Daya Bay Data Release

Vitalii Zavadskyi (zavadskyi@jinr.ru), Maxim Gonchar
Joint Institute for Nuclear Research, Dzhelepov Laboratory of Nuclear Problems

Introduction

Daya Bay experiment has collected one of the largest datasets of IBD events with the final-state neutron captured on gadolinium (5.5 million) [1]. The dataset was used to obtain the world-leading measurement of $\sin^2 2\theta_{13}$ with a relative uncertainty of 3%.

The joint analysis of the full dataset of Daya Bay and the first 59 days of data taking of JUNO [2] demonstrates a significant improvement in the measurement of $\sin^2 2\theta_{12}$ and Δm^2_{21}.

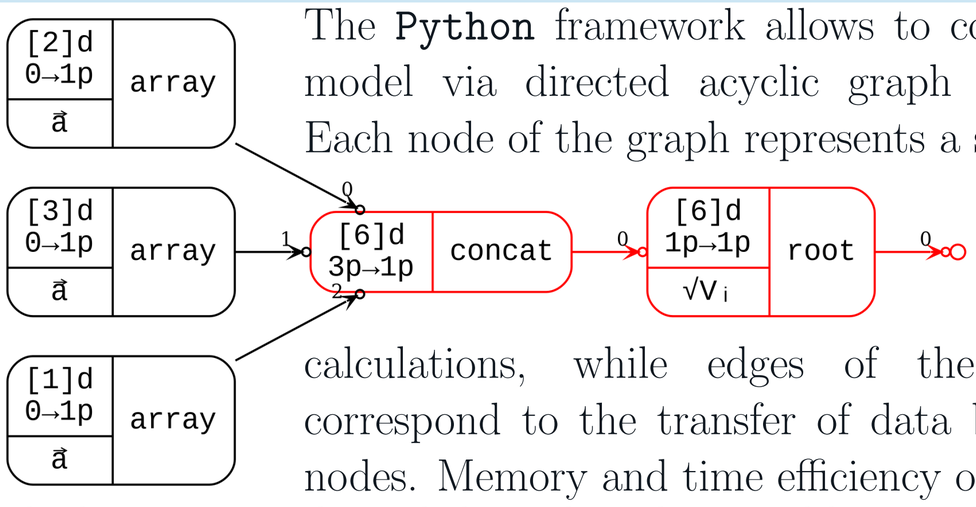
In this poster, we would like to present the `dag-modelling` framework, discuss its internal structure and demonstrate its use for the task of Daya Bay data preservation.

Model

The model of the experiment includes three sources of $\overline{\nu}_e$: main components of the nuclear fuel (Huber-Mueller spectrum), spent nuclear fuel, and off-equilibrium correction. Also, we take into account corrections: effect of inner acrylic vessel, energy resolution, and liquid scintillator non-linearity. Total number of systematic parameters is 152 (1057 if we include Huber-Mueller uncertainties). Also, the model includes 21 free parameters to vary the shape of $\overline{\nu}_e$ spectra.

Minimization of the chi-squared function with 173 parameters takes 10 minutes. Minimization could be sped up by 15 times, if we put systematic parameters in the covariance matrix.

`Dag-modelling` framework



The `Python` framework allows to construct model via directed acyclic graph (DAG). Each node of the graph represents a set of calculations, while edges of the graph correspond to the transfer of data between nodes. Memory and time efficiency of the calculations are achieved through caching and lazy evaluation.

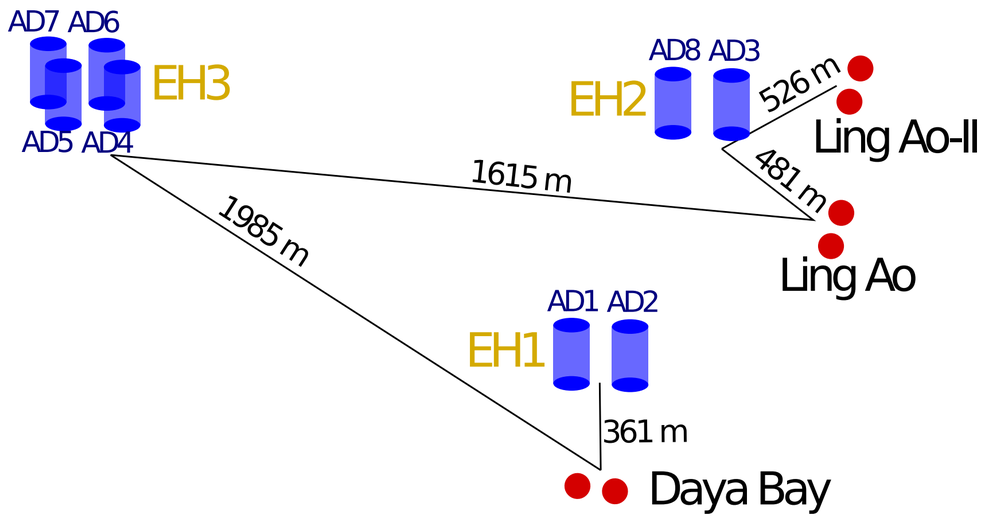
The DAG approach originates from `GNA` [3] (our previous framework). It is a `C++` framework with a large legacy. Switch to `Python` makes `dag-modelling` easy to extend, reduces installation difficulties, allows to install the package via `PyPI`.

Data publication

The dataset could be divided into light and heavy parts. The light part contains all necessary data for the oscillation analysis; the heavy part is supplemented with IBD event trees for each detector. `Zenondo` [4] and `NHEPSDC` [5] contain both versions. `GitHub` [6] and `PyPI` [7] contain only the light part. Datasets are converted into: `root`, `npz`, `hdf5` and `tsv`.

To work with the oscillation part of the dataset, `dayabay-model` package was developed [8]. It can load the dataset and obtain the observation of the Daya Bay experiment. The final observation was cross-checked with the `GNA` model, which was used to obtain the final result [1].

Daya Bay experiment

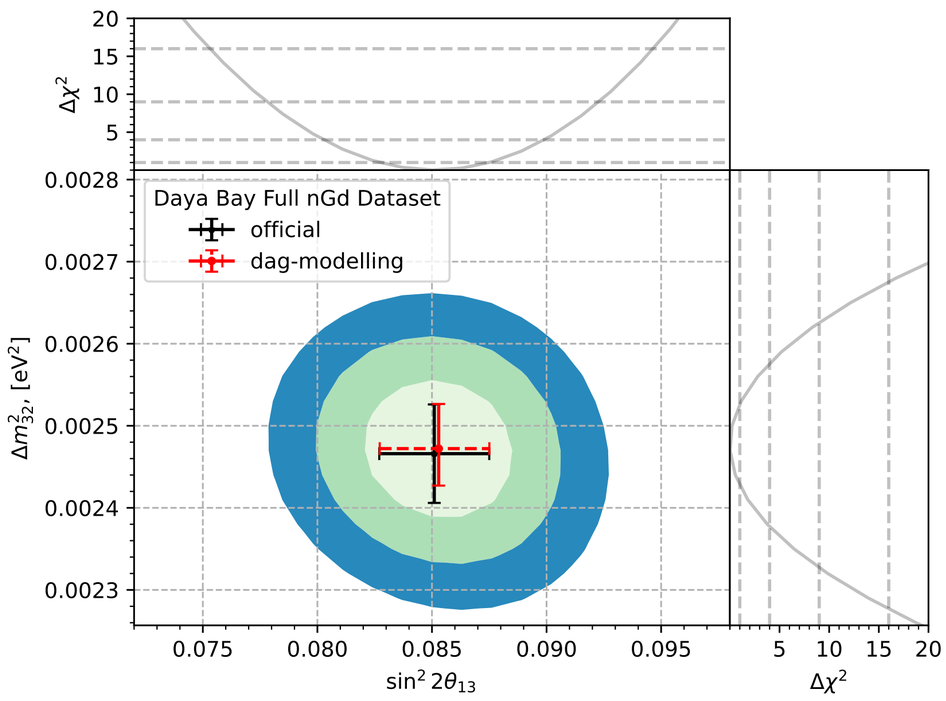


The Daya Bay experiments measured antineutrino flux from 3 nuclear power plants (NPP), with 6 nuclear cores, using 8 antineutrino detectors (AD) located at three experimental halls (EH). Each nuclear core contains 4 isotopes of the nuclear fuel. Modelling the experiment has several requirements: flexible creation of detectors, reactors, and related parameters; optimal calculation of $\overline{\nu}_e$ flux.

	EH1	EH2	EH3
ν_e candidates	2 236 810	2 544 895	764 514
ν_e candidates [1/day]	1342.29	1191.18	300.57
Background events [1/day]	21.93	16.20	4.47
background / signal, %	**1.63**	**1.36**	**1.58**

Numbers of $\overline{\nu}_e$ and background events and their ratio after 10 years of data taking.

Results



The result with the `dayabay-model` [8] matches with the official [1] result at the 0.1σ level.

Acknowledgements

Author thanks Daya Bay collaboration and dagflow-team [9] for the publication of the full dataset.

References

[1] *PRL* **130** (2023) 161802.
[2] arXiV: 2511.14593.
[3] https://git.jinr.ru/gna/gna.
[4] Zenodo: 10.5281/zenodo.17587229.
[5] NHEPSDC: 10.12402.
[6] https://github.com/dayabay-experiment/dayabay-data-official.
[7] PyPI: dayabay-data-official.
[8] https://github.com/dagflow-team/dayabay-model.
[9] https://git.jinr.ru/dagflow-team.