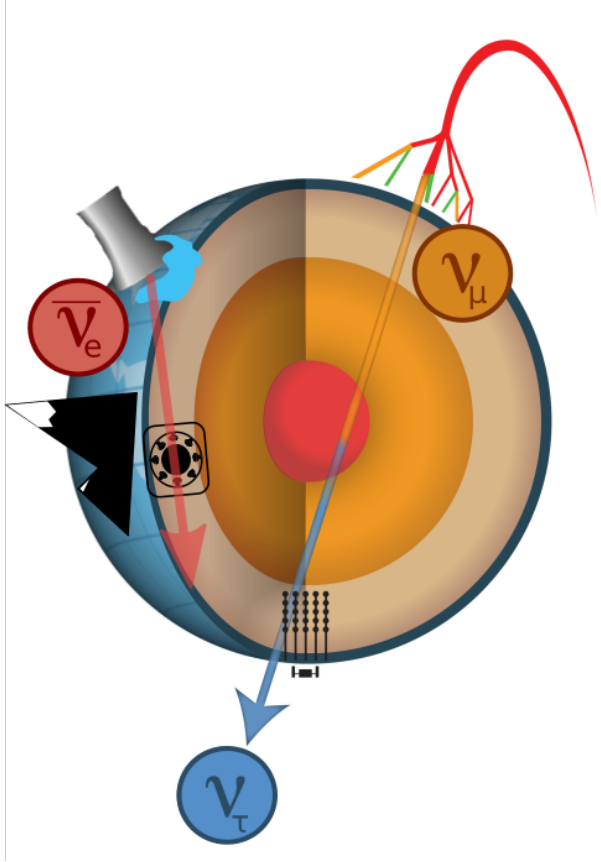
Computational challenges in high-statistics ν oscillation experiments: The PISA framework



GDR Neutrino Meeting | Strasbourg | 6 November 2018

Thomas Ehrhardt (for the PISA authors)

Motivation

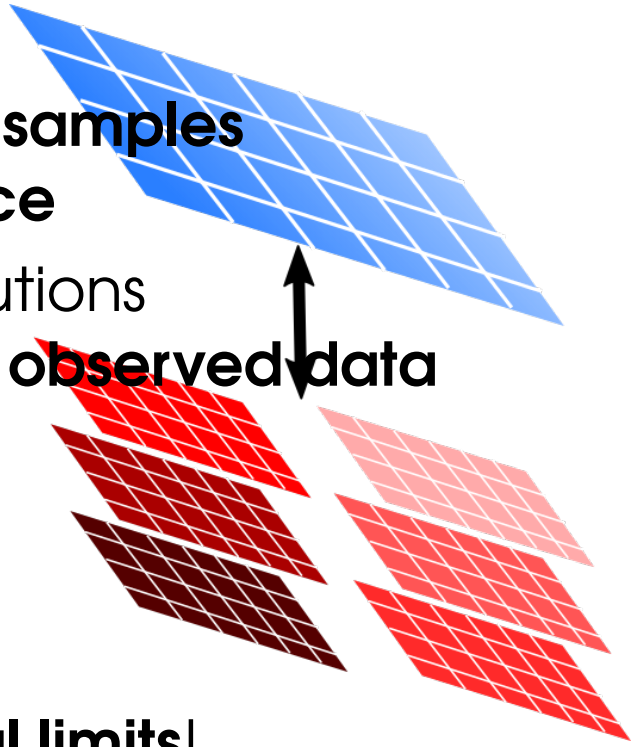
- **analysis in ν oscillation experiments:**

  compare **data** to distributions of **neutrino events** simulated under **different physics models (or parameters)**

- **typical issues:**

  required to generate **large numbers of samples** from **multi-dimensional parameter space**

  statistical precision of sampled distributions needs to (significantly) **exceed that of observed data**



- **most straightforward solution:**

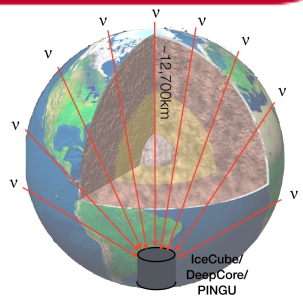
  direct histogramming of large enough MC samples

  often impossible due to **computational limits**!

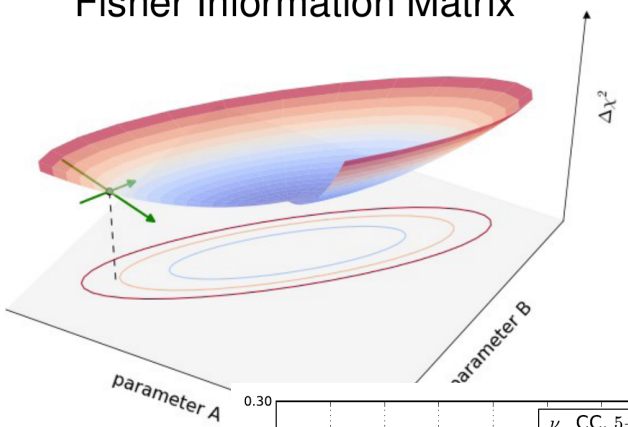
$\Rightarrow$ **PISA**

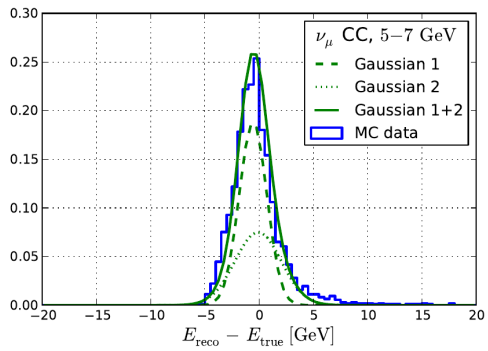
Introduction

- PISA **originally** served as the "**PI**NGU **S**imulation & **A**nalysis" framework
  - **fast** methods to determine **NMO sensitivity**



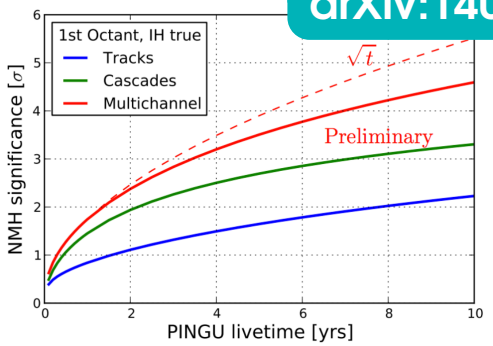
Fisher Information Matrix





- **factorise generation of NMO templates**: flux $\times$ oscillation $\otimes$ detector response
- **manual parameterisation** of detector response

arXiv:1401.2046v1



- by **today**, it has involved into a much more general tool

What is PISA?

Pisa (disambiguation)

From Wikipedia, the free encyclopedia

Pisa is a city in Tuscany, Italy.

Pisa or **PISA** may also refer to:

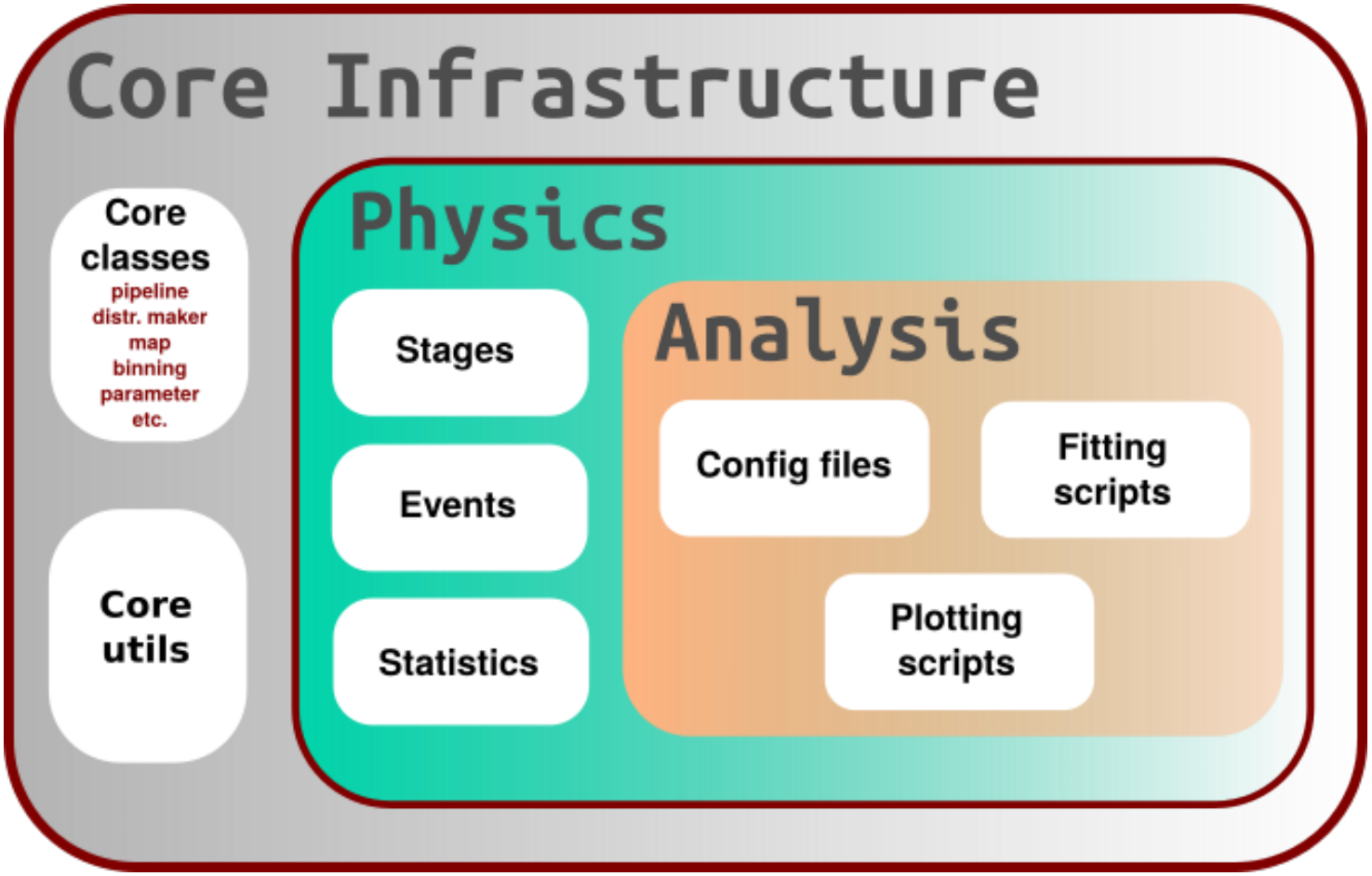
a software framework developed by the IceCube collaboration...

python™ ... written in Python ... ICECUBE SOUTH POLE NEUTRINO OBSERVATORY

... that has the goal of **enabling physics analyses, by**:

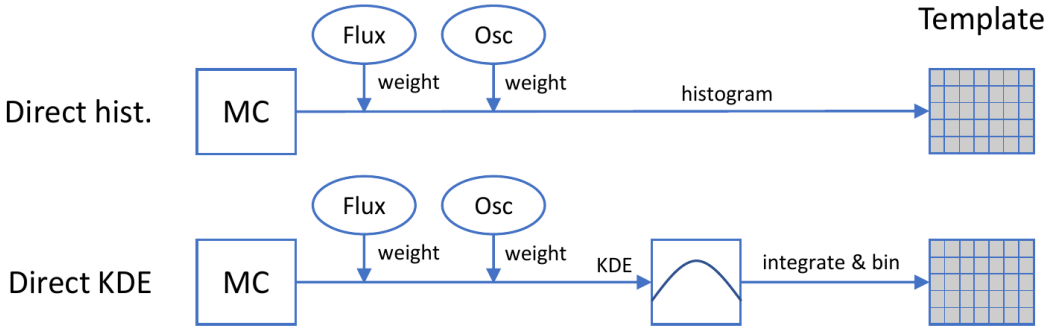
- providing **commonly required functionality**
- implementing **tools to deal with (low-statistics) MC** simulation
- taking care of **reproducibility & documentation**
- providing **performance and accuracy**

PISA architecture



MC event reweighting technique

- allows use of **single set of MC events**: calculate new event weight each time value of physics or nuisance parameter changes
- possible for **independent physics processes** (here: ν production, oscillation, detection, reconstruction)
- **binning events** in observable dimensions = **MC integration**



accuracy $\propto 1/\sqrt{N}$

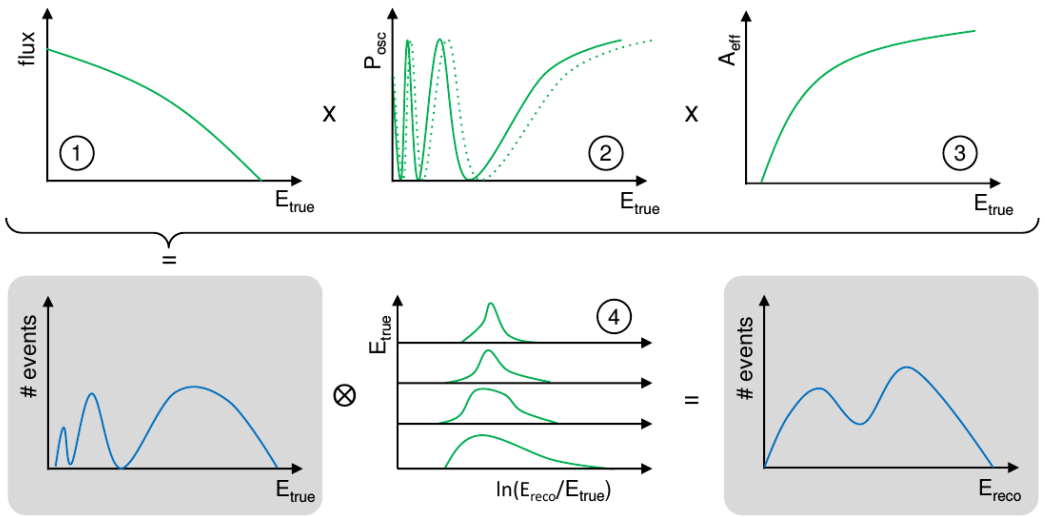
$\Rightarrow$ common practice: **smoothing of event distributions**

neutrino oscillation experiment

- **Kernel Density Estimation (KDE)**:
  - smoothed distribution as **weighted sum over kernel functions** placed at each event's reconstructed observables
  - here: Gaussians with variable bandwidth

Staged approach

- **alternative to the two standard event reweighting variants**
- introduce **stages to reflect independent processes** occurring in experiment
- **exploit computational simplifications** where possible



template =

flux $\times$ osc. prob. $\times$ eff. area $\otimes$ resolutions

- stages calculate **transformations on a grid**

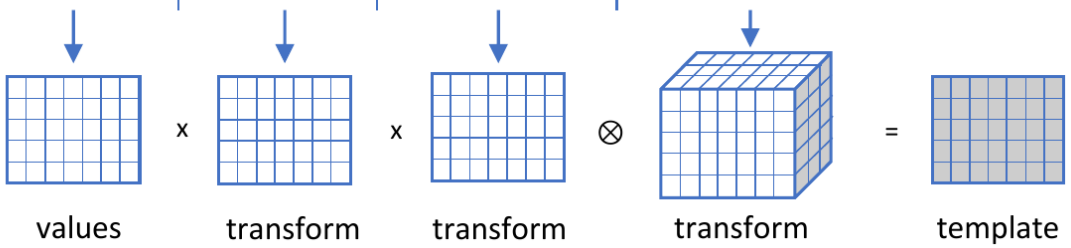
$\Rightarrow$ applied **differentially**

$\Rightarrow$ **grid choice adapted** to each stage

$\Rightarrow$ **exploit caching**

Staged approach: operating modes

	Flux	Osc	Det	Reco
Staged (w/ smoothing)	evaluate model on grid	evaluate model on grid	histogram MC + smooth + spline	KDE MC, integrate & bin
Staged w/o smoothing			histogram MC	histogram MC
Truth			evaluate model on grid	evaluate model on grid



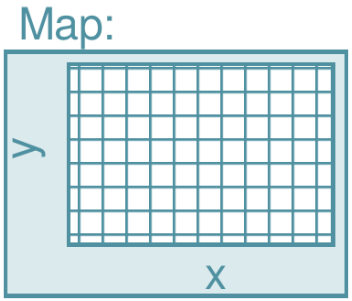
- flux computed from tabulated data, oscillation probabilities from (semi-)analytic formulae
- MC events only required for detector response stages
- can select suitable **smoothing methods** adapted to physics of stage

  $\Rightarrow$ **increases effective amount of MC statistics**

Stages & data structure in PISA

- stages represent different physics effects, interfaced with each other within a ***pipeline***
- a ***service*** is a concrete implementation of a stage
- **modular structure** ⇒ transparent modification of pipeline and exchange of services
  (e.g. Prob3++ ⇔ nuSQuIDS for oscillations)
- each service has **associated parameters**:
  defined in a pipeline config file

fitting procedure:
change parameter(s)
→ re-run pipeline
→ compare template



Events:

$[x_1, x_2, \ldots, x_n]$
$[y_1, y_2, \ldots, y_n]$
$[w_1, w_2, \ldots, w_n]$

- data (e.g. MC sample) represented by numba `SmartArrays`, passed on from each stage
- flexibly **transform between binned** (map) and **unbinned** (events) **data representations**

PISA: Not just a "fitter"

- apart from core modules and (high-level) routines for performing physics analyses, PISA provides lots of (lower-level) **utility modules** which make the user's life much easier, e.g.:

advanced configuration file parsing

---

comparison tools
⇒ hashing + caching at chosen floating point precision

---

consistent + reproducible random number generation

---

generic & clever file I/O

---

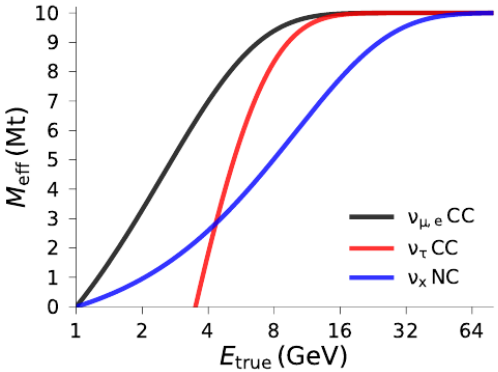
profiling & logging

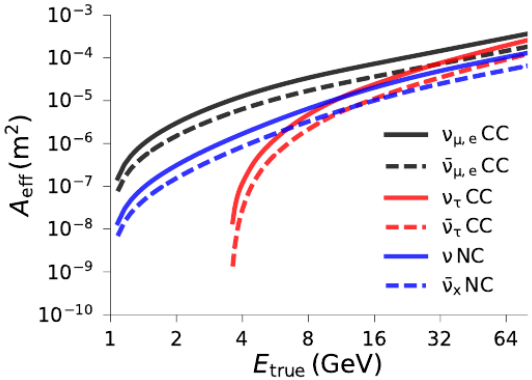
---

etc.

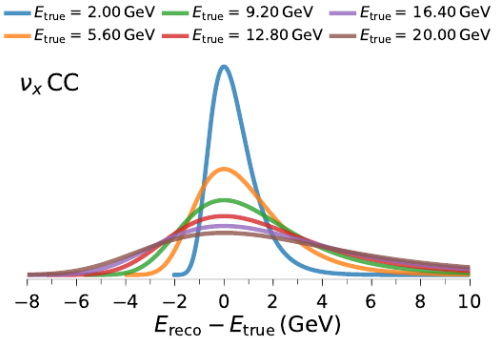
A toy NMO analysis

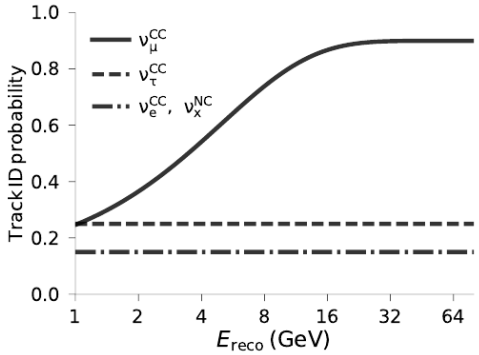
- employ **parametric toy detector model** to validate staged approach





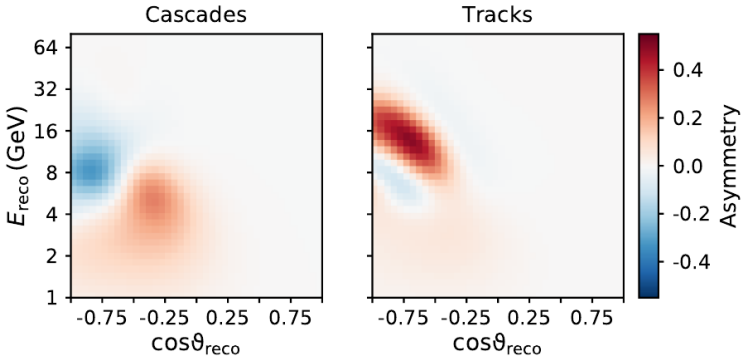
effective mass/area vs. true neutrino energy





energy resolution & event classification vs. true/reco'd neutrino energy

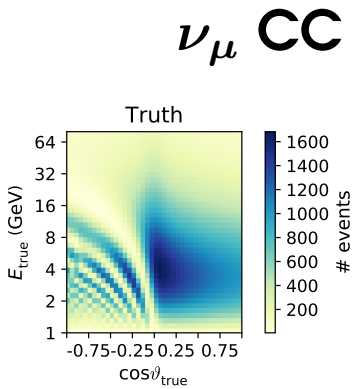
- obtain "typical" **NMO asymmetry** signatures in cascade- and track-like events (signed binwise $\sqrt{\chi^2}$)

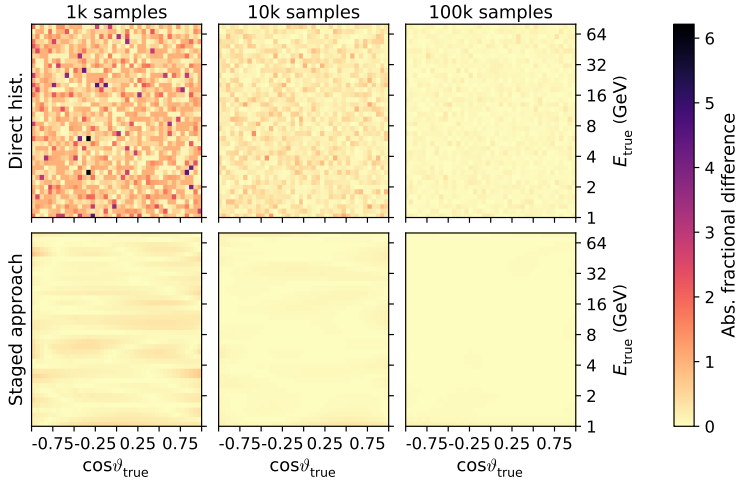


Validation of detection stage

- approach: **sample *N* MC events from unbinned toy distributions**
- **staged approach**:
  1. evaluate detector's effective areas on fine grid in true (energy, cosine zenith)
  2. apply Gaussian smearing along 2D grid
  3. apply cubic splines along energy and cosine zenith (sequentially)
  4. multiply by oscillated fluxes
- **direct histogramming**: directly bin event weights in true (energy, cosine zenith)
- compare to **parametric reference distribution ("truth")**

⇒ staged approach w/ smoothing shows good agreement even for very small sample size



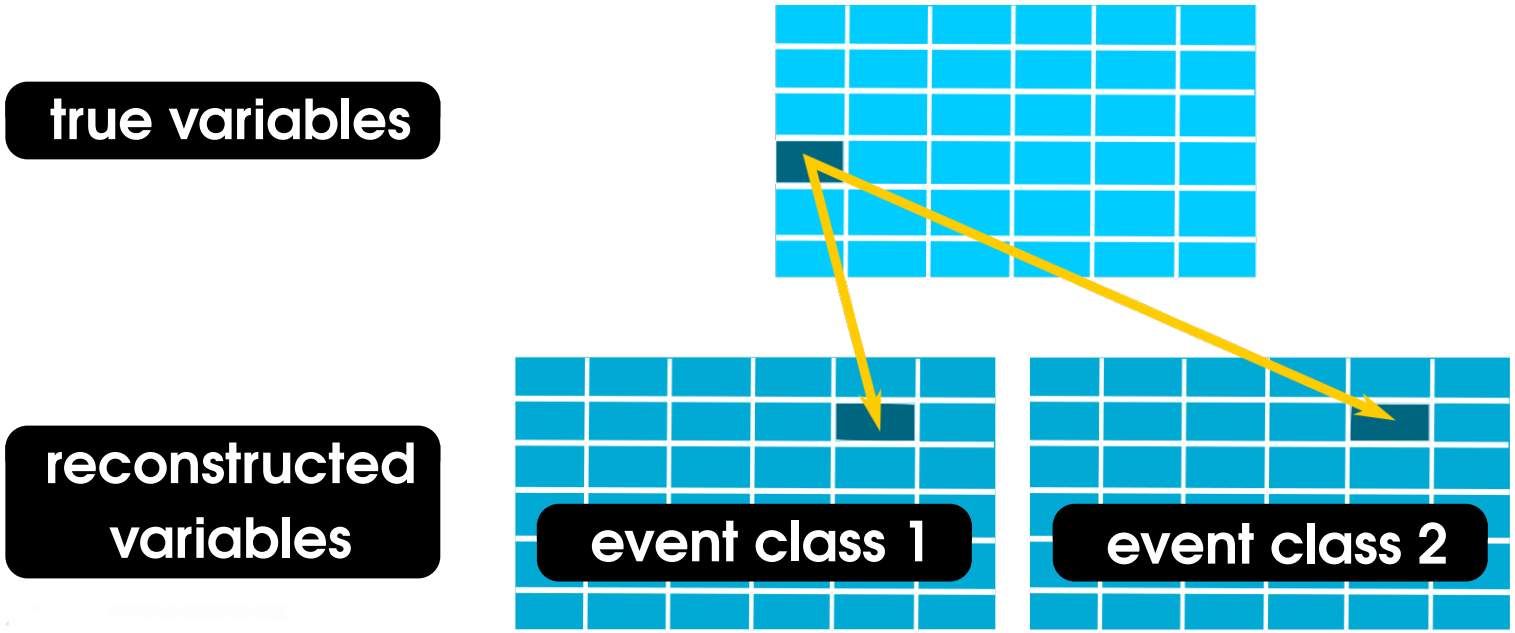


Characterising reconstruction resolutions

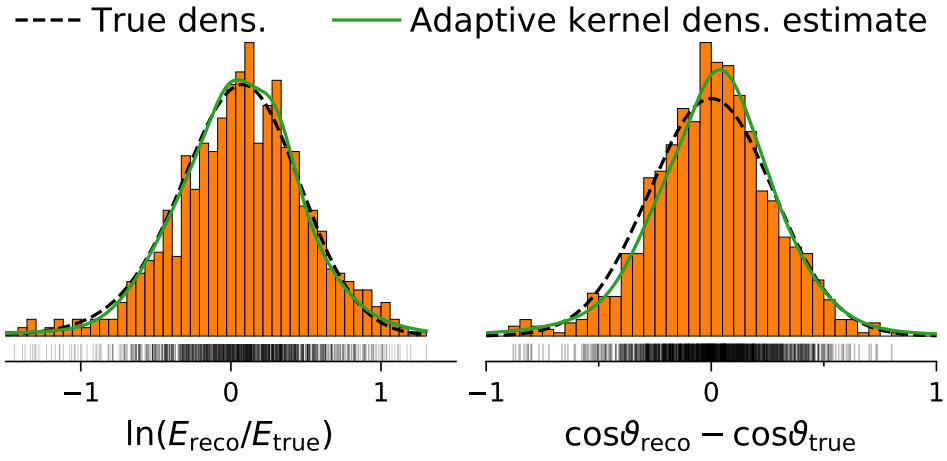
- **detector resolution functions** constructed from same MC events
- subsequent integration yields transformation:

$$(E_{\mathrm{true}}, \cos\vartheta_{\mathrm{true}}) \rightarrow (E_{\mathrm{reco}}, \cos\vartheta_{\mathrm{reco}}, \text{event classification})$$

- small MC amounts critical due to **high dimensionality** of transformation
- advantageous to characterise quantities with **reduced dependency on true variables**



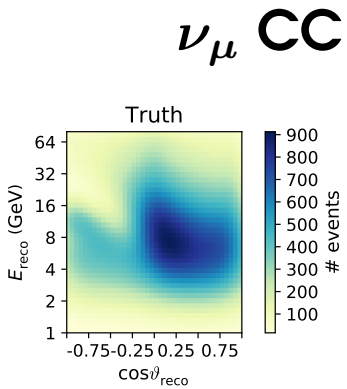
Validation of final-level templates

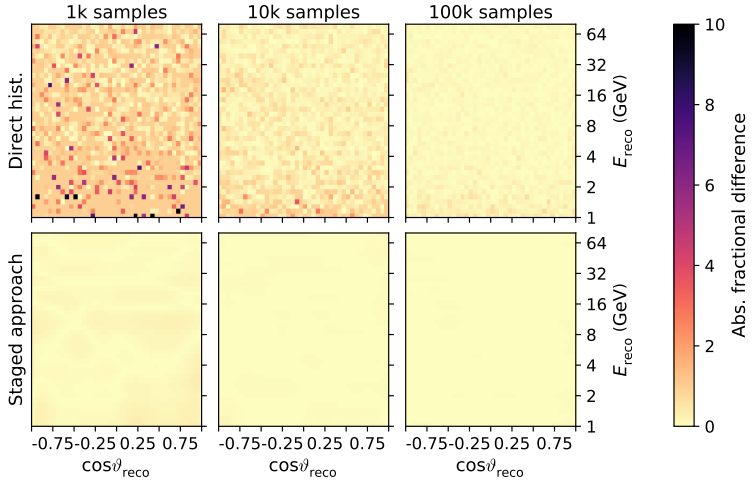


- generate **single 1d resolution function per input-output coordinate**
- found **adaptive (variable bandwidth) KDE** to outperform other smoothing methods

- ideally finely subdivide dependent dimensions (here: $E_{\mathrm{true}}, \cos\vartheta_{\mathrm{true}}$), but **trade-off with MC statistics**

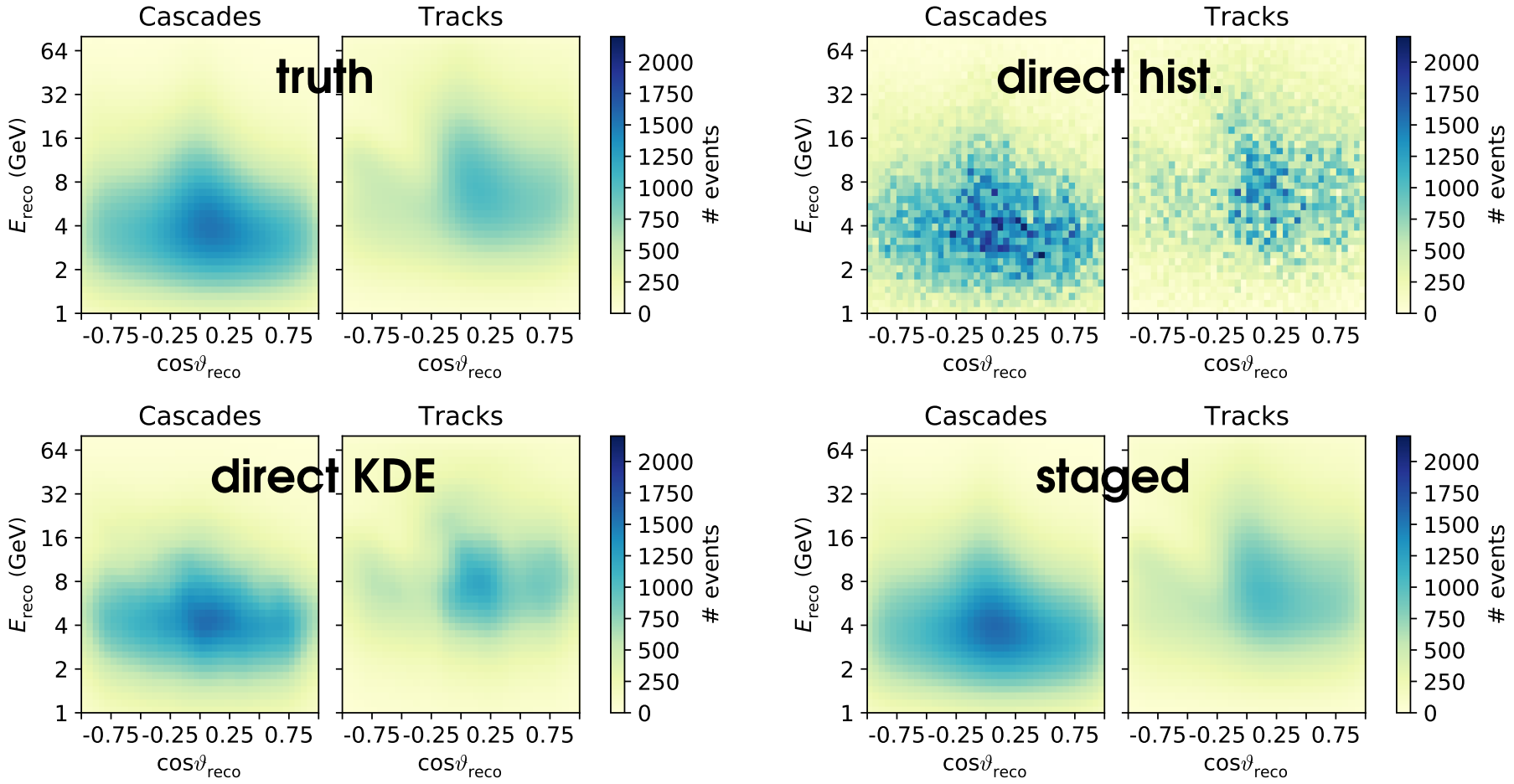
⇒ templates from staged approach w/ smoothing considerably more accurate than from direct hist.





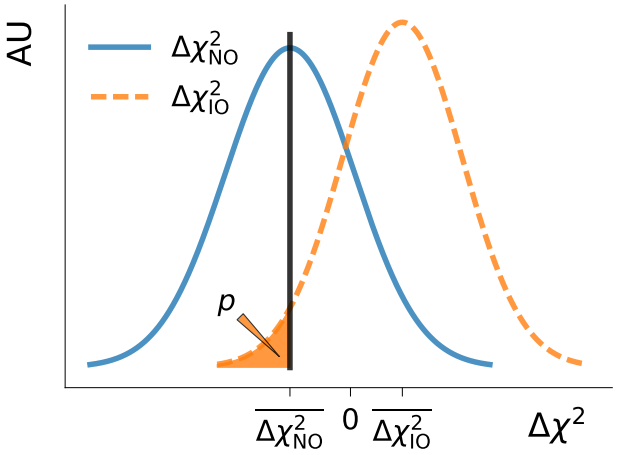
Final-level template comparison

- **final-level templates** of the three different methods and **truth**, for one MC event sample of size 10^4



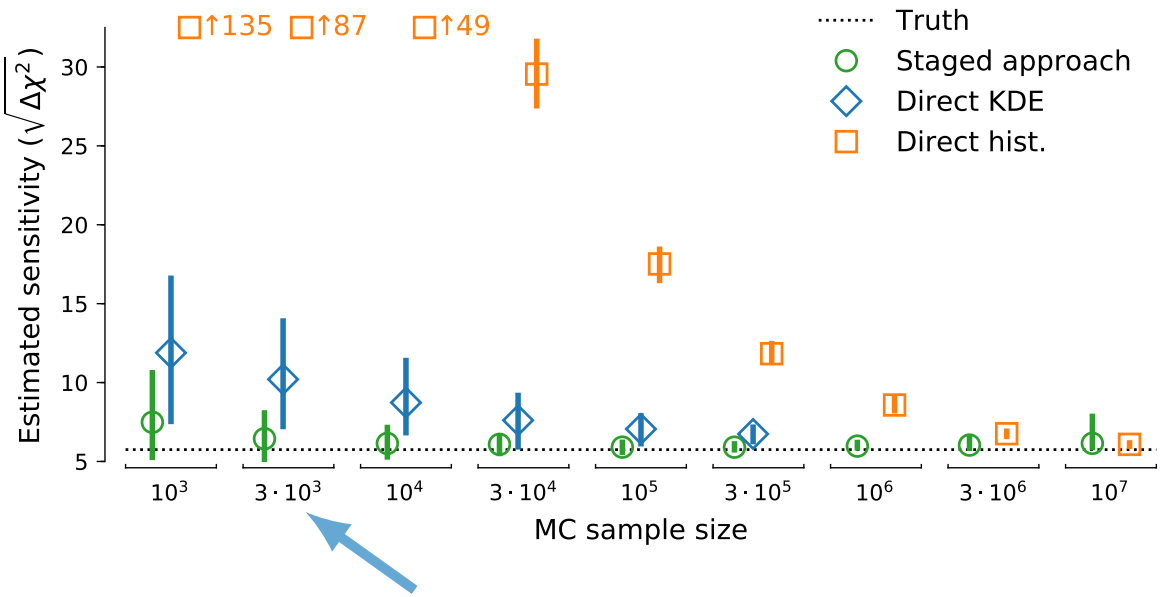
⇒ **only the staged approach passes the "eye test"**

Results of toy NMO analysis



- perform fit of IO template to NO Asimov template and record $\sqrt{\chi^2}$ **as sensitivity proxy**
- **compare (distributions of) predictions of the three methods to true significance**
- **repeat for different MC sample sizes**

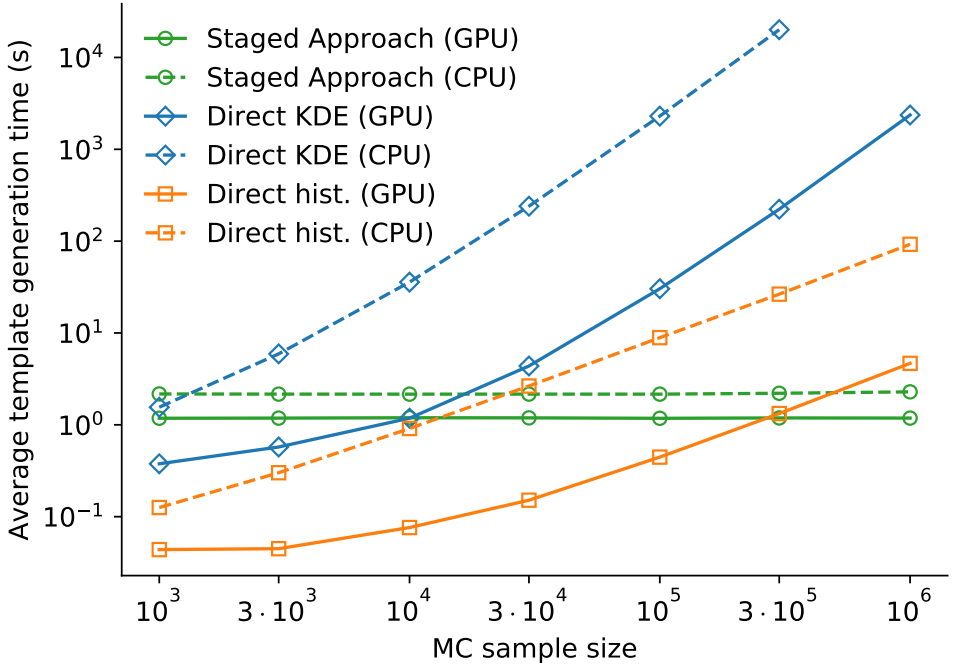
- require $\sim 10^7$ MC events for **direct hist.**
- some improvement from **direct KDE**, but too slow for larger sample sizes
- **staged approach**: amount of MC needed is **reduced by orders of magnitude**



less than one MC event per final-level template bin

Timing benchmarks

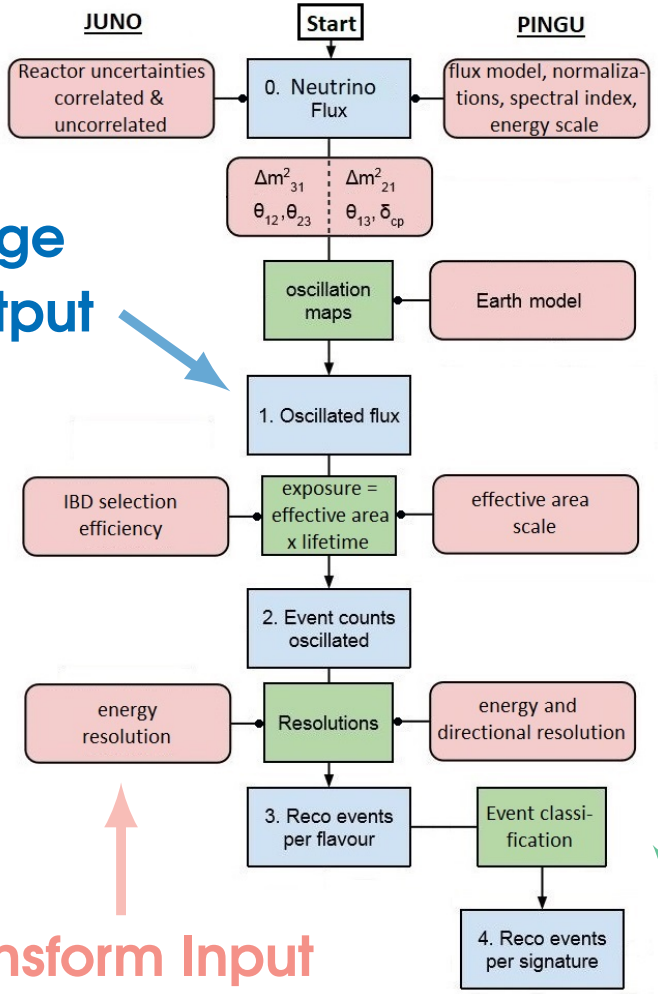
- **usefulness** of given analysis method also **crucially dependent on duration of computation** (here: template generation)



- **staged approach only dependent on MC sample size for start-up**
  - **direct hist.** fast but biased for small sizes
  - **direct KDE** impractical to use for large sizes

PISA NMO Analysis with JUNO + PINGU

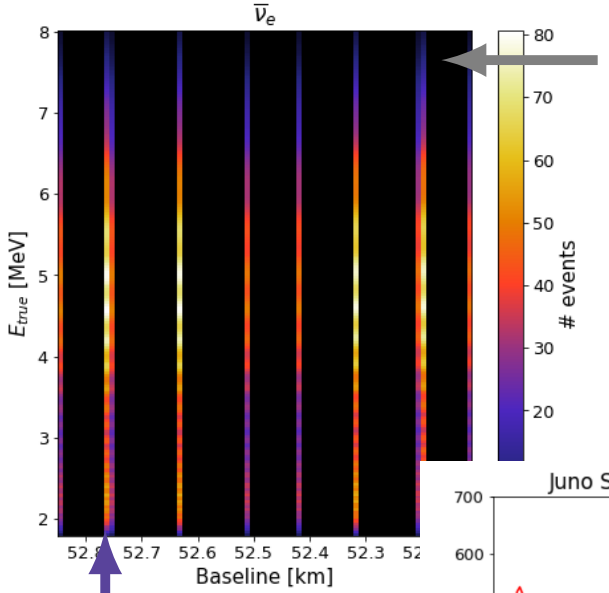
(to be published)



Stage Output

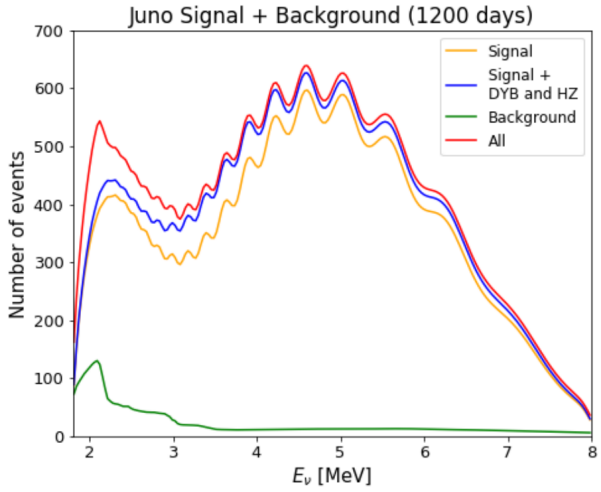
Transform Input

Transform



signal-less cosine-zenith interval

single-reactor energy spectrum



two separate pipelines!

⇒ tools for combining different types of experiments, with joint & separate systematics

Summary

- **`PISA` software**:
  - from a map/histogram-based simulation/fitting tool tailored to PINGU NMO to a **general-purpose modular physics analysis framework**
  - **easily extendable staged approach with efficient smoothing methods** in place: mitigate low MC statistics/increase effective amount of MC
  - **template generation time independent of MC sample size** (excluding start-up costs)
- technical paper submitted to J. Comp. Phys.; preprint available at **arXiv:1803.05390**
- code maintained by IceCube collaboration, not yet open-sourced... stay tuned


Thank you for your attention!
© C. Givaga / Fotolia

BACKUP

Grid points in staged approach

- grid choice for stage transformations and stage outputs in toy NMO analysis:

Stage	Transformation	Output
Flux	-	$400\ E_{\text{true}} \times 400\ \cos\vartheta_{\text{true}}$
Oscillation	400×400	$400\ E_{\text{true}} \times 400\ \cos\vartheta_{\text{true}}$
Detection	400×400	$200\ E_{\text{true}} \times 200\ \cos\vartheta_{\text{true}}$
Reconstruction	$200 \times 200 \times 40 \times 40 \times 2$	$40\ E_{\text{reco}} \times 40\ \cos\vartheta_{\text{reco}} \times 2$ classes

- staged approach w/o smoothing shown to converge to output of direct histogramming (in asymptotic limit)

ν channel evolution across stages