is and Where to find

- Theano-based analysis framework for diffuse analyses
- Developed for the measurement of the energy spectrum of astrophysical muon-neutrinos
- Very modular and configurable setup

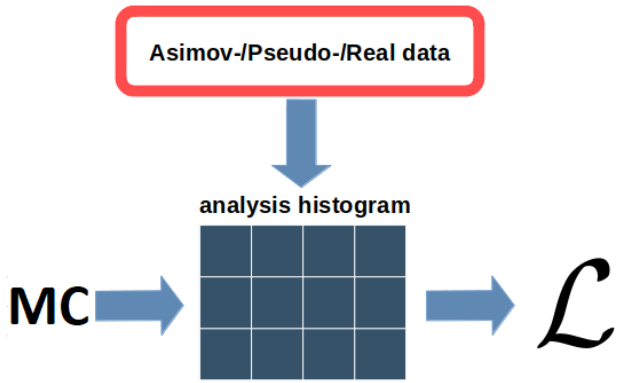
- Growing userbase:
  - ✓ Christian: diffuse + galactic plane
  - ✓ Jöran: 9.5yr diffuse numu
  - ➢ Sally: Upgoing Muon Cross Section
  - ➢ Sarah: Intermediate energy cross-section
  - ➢ Richard, Zelong, Erik: (first) GlobalFit combinig tracks and cascades
  - ➢ You?

- Now living in the new IceCube github organization: icecube/NNMFit
- Richard started working on an NNMfit wiki (also on github): NNMFit/wiki
- Slack channel #nnmfit
- Increased development activities since the GlobalFit efforts started

Binned Likelihood Analyses - Conceptually

- Comparison of a MonteCarlo prediction with either real, pseudo or Asimov data an analysis histogram using some Likelihood function:
  1. Filling the data into the analysis histogram
  2. Calculate MonteCarlo prediction for a specific hypothesis
  3. Compute Likelihood

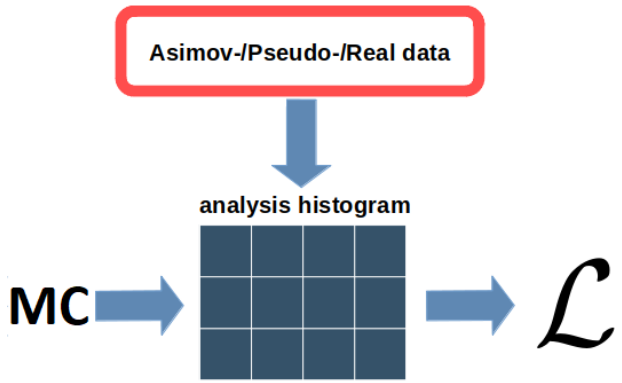
- The MonteCarlo prediction consists of several components which themselves can depend on multiple parameters each
  - Astrophysical Neutrino Flux
  - Conventional Atmospheric Flux
  - Prompt Neutrino Flux
  - Atmospheric Muons
  - …



Binned Likelihood Analyses - Conceptually

- Comparison of a MonteCarlo prediction with either real, pseudo or Asimov data an analysis histogram using some Likelihood function:
  1. Filling the data into the analysis histogram
  2. Calculate MonteCarlo prediction for a specific hypothesis
  3. Compute Likelihood

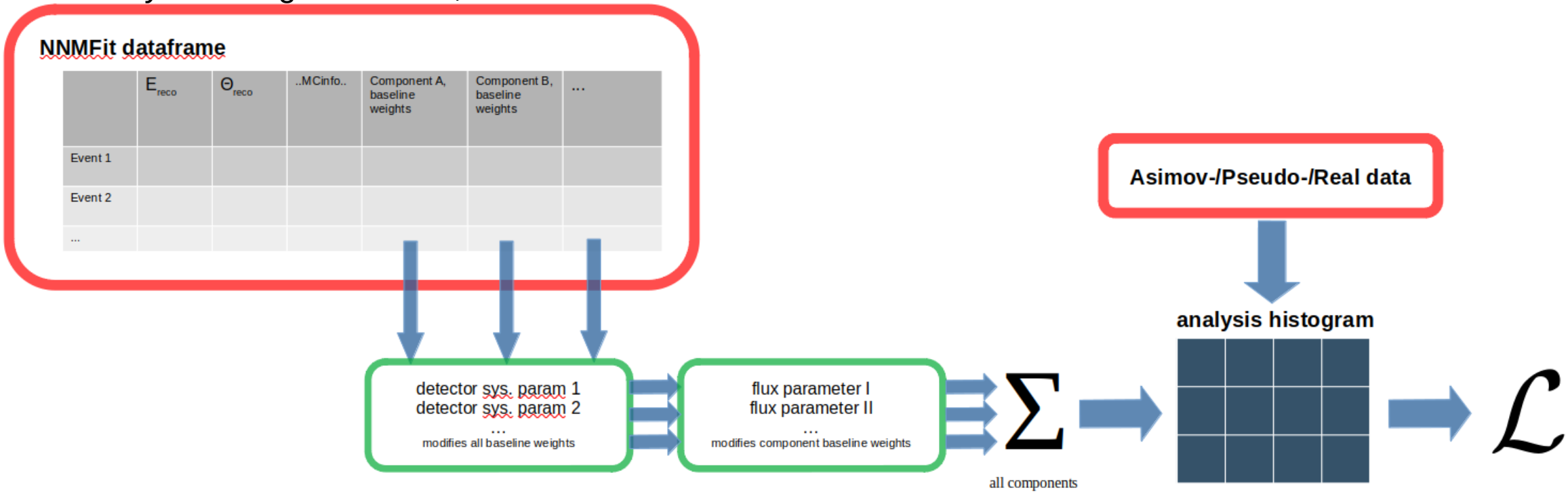
- The MonteCarlo prediction consists of several components which themselves can depend on multiple parameters each
  - Astrophysical Neutrino Flux
  - Conventional Atmospheric Flux
  - Prompt Neutrino Flux
  - Atmospheric Muons
  - …



- NNMFit builds a computational Theano-graph to calculate the Likelihood
  - This graph is a function of all parameters that are used to model all flux components, detector systematics, …
- Why Theano?
  - Fast, internal C++ compilation of the graph
  - Allows gradient calculation of the Likelihood function with respect to the input parameters

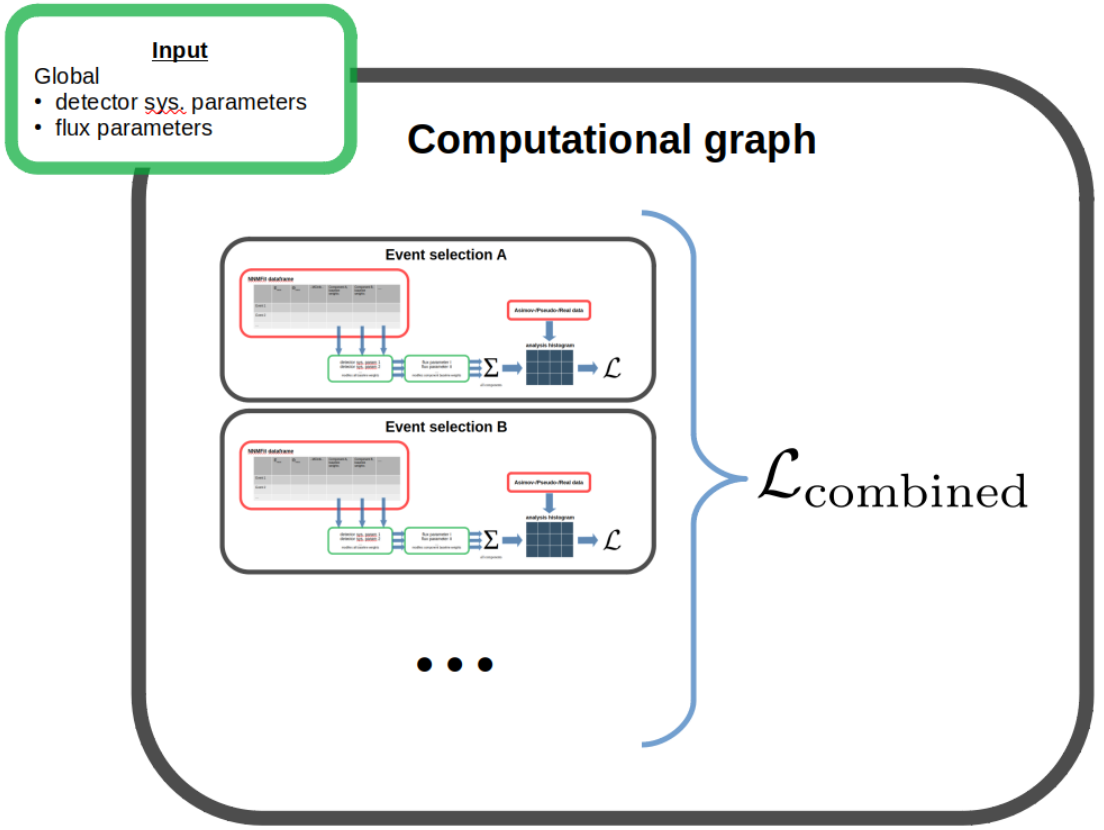
NNMFit – Overview

- Main input for NNMFit is a pandas dataframe, containing all reco + MC truth quantities
  - It also includes e.g. conventional + prompt neutrino baseline weights
- Based on this dataframe all flux + detector systematic parameters get applied by modifying the baseline weight within the Theano graph
- The analysis histogram is built, and the Likelihood is calculated



NNMFit – Detector Configs

- NNMFit is not limited to a single analysis histogram
  - Historically, multiple histograms were needed to combine IC59, IC79 and IC86
  - ➢ “Detector configs” in NNMFit slang
- Global parameters are shared between all detector configs
  - Flux parameters
  - Detector systematics
- For the GlobalFit efforts, we used these detector configs for different event selections:
  - NuMu tracks
  - Cascade signal
  - Cascade starting tracks
  - Cascade muon selection
  - …



NNMFit – Recent Developments

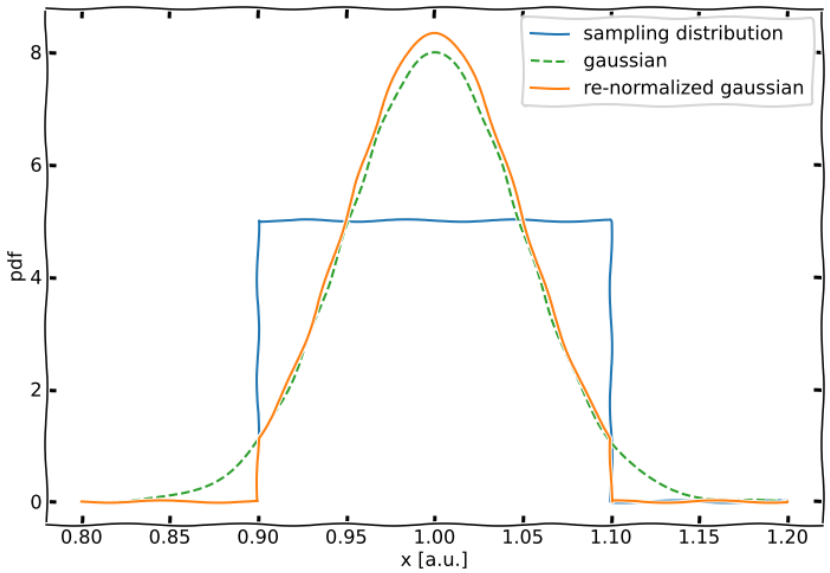
- Added python 3 support
- Adopted NNMFit to use SnowStorm simulation sets
  - Read SnowStorm MC and its parameters
  - Re-weighting of SnowStorm MC sets (details later)
- Extended detector_configs capability to deal with multiple event selections in parallel
- Added ability to read MuonGun simulations
  - So far, NNMFit has been very streamlined for NuGen
- Updated atmospheric (conventional and prompt) neutrino predictions to the most recent MCEq version
  - NNMFit is not limited to MCEq! NuFlux can easily be added
- Finalized SAY-Likelihood implementation
- Main `GlobalFit/master` branch
  - Development in `GlobalFit/dev/…` branches, updates to (GlobalFit) master via PullRequests
- Non GlobalFit NNMFit branch `main_stable_py3`
  - No SnowStorm support, relies on “traditional”/old systematic sets + treatment

NNMFit – SnowStorm Reweighting

- Re-weighting of the SnowStorm parameters within the MC set to some (arbitrary) distribution on analysis level (e.g. normal dist.)
  - Directly yields a MC prediction for a specific choice of nuisance parameters, i.e. the current hypothesis, in the fit

- Currently implemented distribution:
  - Gaussian (configurable width)
  - Uniform/Box
  - Symmetrical Gaussian
    - Ensures a symmetric (with respect to the mean value) range for applying the reweigthing
  - Symmetrical Box

➢ Both “symmetrical” reweighting functions work; however, they are non continuous functions and therefore cause minimizer issues…



$$w_i = \frac{p_{sys}\,(sys_i, \xi_i)}{p_{sys}^{sim}(sys_i)} \cdot \ldots$$

p_{sys} : reweighting distribution

p_{sys}^{sim} : sampling distribution

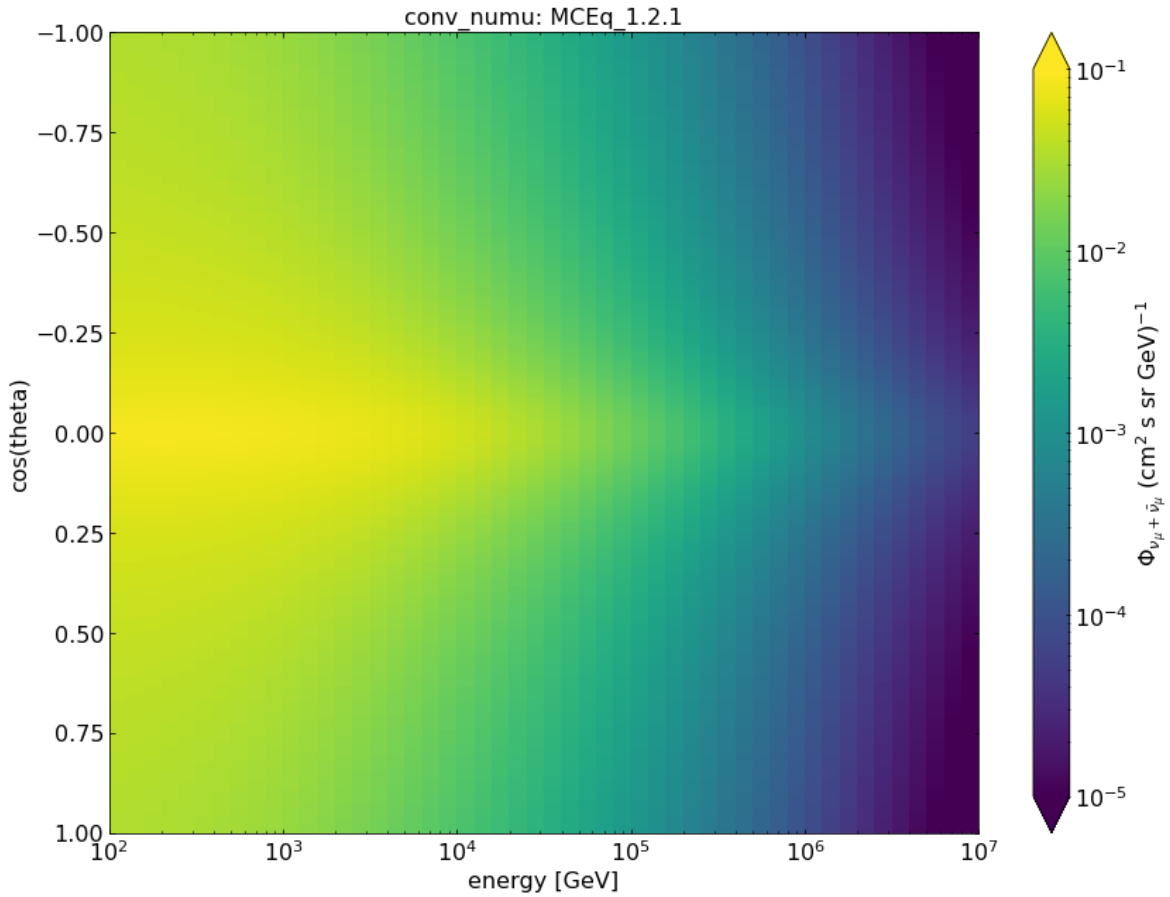
sys_i : event′s systematic value

ξ_i : nuisance parameter

NNMFit and MCEq

- NNMFit uses MCEq to calculate the conventional + prompt neutrino flux weights
- Weight calculation using pre-built 2d-splines
- One spline is a combination of primary CR + hadronic interaction model
  - H4a, H3a, GSt-3gen, GST4-gen
  - SIBYLL2.3c, QGSJET, DPMJET, EPOS-LHC
- Using `MSIS00_IC SouthPole` atmosphere and averaging the prediction from all months
- Using the Barr/Bartol parameters for modelling uncertainties

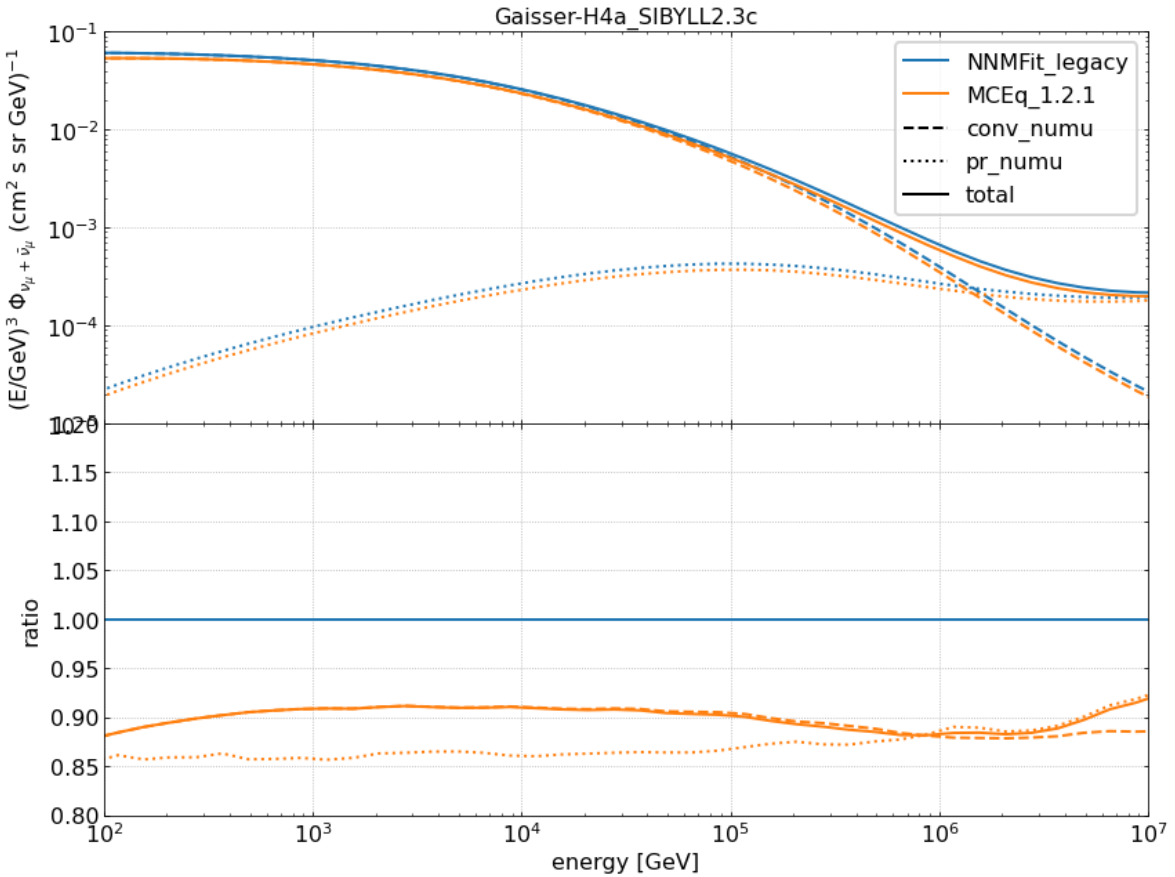
➢ Updated the splines from an old MCEq release to latest v1.2.1 release



NNMFit – MCEq Splines

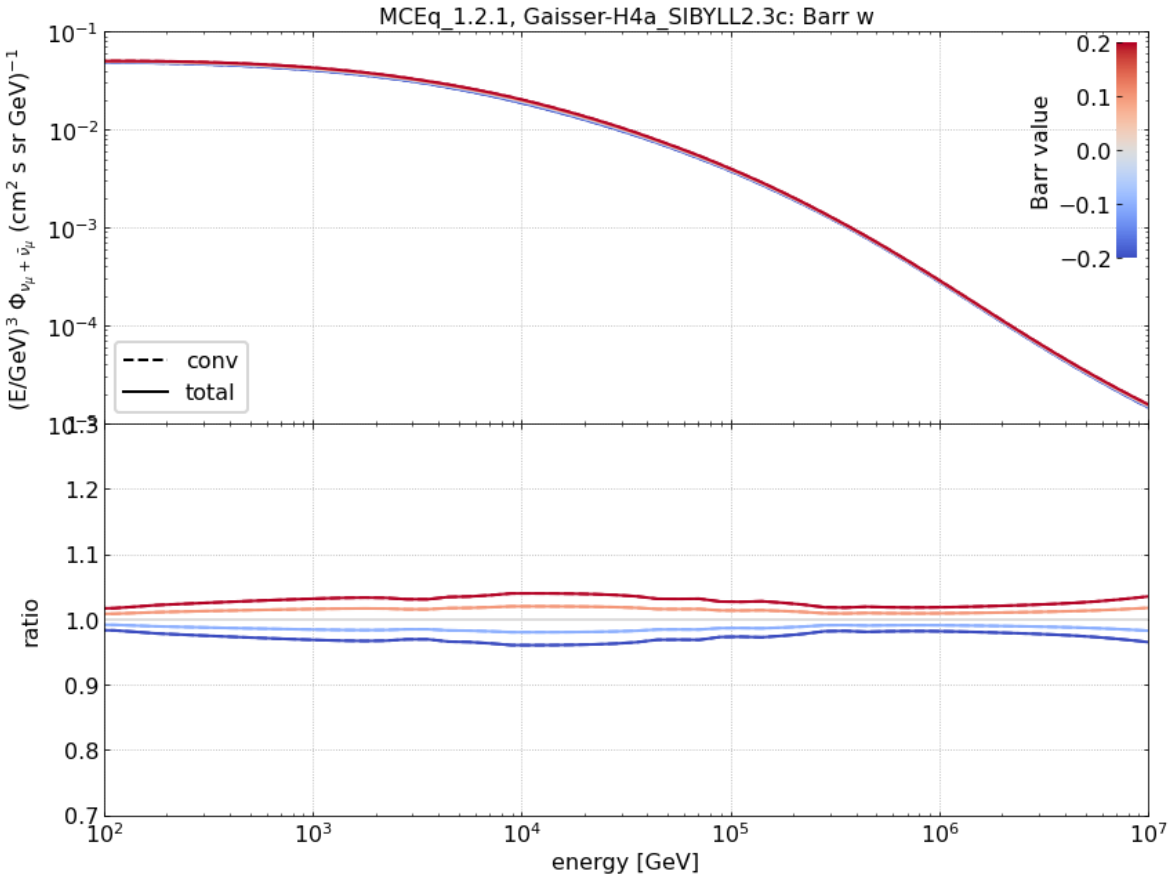
- Comparing both splines showed an overall ~10% lower neutrino rate and some shape differences
  - “legacy” splines from mid 2019 (MCEq_RC1)
  - Major rewrite of the MCEq core for v1.0 release
  - Anatoli strongly recommended the use of v1.2.1

- Jöran repeated his fit: No significant change of the signal parameters, only and ~10% increased conv_norm



NNMFit – MCEq Splines

- Comparing both splines showed an overall ~10% lower neutrino rate and some shape differences
  - “legacy” splines from mid 2019 (MCEq_RC1)
  - Major rewrite of the MCEq core for v1.0 release
  - Anatoli strongly recommended the use of v1.2.1
- Jöran repeated his fit: No significant change of the signal parameters, only and ~10% increased conv_norm
- Also re-evaluated the effects of the Barr/Bartol parameters on the neutrino flux



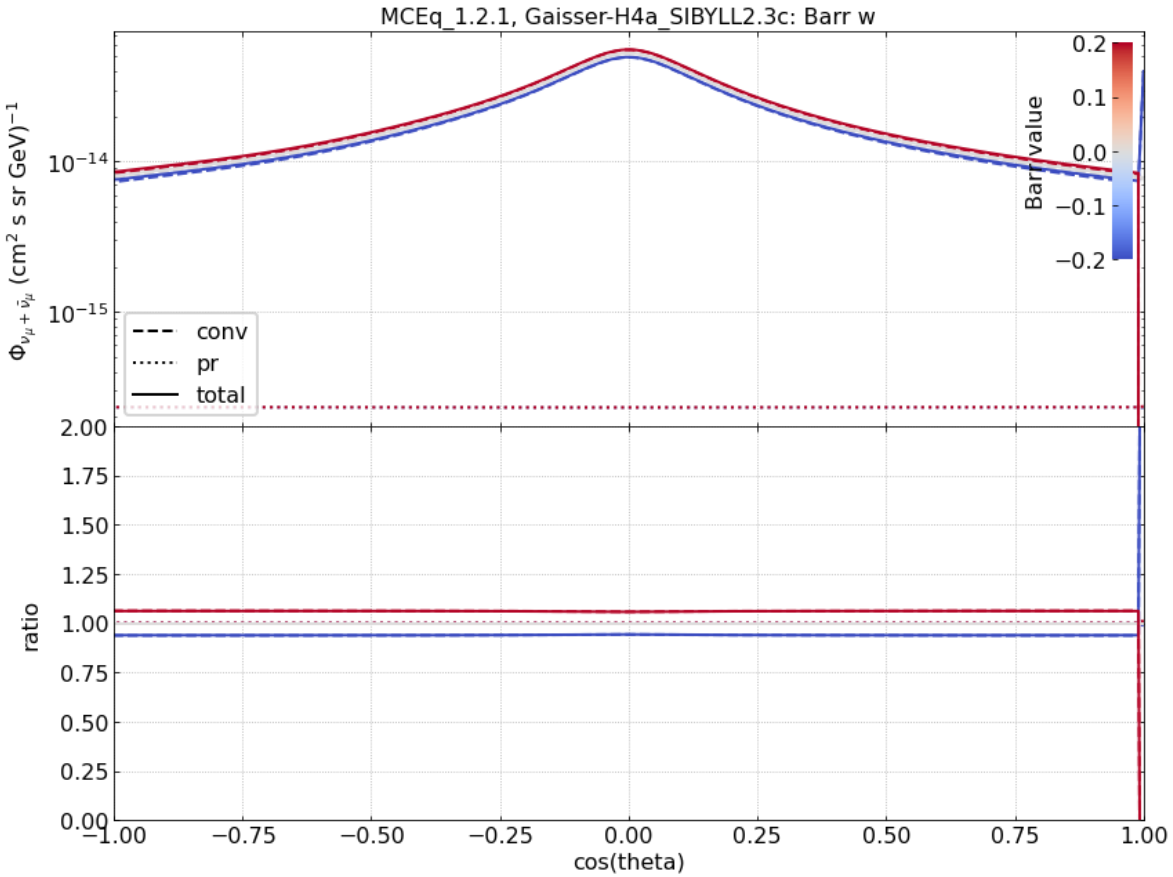
NNMFit – MCEq Splines

- Comparing both splines showed an overall ~10% lower neutrino rate and some shape differences
  - “legacy” splines from mid 2019 (MCEq_RC1)
  - Major rewrite of the MCEq core for v1.0 release
  - Anatoli strongly recommended the use of v1.2.1

- Jöran repeated his fit: No significant change of the signal parameters, only and ~10% increased conv_norm

- Also re-evaluated the effects of the Barr/Bartol parameters on the neutrino flux

➢ Issue/way to large Barr/Bartol flux gradient in the first downgoing bin…
  ➢ Interpolate that bin?



NNMFit –Summary and Outlook

- ✓ Adopted NNMFit to use SnowStorm simulation sets
  - ✓ Read SnowStorm MC and its parameters
  - ✓ Re-weighting of SnowStorm MC sets with different distributions
- ✓ Extended detector_configs capability to deal with multiple event selections in parallel
  - ✓ Northern_tracks
  - ✓ Cascades
- ✓ Updated atmospheric (conventional and prompt) neutrino predictions to the most recent MCEq version
- ✓ Finalized SAY-Likelihood implementation
- ✓ Started a NNMfit wiki on github: NNMFit/wiki
  - ➢ Please feel free to edit it!

- ➢ Need additional testing of the different reweighting distributions
- ➢ NuGen + MuonGun are now implemented: Add CORSIKA for future/more event selections?

- ❖ **Active development on github** icecube/NNMFit
- ❖ **Check it out and try to use it!**
  - ❖ **Read access for all members of the icecube organization**
  - ❖ **Just ping us in #nnmfit on slack and we will ad you to the NNMFit team so you can create you r own branch**

Appendix

NNMFit – Configuration Files

- NNMfit heavily uses config files for configuring the actual analysis, signal hypothesis, nuisance parameters, minimizer settings, …

- Most important configs:
  - `main.cfg`
  - `analysis_config.yml`
  - `detector_config.cfg`
  - `components.yml`

➢ Not going to all all configs in detail here, example with a basic fit and what settings apply in the Hands-On session

Example – Installation and Initial Setup

Example – Create a Dataset for NNMFit

Example – Define an Analysis/Event Selection/Detector Configuration

Example – Run a (simple) Fit