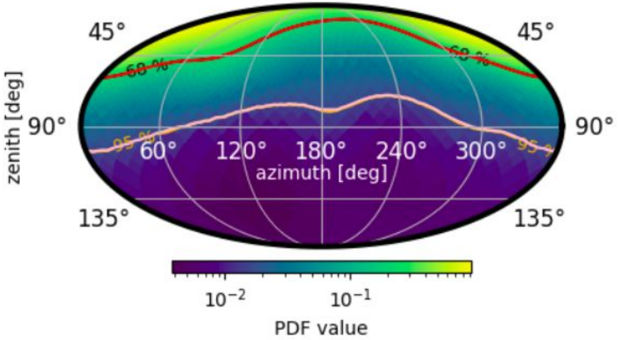
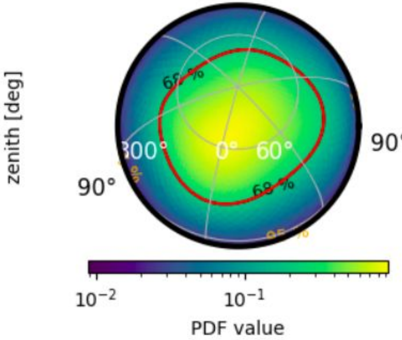
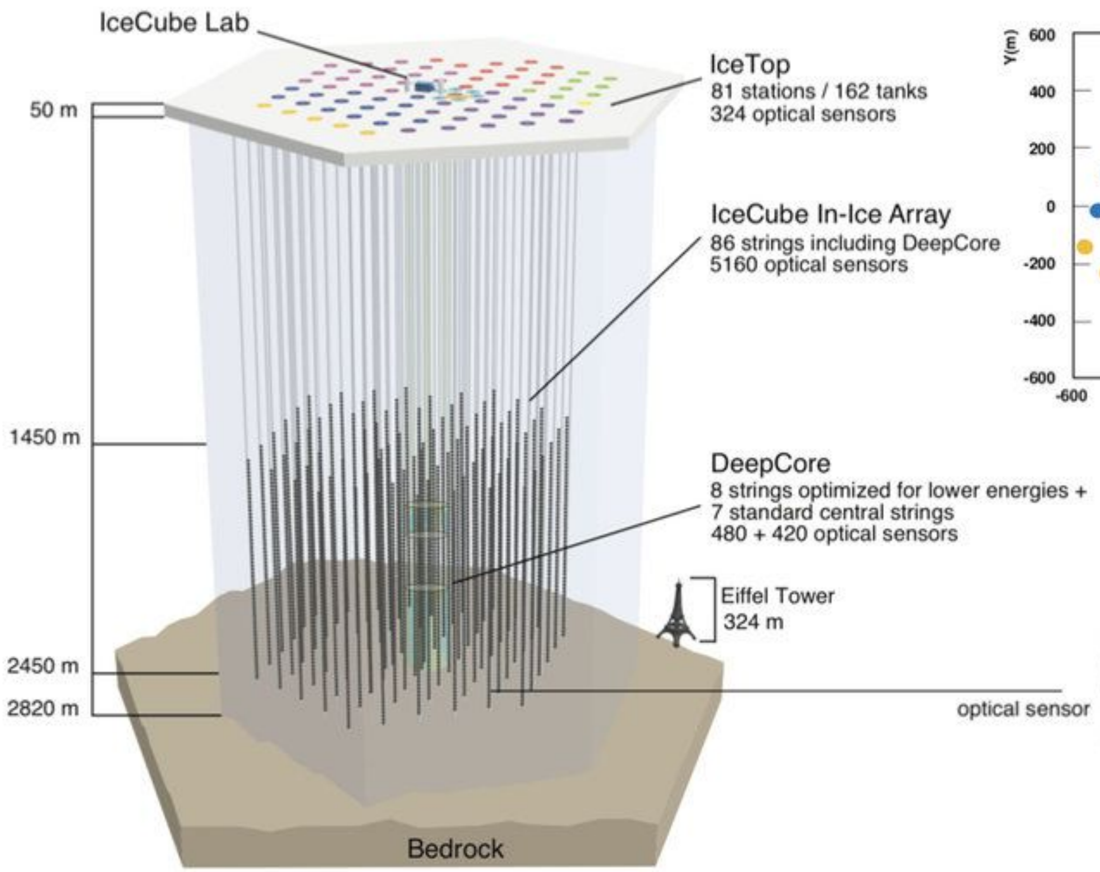
Transformer-encoded normalizing flows for IceCube reconstruction

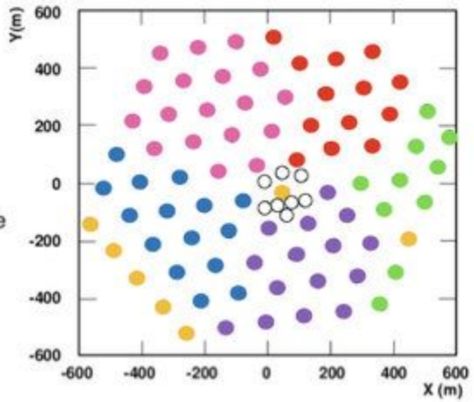




HAMLET-physics meeting, Aug 20th, 2026, Thorsten Glüsenkamp

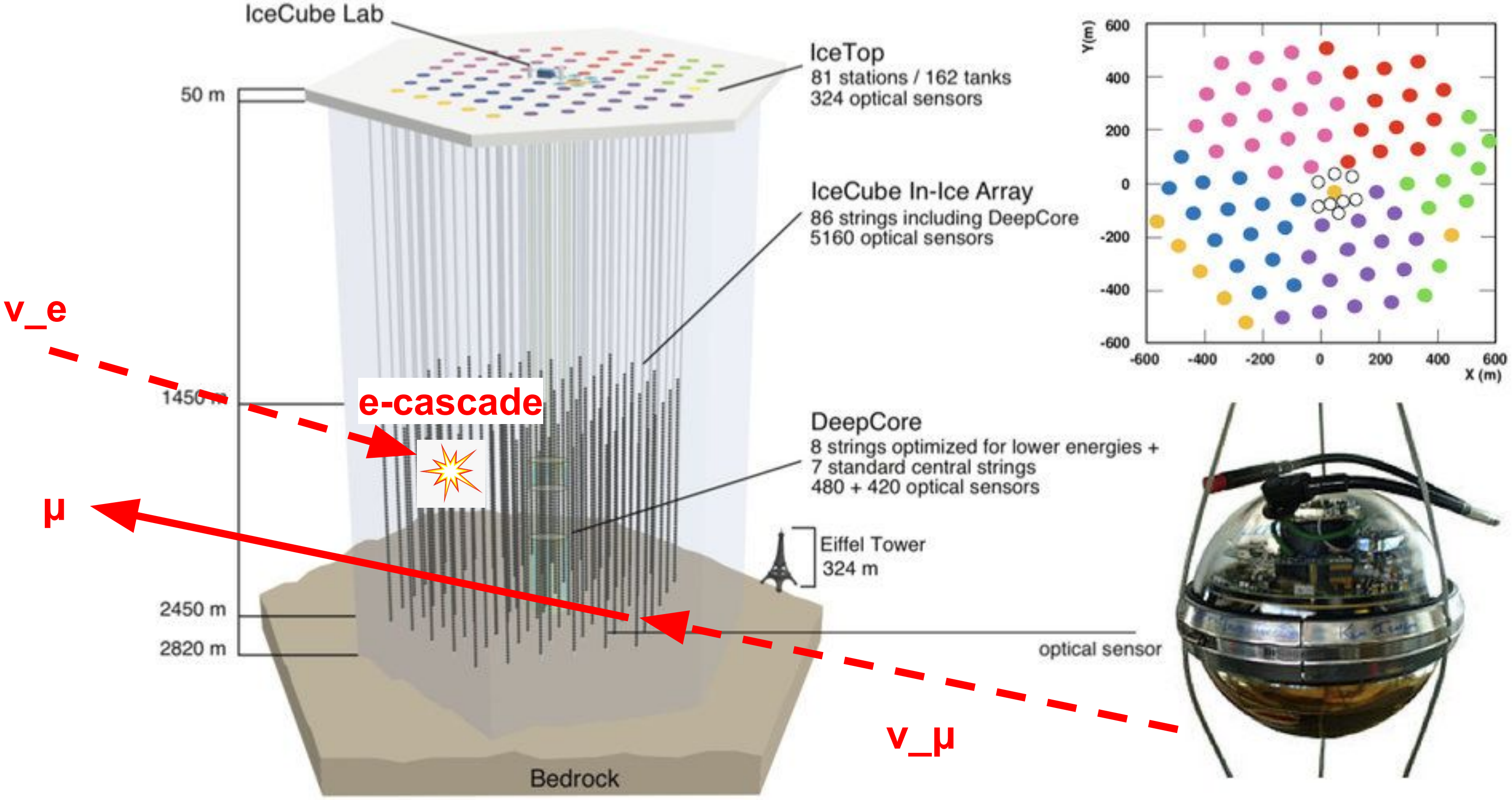
IceCube Detector





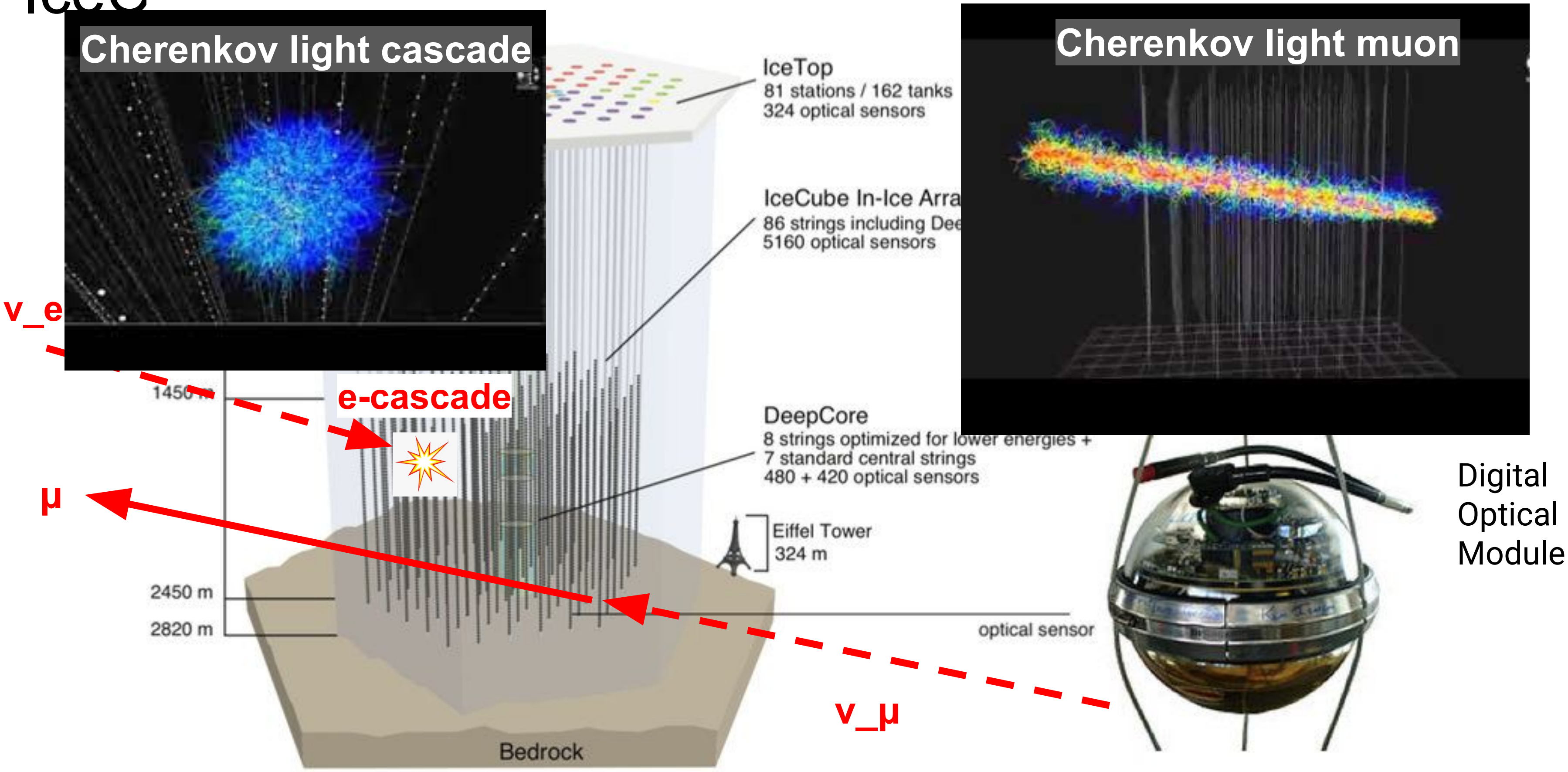
Digital Optical Module

IceCube Detector

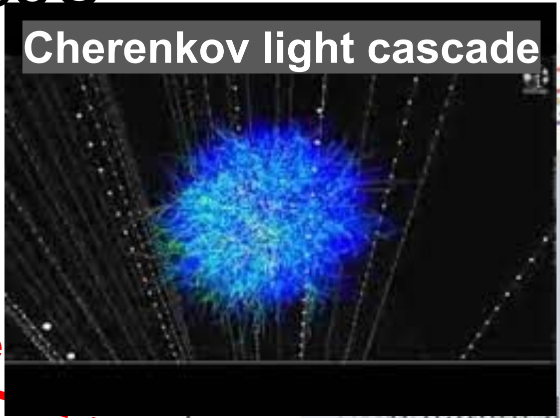


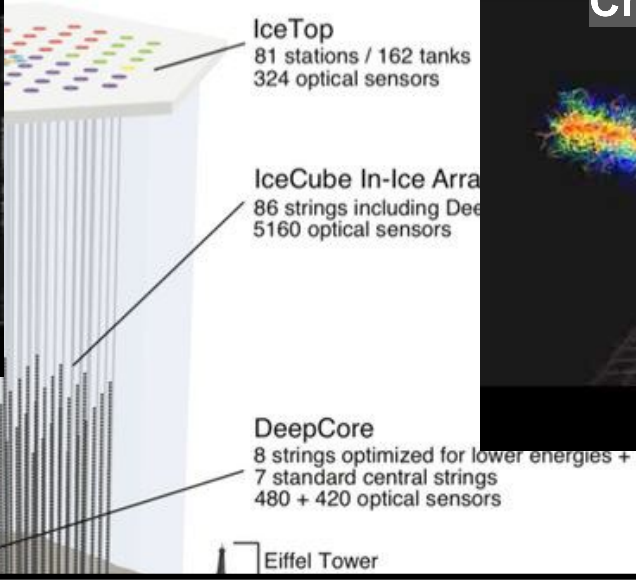
Digital Optical Module

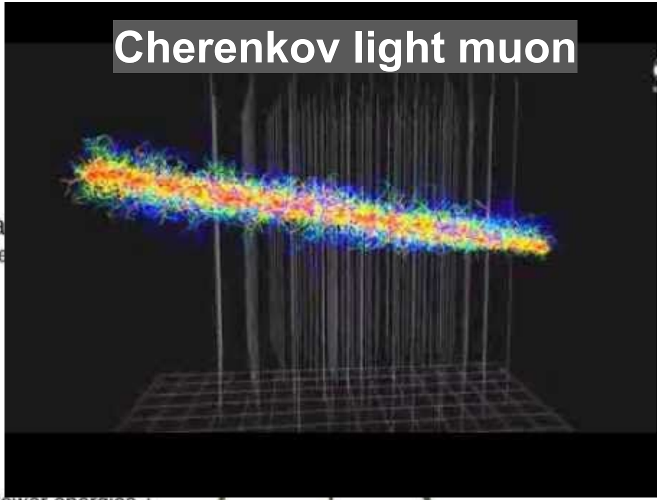
IceCube Detector



IceCube Detector



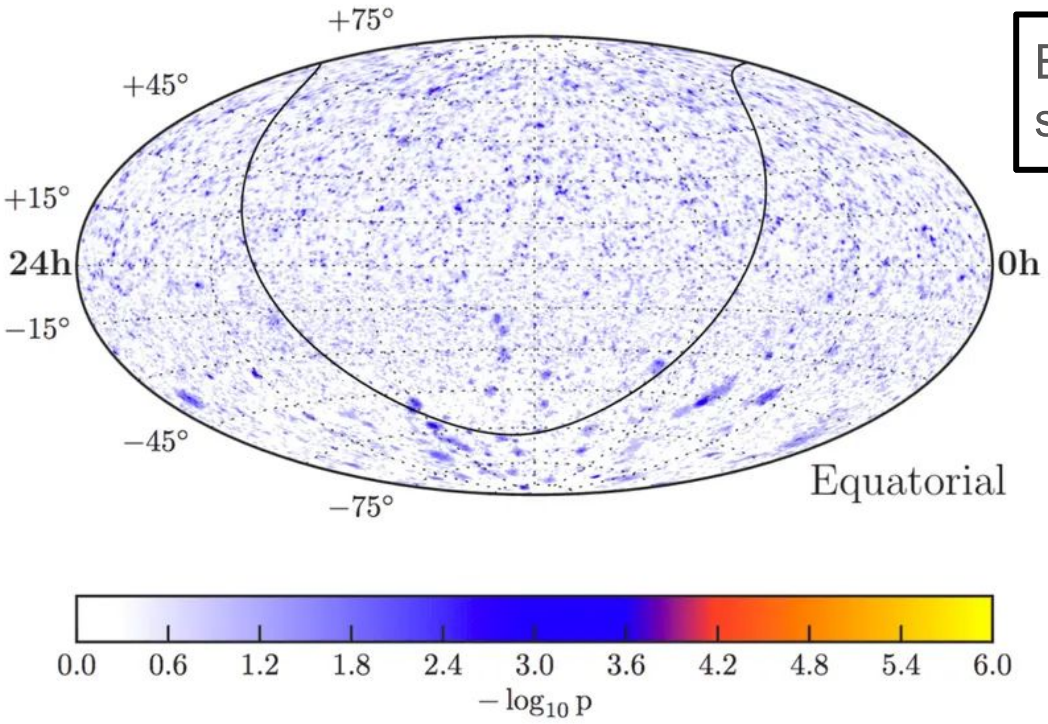




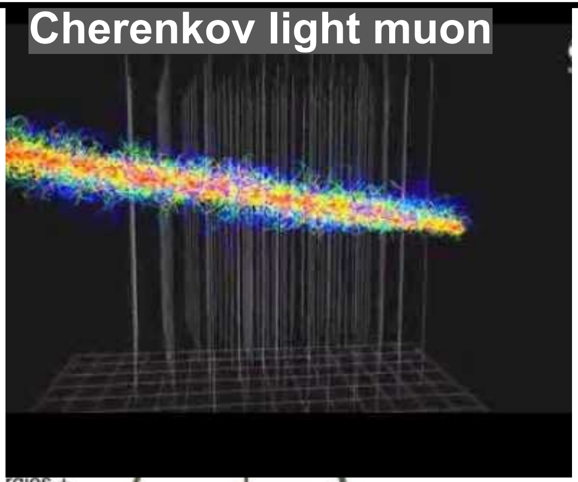
Digital Optical Module

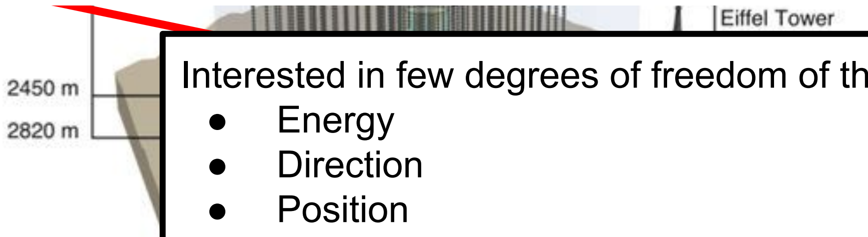
Interested in few degrees of freedom of these events:

- Energy
- Direction
- Position
- …



E.g. direction for point source searches



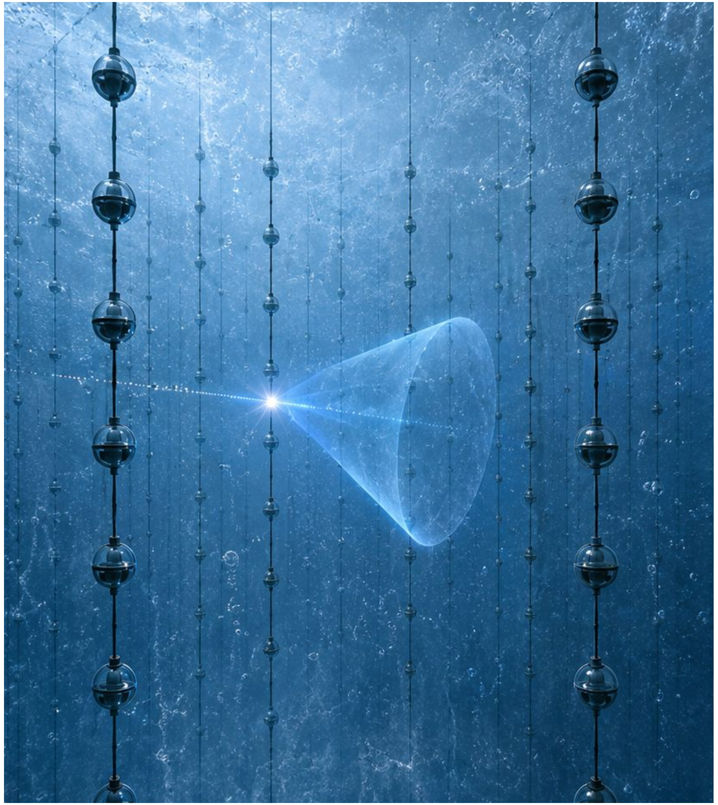


Interested in few degrees of freedom of these events:

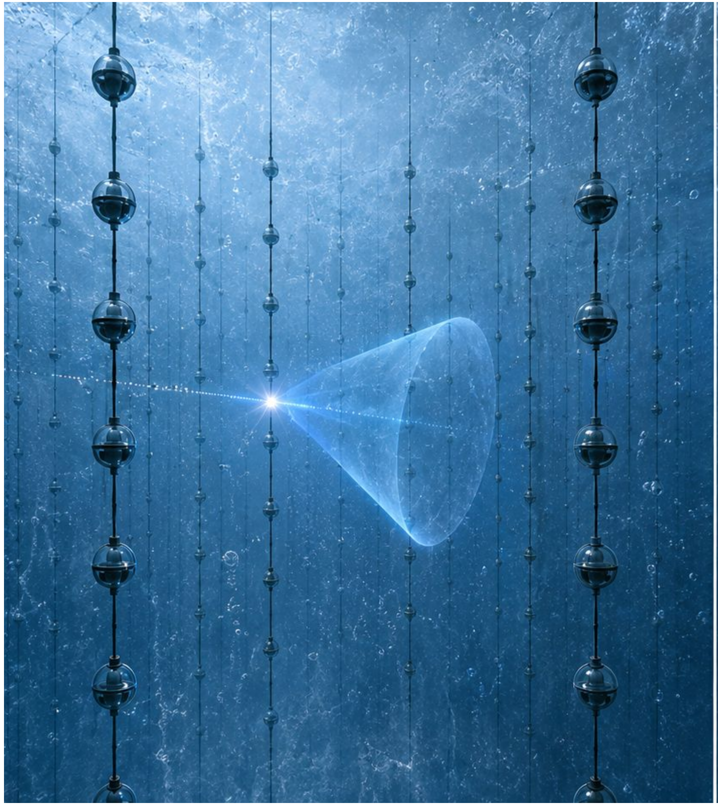
- Energy
- Direction
- Position
- …

Digital Optical Module

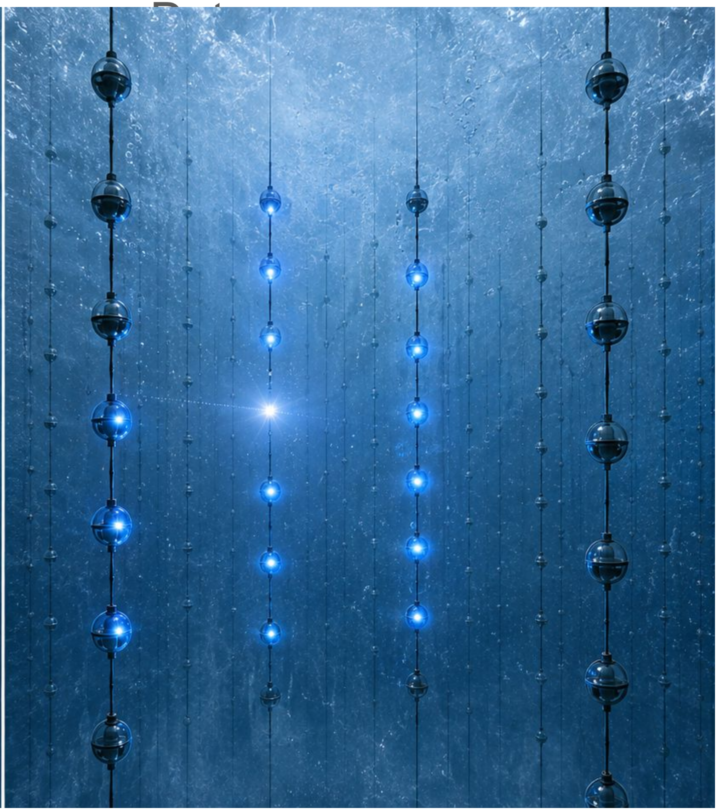
Interaction happens…



Interaction happens…



A few nanoseconds later..



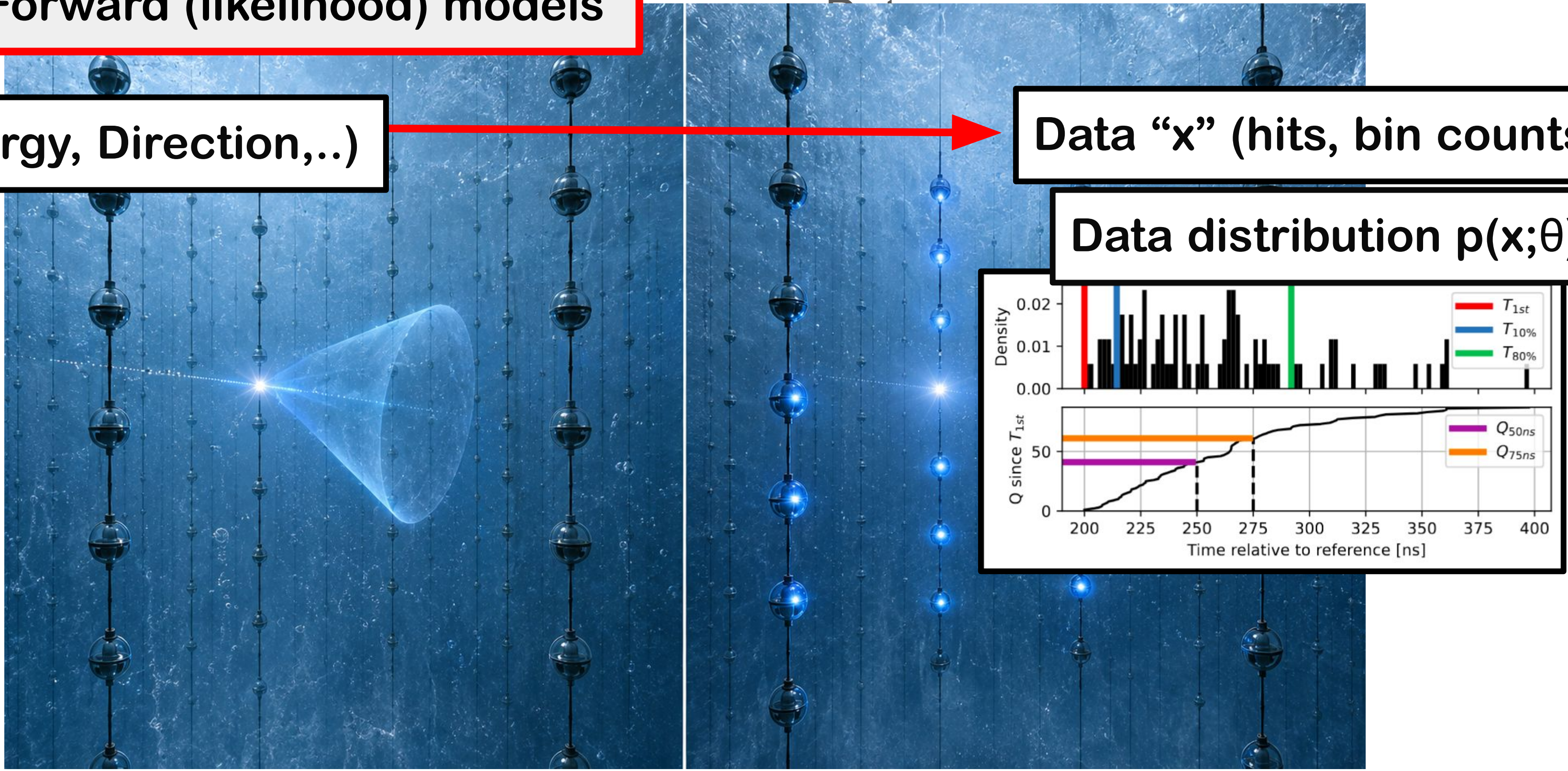
Interaction happens…

A few nanoseconds later..

Forward (likelihood) models

θ={Energy, Direction,..) → **Data “x” (hits, bin counts)**

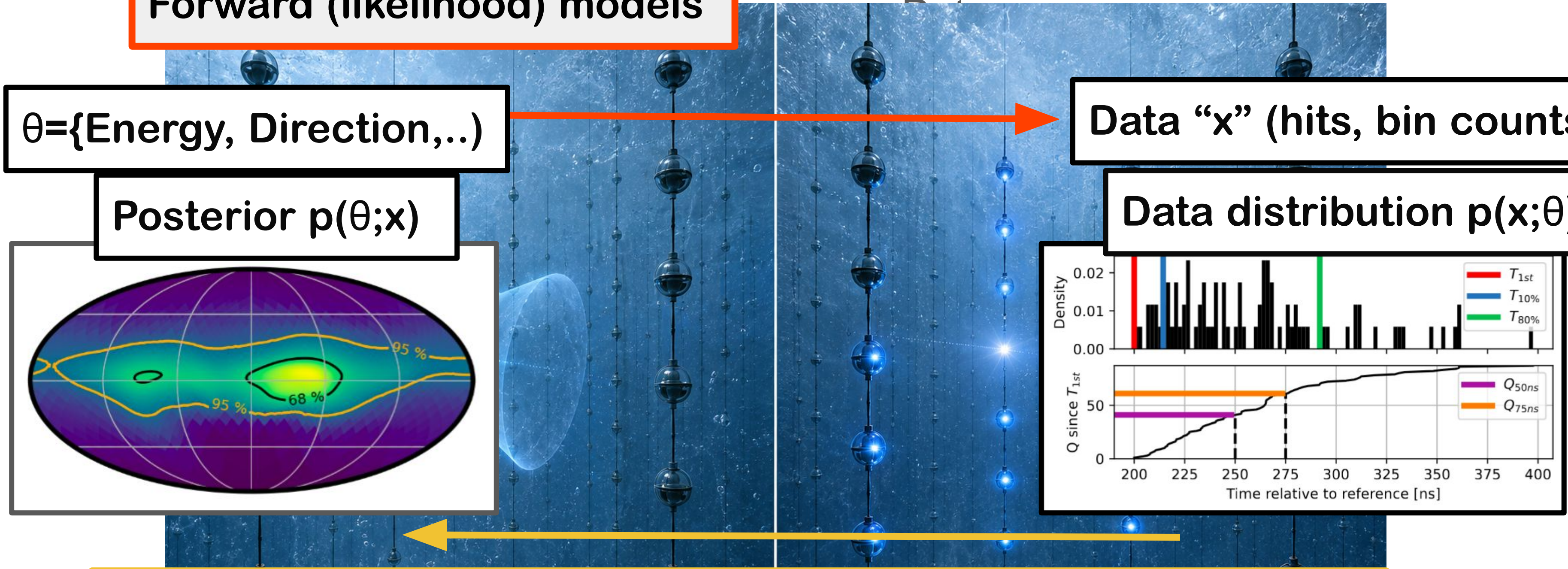
Data distribution p(x;θ)



Interaction happens…

A few nanoseconds later..

Forward (likelihood) models



θ={Energy, Direction,..)

Data “x” (hits, bin counts)

Posterior p(θ;x)

Data distribution p(x;θ)

Inverse (likelihood-free) modelling

(amortized neural posterior estimation, NN-Regression / Classification, …)

Interaction happens…

A few nanoseconds later..

Forward (likelihood) models

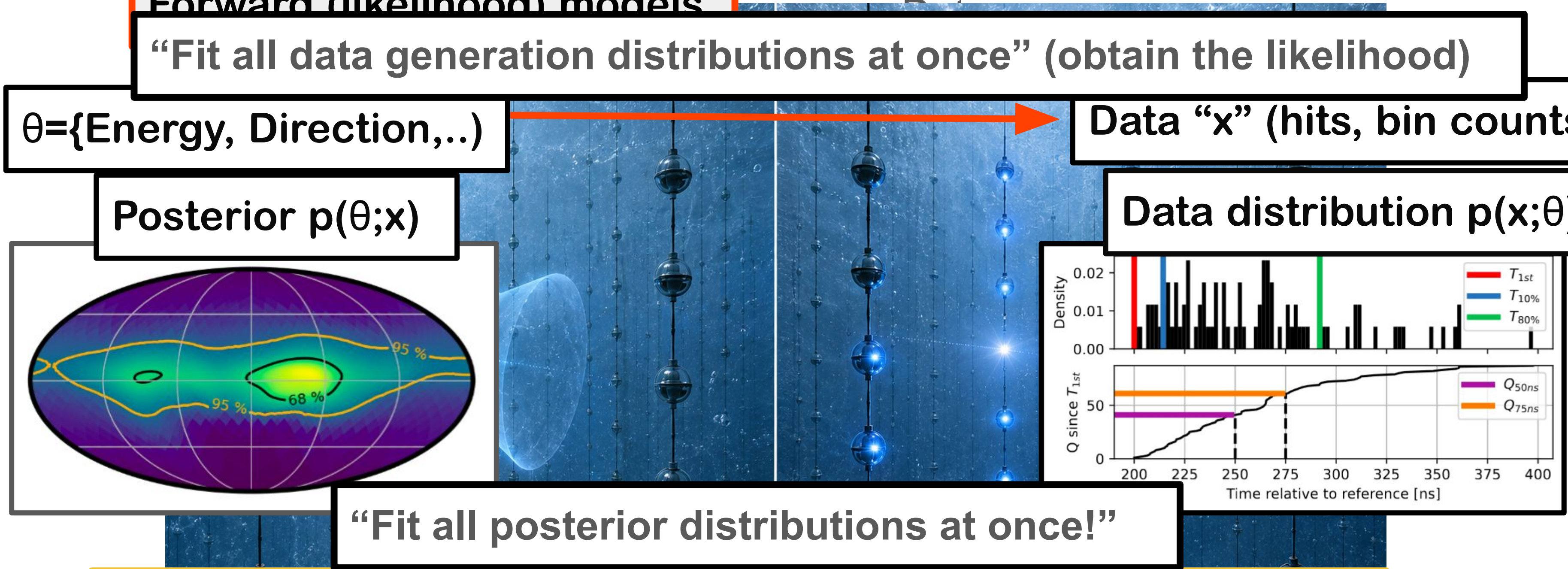
"Fit all data generation distributions at once" (obtain the likelihood)

θ={Energy, Direction,..)

Data "x" (hits, bin counts)

Posterior p(θ;x)

Data distribution p(x;θ)



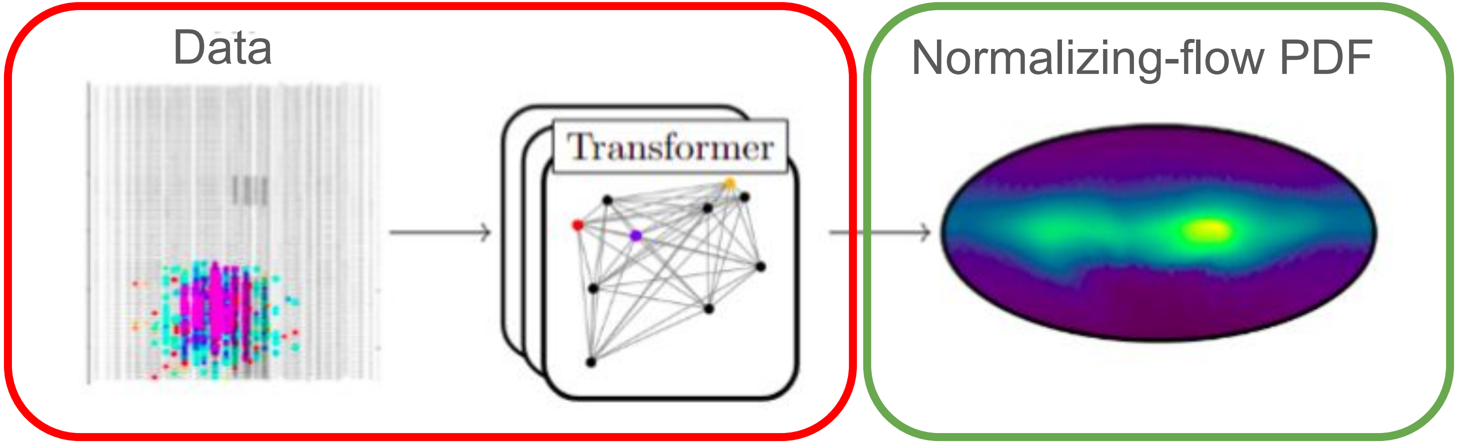
"Fit all posterior distributions at once!"

Inverse (likelihood-free) modelling

(amortized neural posterior estimation, NN-Regression / Classification, …)

The model: a **posterior** PDF prediction machine

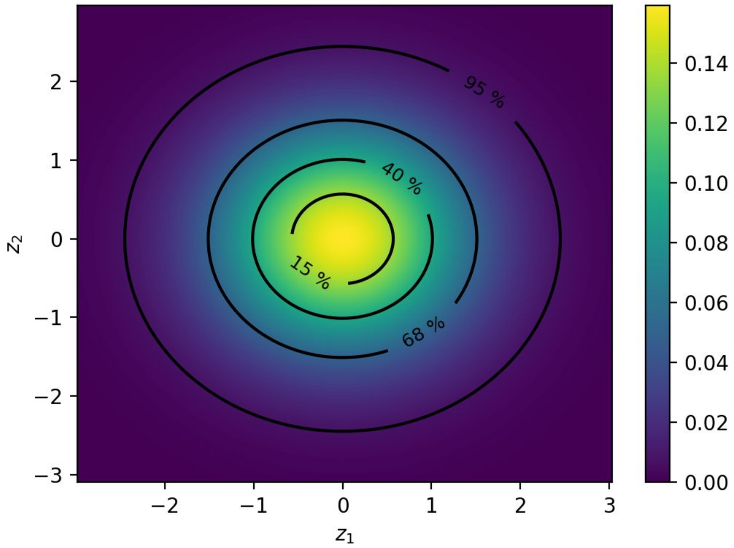
STEP 1



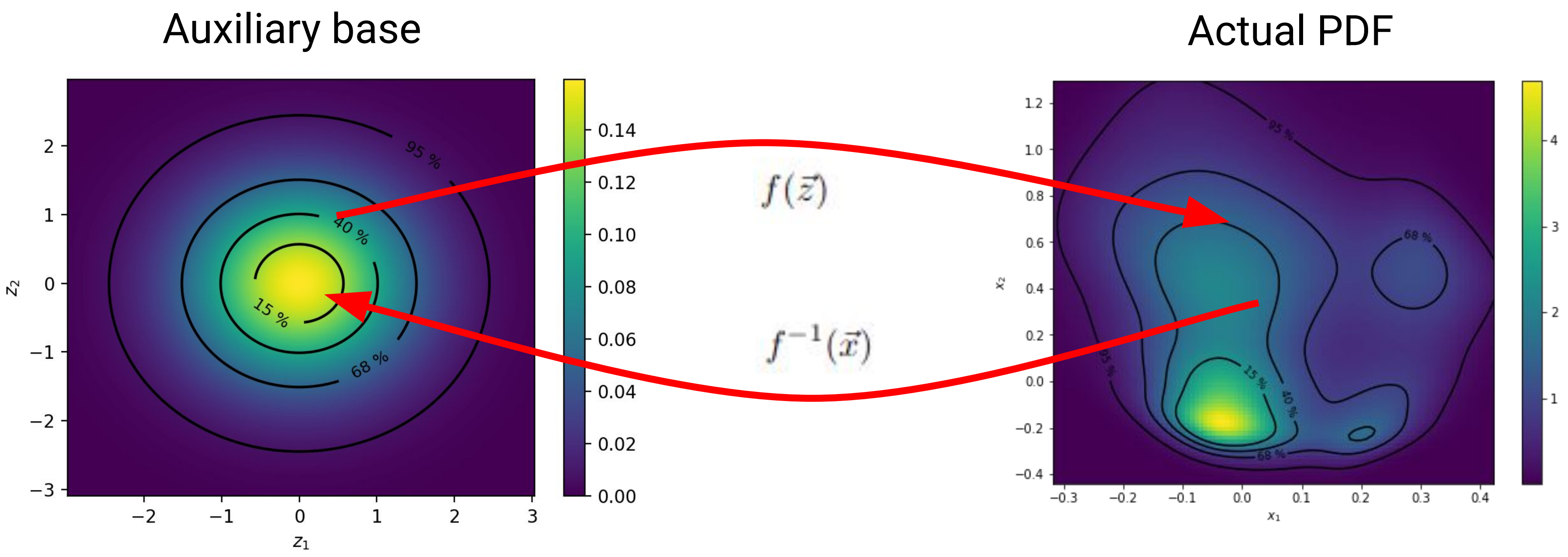
STEP 2

Normalizing flows: flexible PDFs

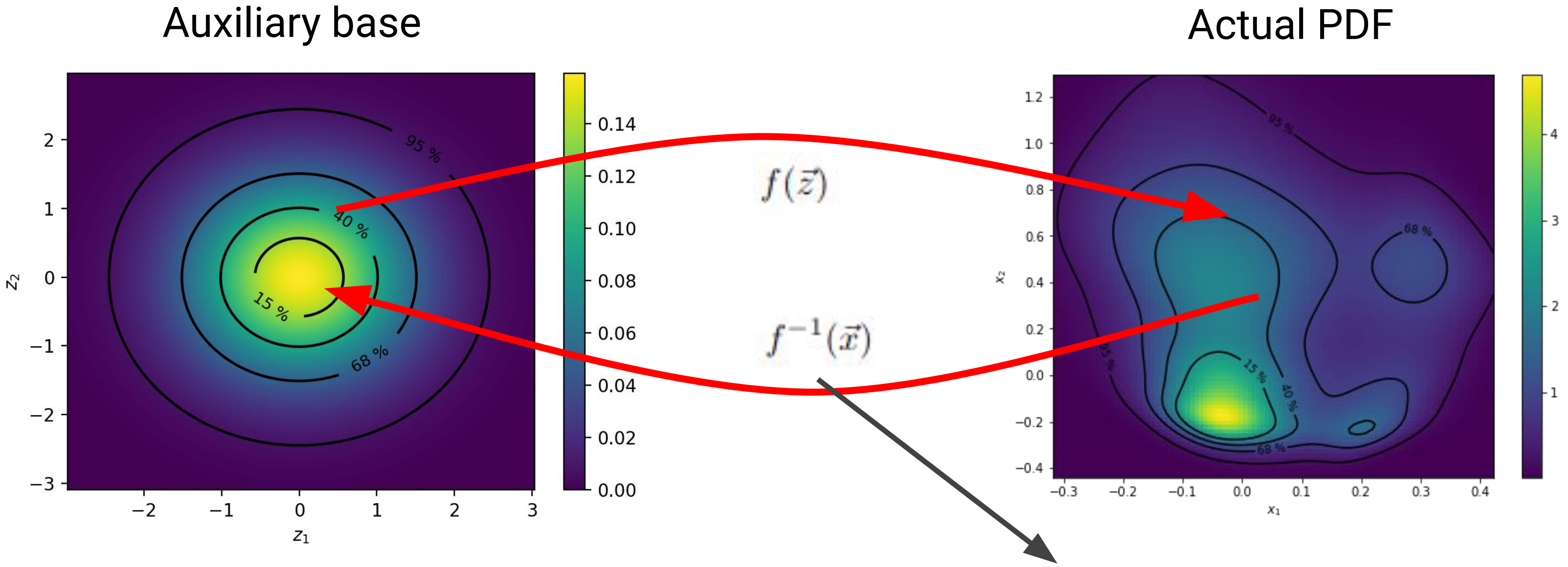
Auxiliary base



Normalizing flows: flexible PDFs



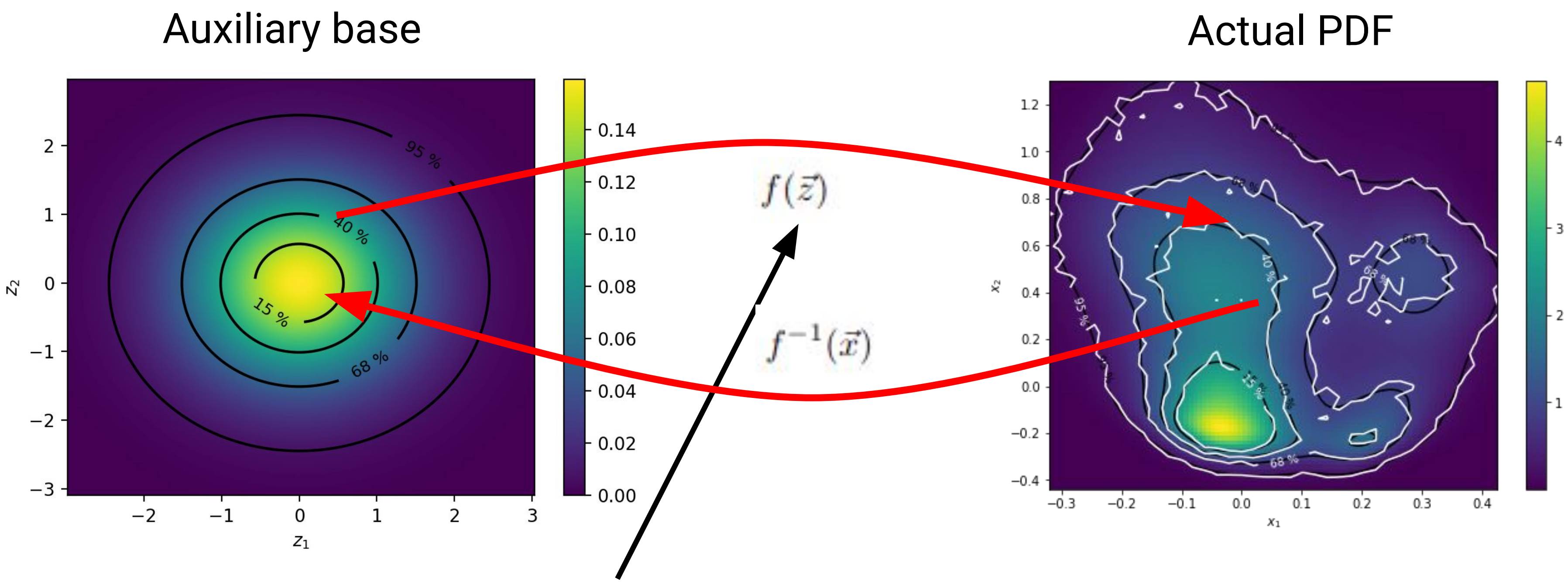
Normalizing flows: flexible PDFs



exact log-probability using inverse

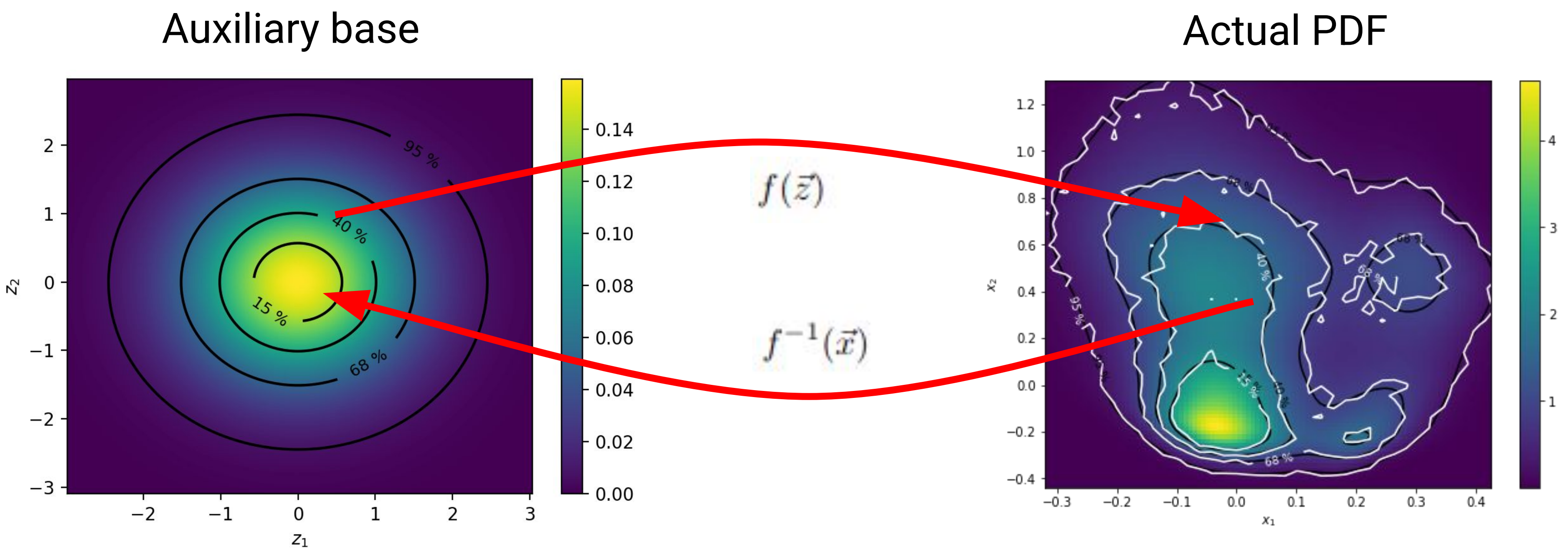
$$p(\vec{x}) = p_0(f^{-1}(\vec{x})) \cdot \left|\det \frac{\partial f^{-1}(\vec{x})}{\partial \vec{x}}\right|$$

Normalizing flows: flexible PDFs



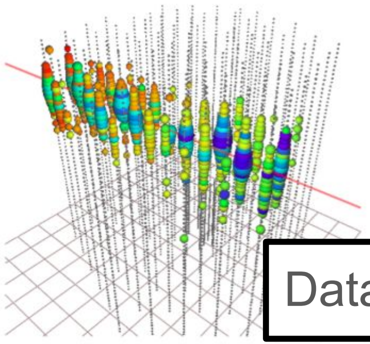
efficient sampling via forward f(z)

Normalizing flows: flexible PDFs



PDF with **very efficient sampling + analytic log-probability**
... no deep learning yet!

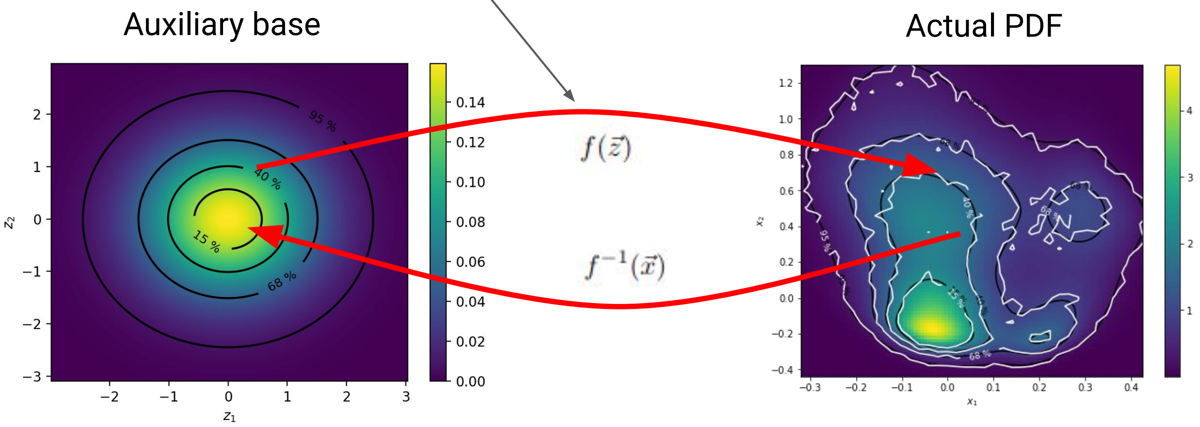
Conditioning via NN -> conditional PDF



f(z) depends on parameters “u”: $f_u(z)$

Predict u / diffeomorphism via “X”

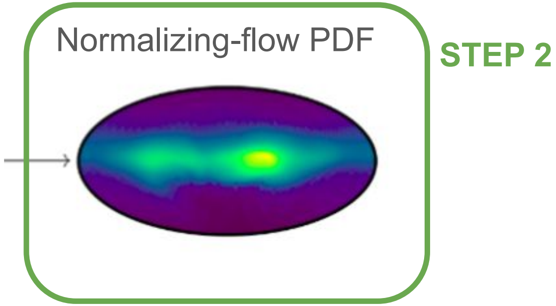
Data “X” → **u = NNθ(X)**



The PDF - conditional normalizing flow on sphere

The specific normalizing-flow recipe:

Final PDF: $$\boxed{p_u(\theta) = p_0(f_u^{-1}(\theta)) \cdot |\det J_u^{-1}(\theta)|}$$



The PDF - conditional normalizing flow on sphere

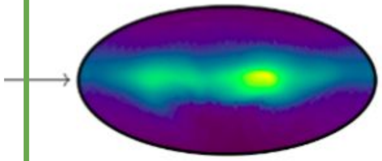
The specific normalizing-flow recipe:

Final PDF: $$\boxed{p_u(\theta) = p_0(f_u^{-1}(\theta)) \cdot |\det J_u^{-1}(\theta)|}$$

modified Jacobian for manifold

Normalizing-flow PDF

STEP 2



The PDF - conditional normalizing flow on sphere

The specific normalizing-flow recipe:

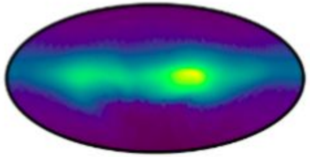
modified Jacobian for manifold

Final PDF: $\boxed{p_u(\theta) = p_0(f_u^{-1}(\theta)) \cdot |\mathrm{det} J_u^{-1}(\theta)|}$

Building blocks of PDF are 15 invertible functic f_{ϕ_i}

$$f_u(\theta) = [f_{\phi_{15}} \circ \ldots \circ f_{\phi_1}](\theta)$$

Normalizing-flow PDF



STEP 2

The PDF - conditional normalizing flow on sphere

The specific normalizing-flow recipe:

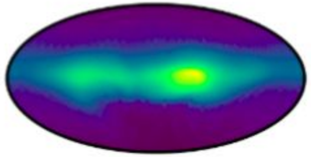
modified Jacobian for manifold

Final PDF: $$\boxed{p_u(\theta) = p_0(f_u^{-1}(\theta)) \cdot |\mathrm{det} J_u^{-1}(\theta)|}$$

Building blocks of PDF are 15 invertible functic f_{ϕ_i}

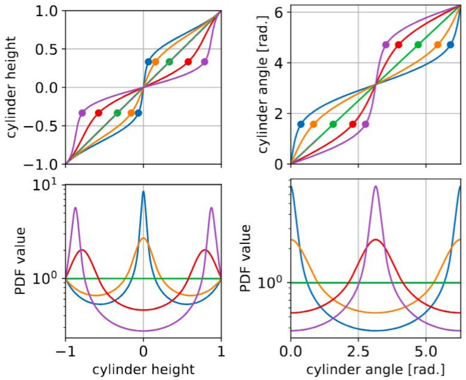
Normalizing-flow PDF

STEP 2



$$f_u(\theta) = [f_{\phi_{15}} \circ \ldots \circ f_{\phi_1}](\theta)$$

Each function block f_{ϕ_i}



1) “Smooth RQ Spline”

The PDF - conditional normalizing flow on sphere

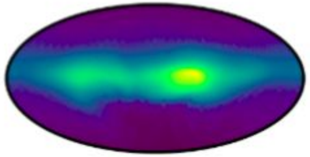
The specific normalizing-flow recipe:

modified Jacobian for manifold

Final PDF: $$p_u(\theta) = p_0(f_u^{-1}(\theta)) \cdot |\det J_u^{-1}(\theta)|$$

Normalizing-flow PDF

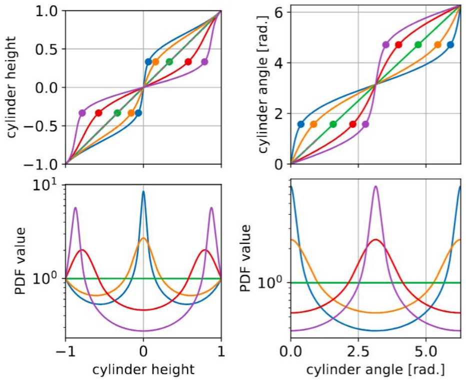
STEP 2



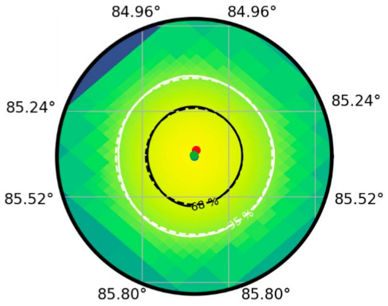
Building blocks of PDF are 15 invertible functic f_{ϕ_i}

$$f_u(\theta) = [f_{\phi_{15}} \circ \ldots \circ f_{\phi_1}](\theta)$$

Each function block f_{ϕ_i}



“Fisher zoom”



$$F(z) = 1 + (1/\kappa) \cdot \ln\left(\frac{1+z}{2} + \left(1 - \frac{1+z}{2}\right) \cdot e^{-2\kappa}\right)$$

1) “Smooth RQ Spline”

2) von-Mises Fisher

The PDF - conditional normalizing flow on sphere

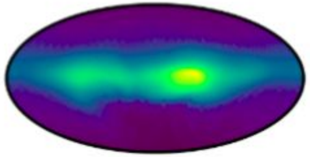
The specific normalizing-flow recipe:

modified Jacobian for manifold

Final PDF: $p_u(\theta) = p_0(f_u^{-1}(\theta)) \cdot |\mathrm{det} J_u^{-1}(\theta)|$

Normalizing-flow PDF

STEP 2

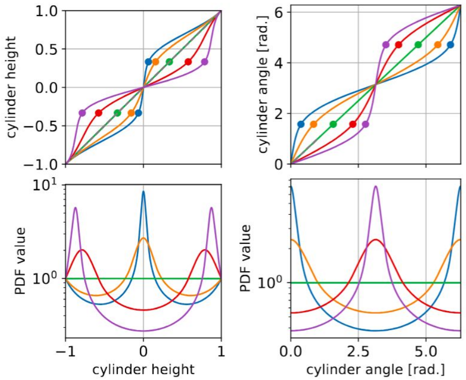


Building blocks of PDF are 15 invertible functic f_{ϕ_i}

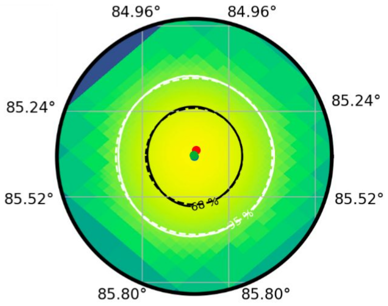
$$f_u(\theta) = [f_{\phi_{15}} \circ \ldots \circ f_{\phi_1}](\theta)$$

Each function block f_{ϕ_i}

~ 220 params



"Fisher zoom"



$$\mathrm{F}(z) = 1 + (1/\kappa) \cdot \ln\left(\frac{1+z}{2} + \left(1 - \frac{1+z}{2}\right) \cdot e^{-2\kappa}\right)$$

$$R = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

- "Mechanistic" way of PDF creation
- 15 blocks seem sufficient for all PDFs in IceCube
- von-Mises Fisher is special case

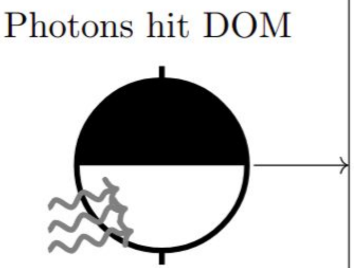
1) "Smooth RQ Spline"

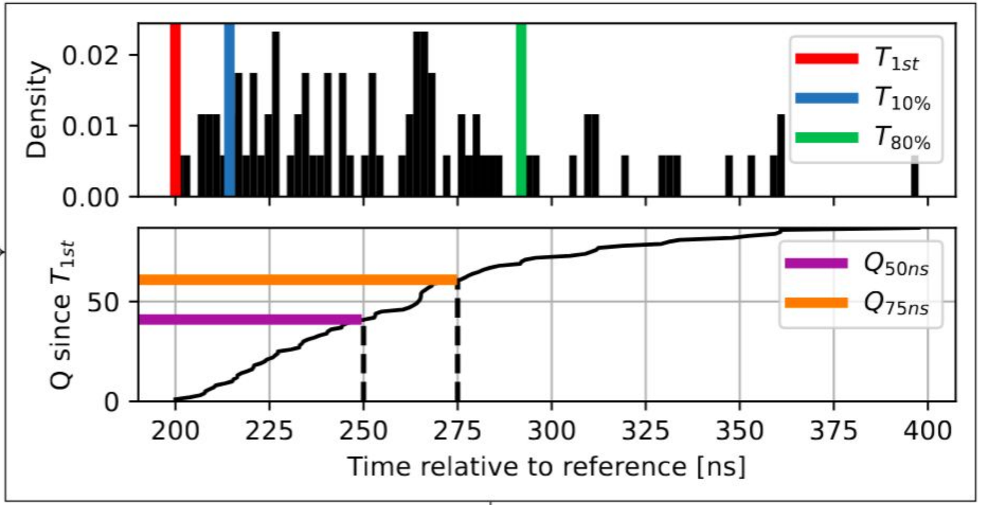
2) von-Mises Fisher

3) Rotation

Encoding

Step 1 - encoding the data





Can include all dom types

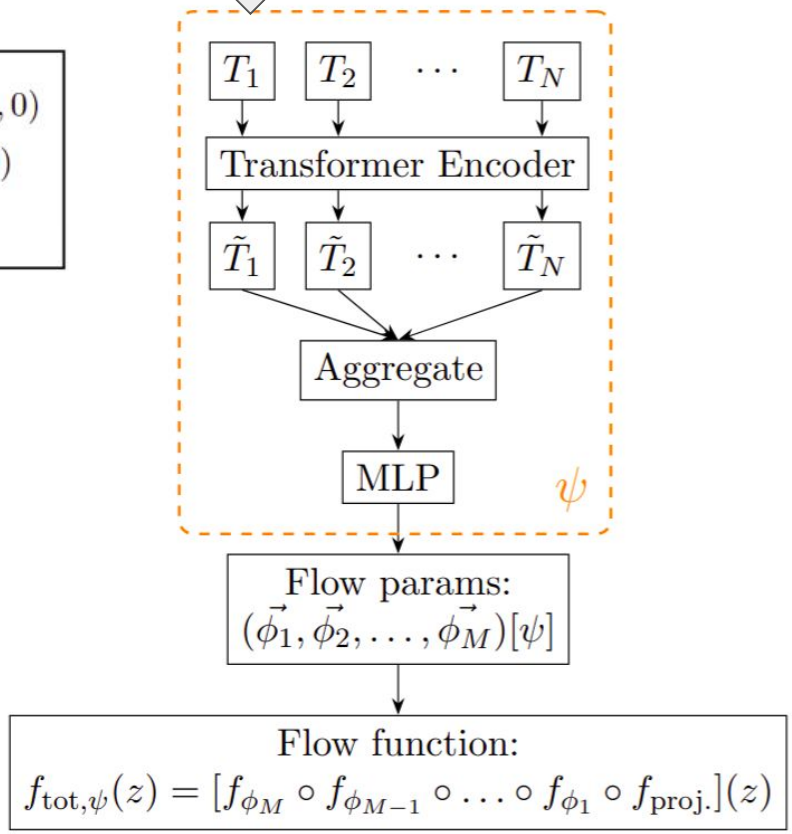
$$\text{Normal DOM} \rightarrow T_i = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{c}_x, \tilde{c}_y, \tilde{c}_z, \tilde{\vec{\boldsymbol{Q}}}, \tilde{T}_{1st}, \tilde{\vec{\boldsymbol{T}}}_{\boldsymbol{o}}, \mathrm{PMT}_{\mathrm{type}}, 1, 0, 0)$$

$$\text{Saturated DOM} \rightarrow T_i = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{c}_x, \tilde{c}_y, \tilde{c}_z, \vec{\mathbf{0}}, \tilde{T}_{1st}, \vec{\mathbf{0}}, \mathrm{PMT}_{\mathrm{type}}, 0, 1, 0)$$

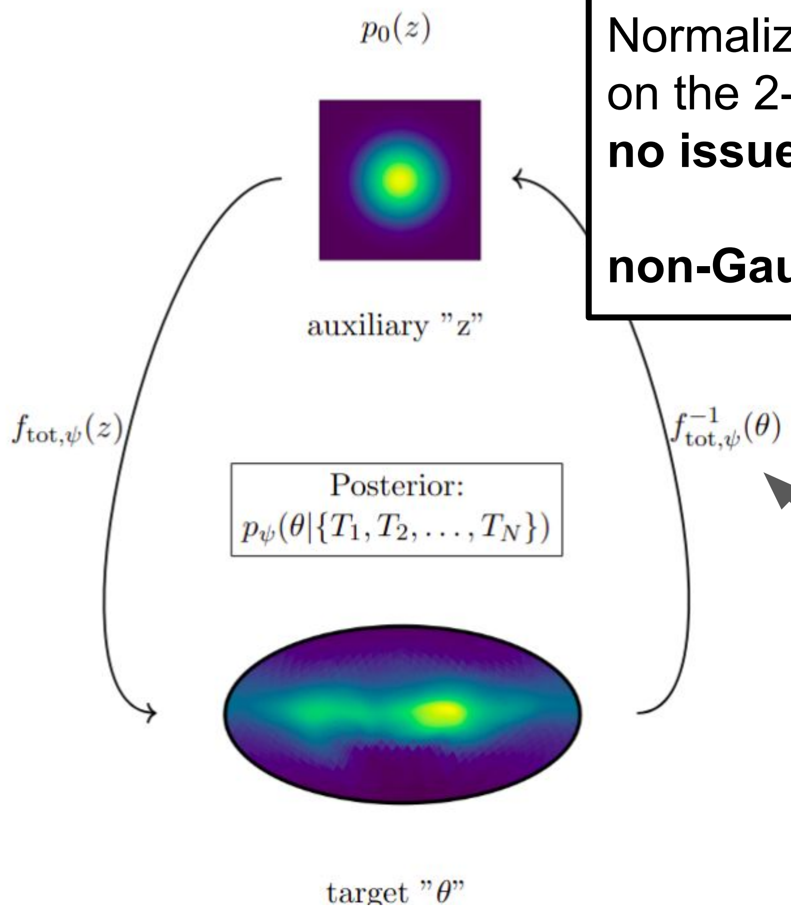
$$\text{Empty DOM} \rightarrow T_i = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{c}_x, \tilde{c}_y, \tilde{c}_z, \vec{\mathbf{0}}, 0, \vec{\mathbf{0}}, \mathrm{PMT}_{\mathrm{type}}, 0, 0, 1)$$

Step 1 - encoding the data

$$
\begin{array}{ll}
\text{Normal DOM} & \rightarrow T_i = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{c_x}, \tilde{c_y}, \tilde{c_z}, \tilde{\vec{\boldsymbol{Q}}}, \tilde{T}_{1st}, \tilde{\vec{\boldsymbol{T}}}_{\boldsymbol{o}}, \mathrm{PMT}_{\mathrm{type}}, 1, 0, 0) \\
\text{Saturated DOM} & \rightarrow T_i = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{c_x}, \tilde{c_y}, \tilde{c_z}, \vec{\mathbf{0}}, \tilde{T}_{1st}, \vec{\mathbf{0}}, \mathrm{PMT}_{\mathrm{type}}, 0, 1, 0) \\
\text{Empty DOM} & \rightarrow T_i = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{c_x}, \tilde{c_y}, \tilde{c_z}, \vec{\mathbf{0}}, 0, \vec{\mathbf{0}}, \mathrm{PMT}_{\mathrm{type}}, 0, 0, 1)
\end{array}
$$

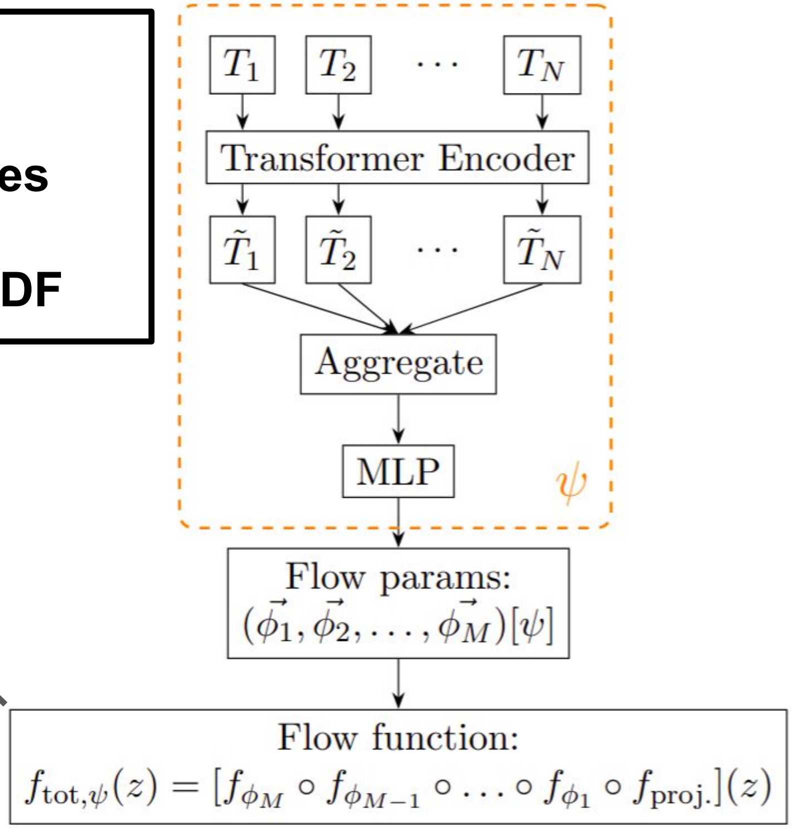


Step 2 - predicting the posterior normalizing flow

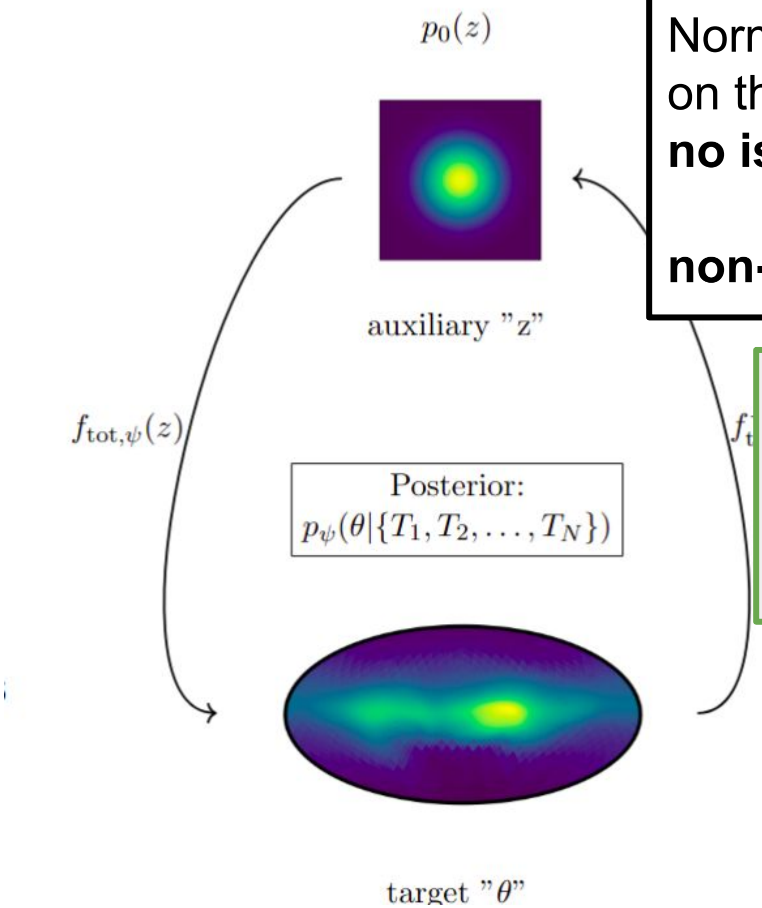


Normalizing-flow
on the 2-sphere:
no issues at poles

non-Gaussian PDF



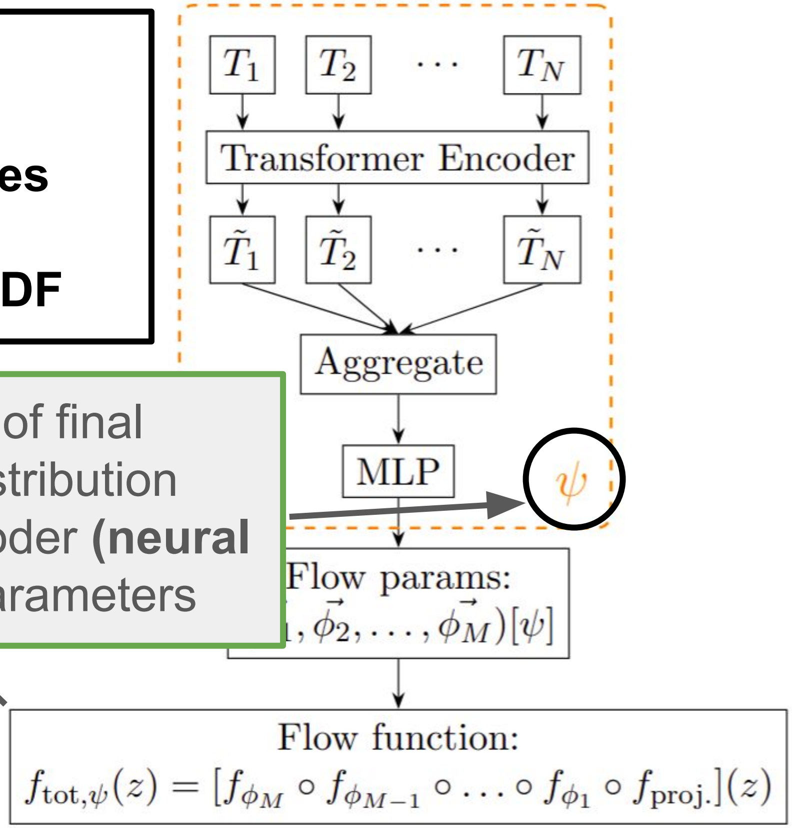
Step 2 - predicting the posterior normalizing flow



Normalizing-flow
on the 2-sphere:
no issues at poles

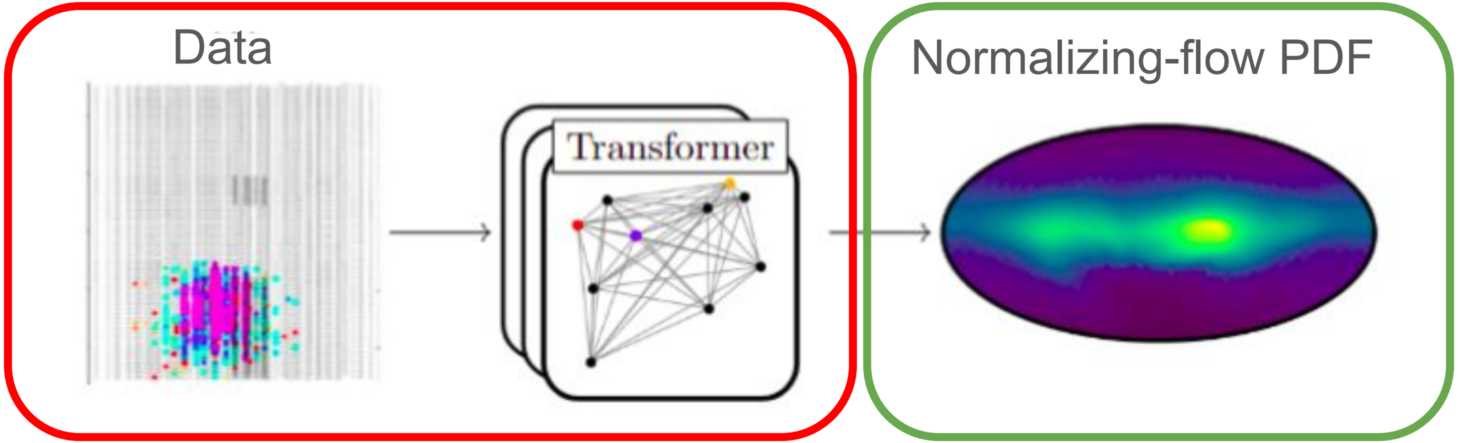
non-Gaussian PDF

Parameters of final Posterior distribution are the encoder **(neural network)** parameters



The model: a **posterior** PDF prediction machine

STEP 1



STEP 2

data encoding / compression

PDF prediction from compressed data

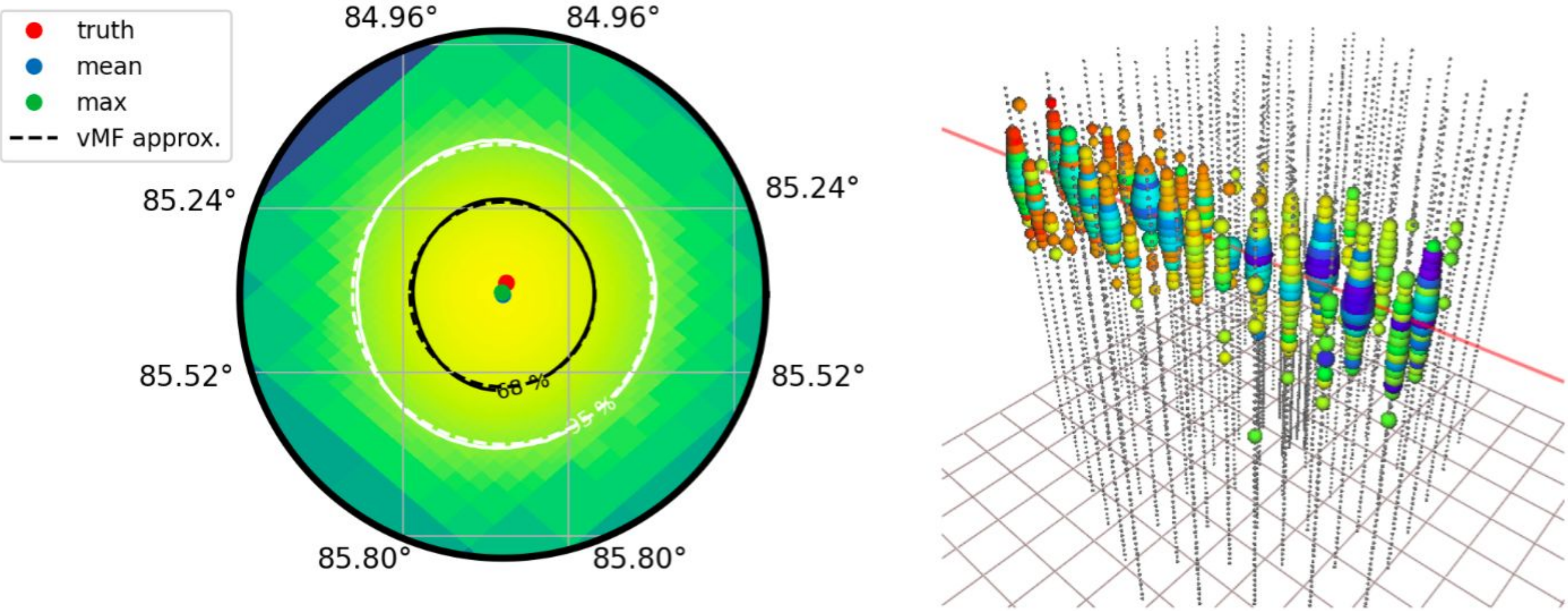
Target: 2-dim (sphere)

Posterior:

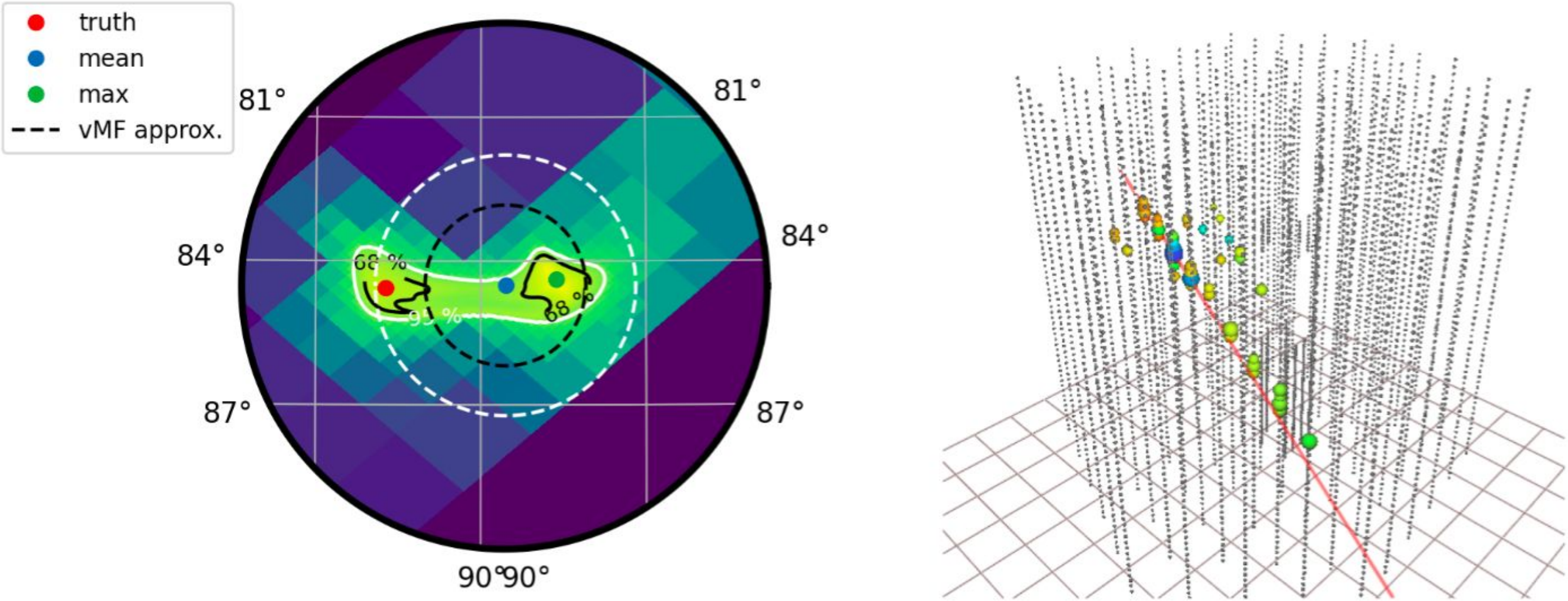
$$p_\psi(\theta|\{T_1, T_2, \ldots, T_N\})$$

5-10 mio parameters

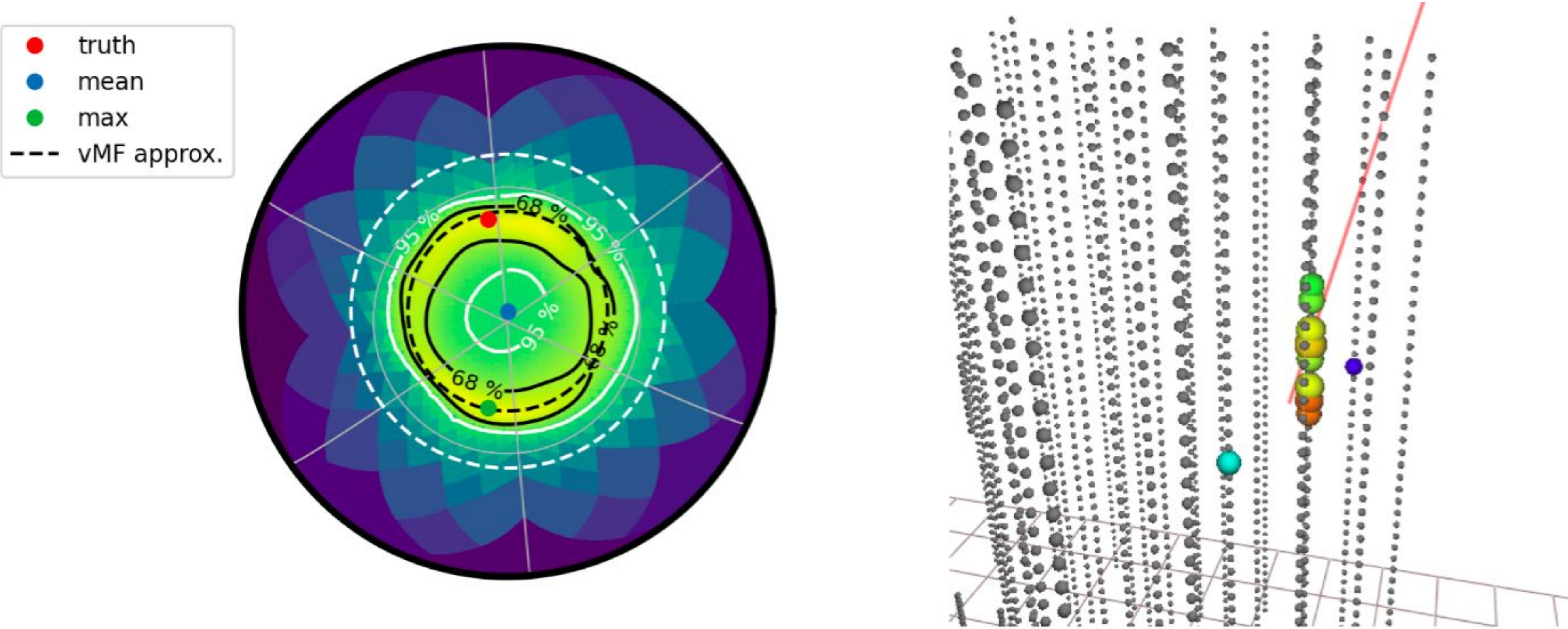
Input: ~ 100-100k dim (IceCube data)



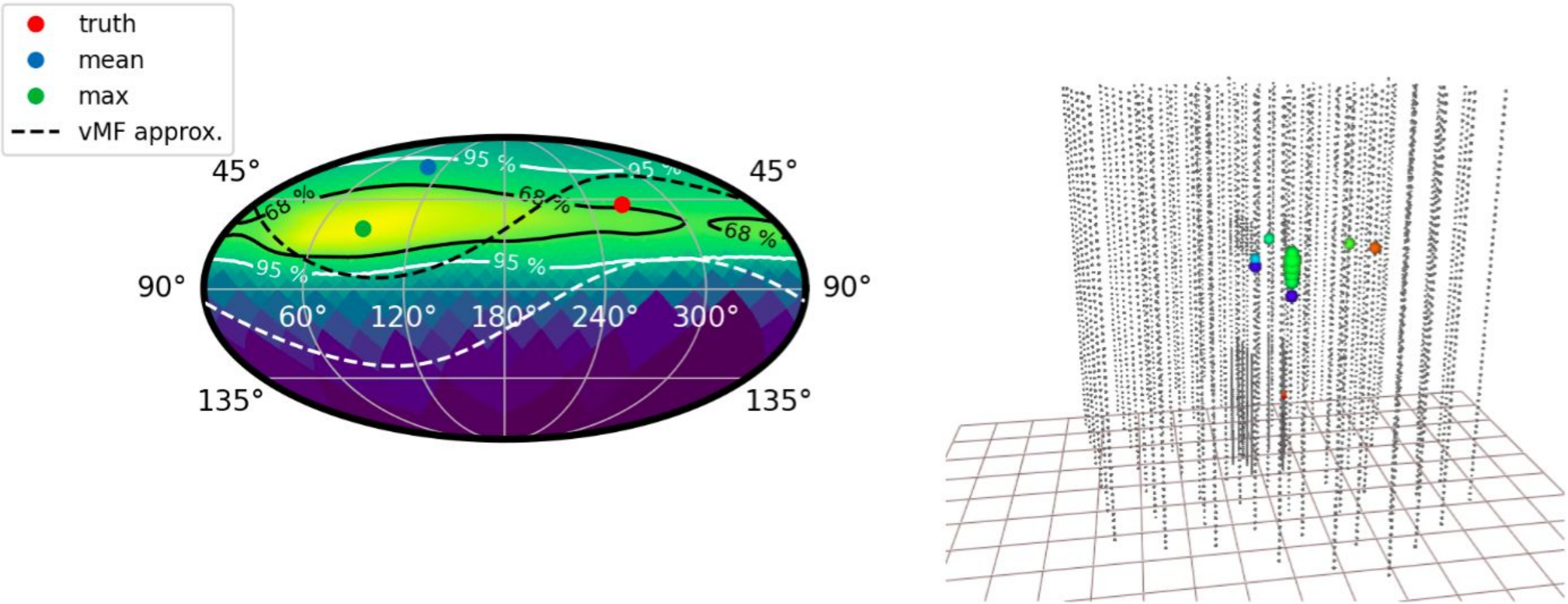
(a) Bright horizontal track.



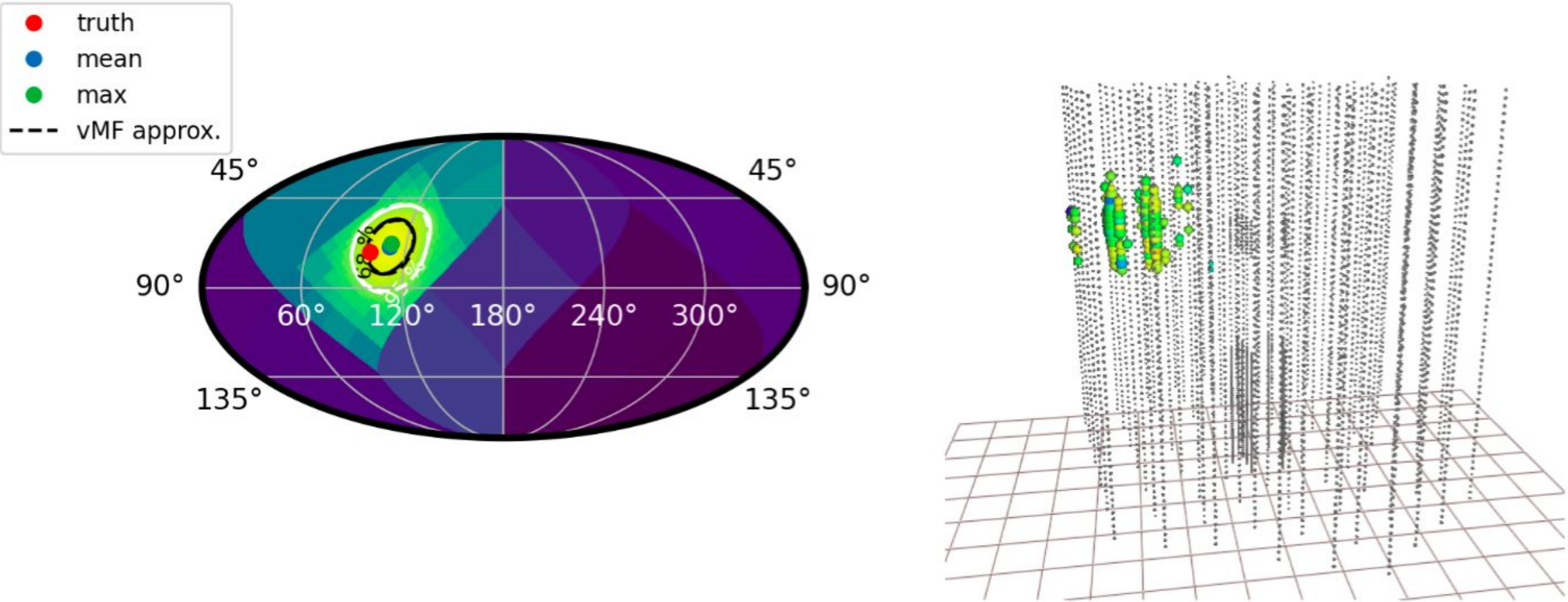
(b) Muon track along string lines.



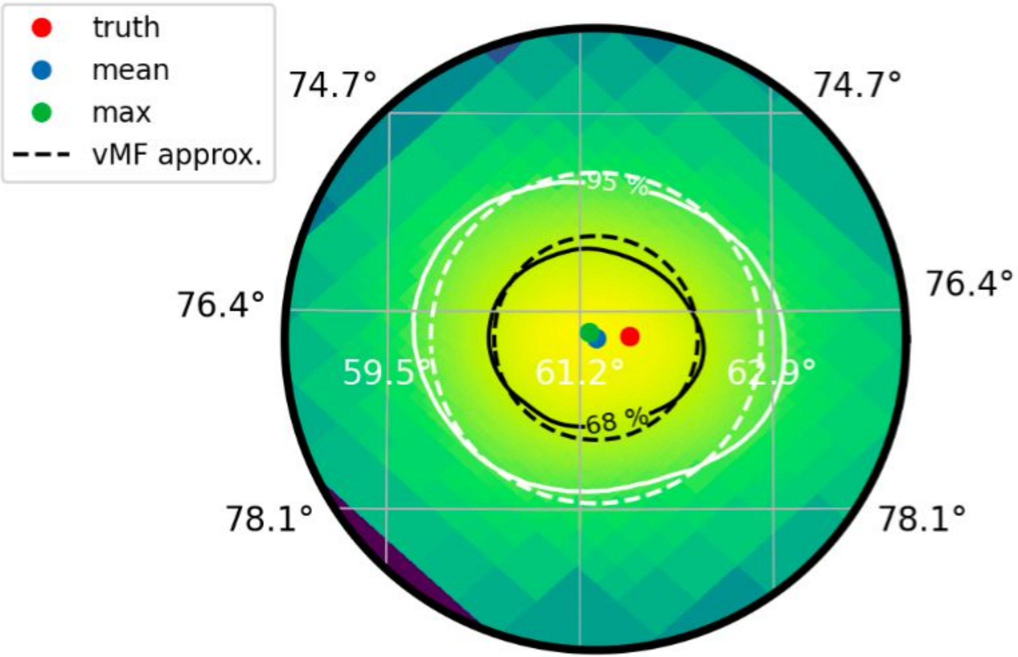
(c) Low energy contained upgoing track along a single string line.

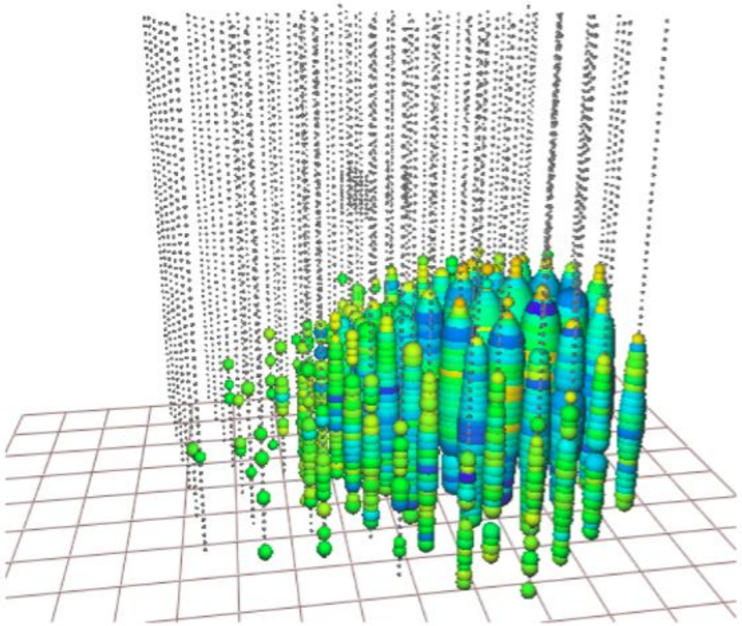


(a) A low-energy shower that is mostly visible in a single string.



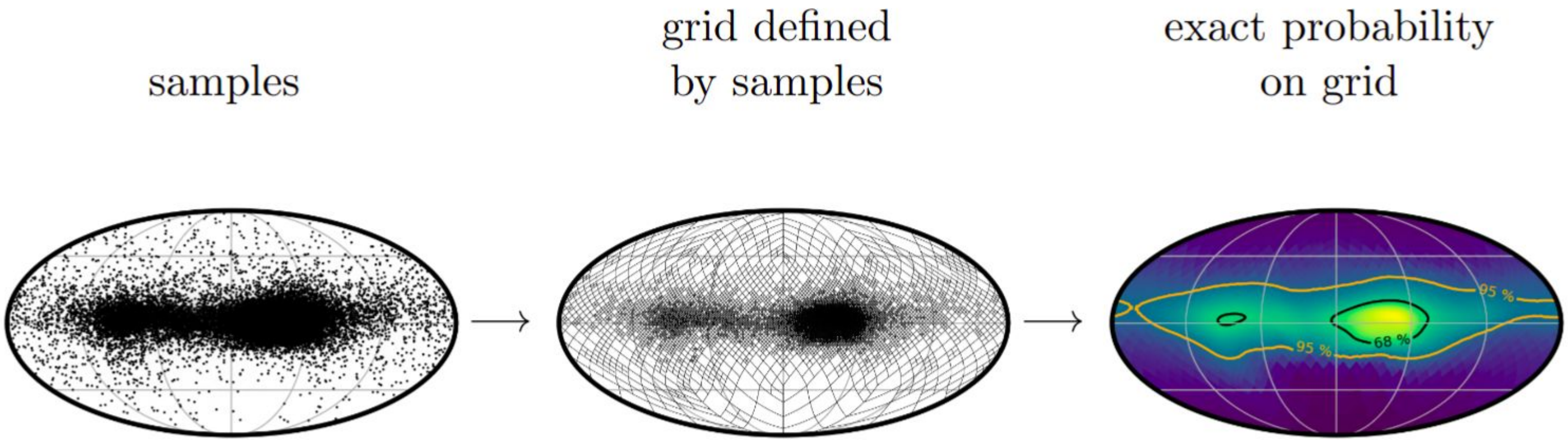
(b) A shower which is visible in several neighboring strings.





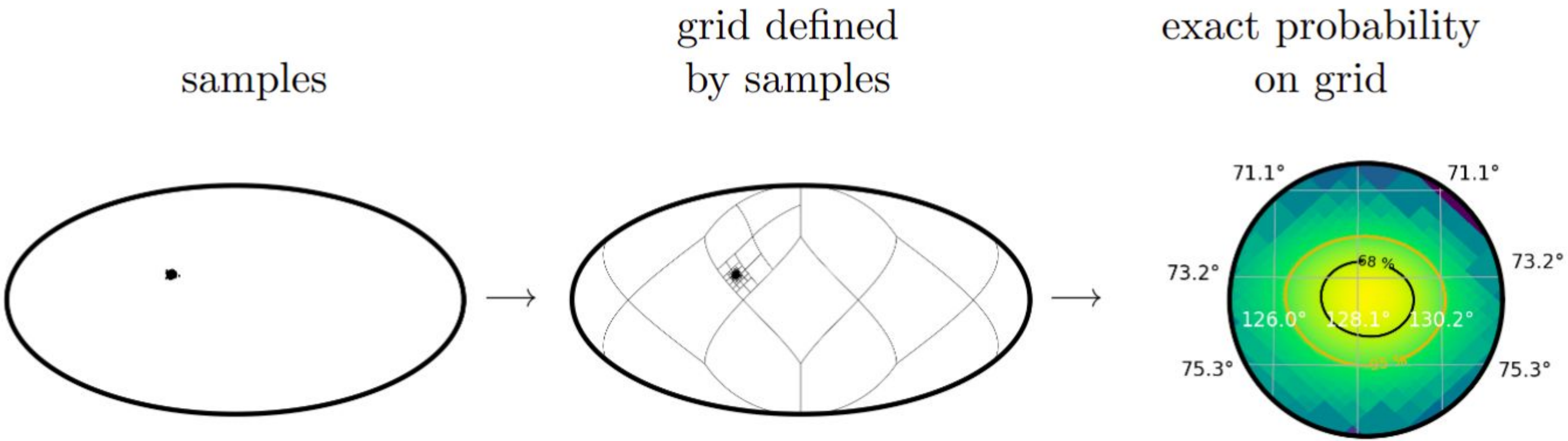
(c) A multi-PeV shower in the clear ice.

Adaptive grid in constant time (extended “all-sky” posterior)



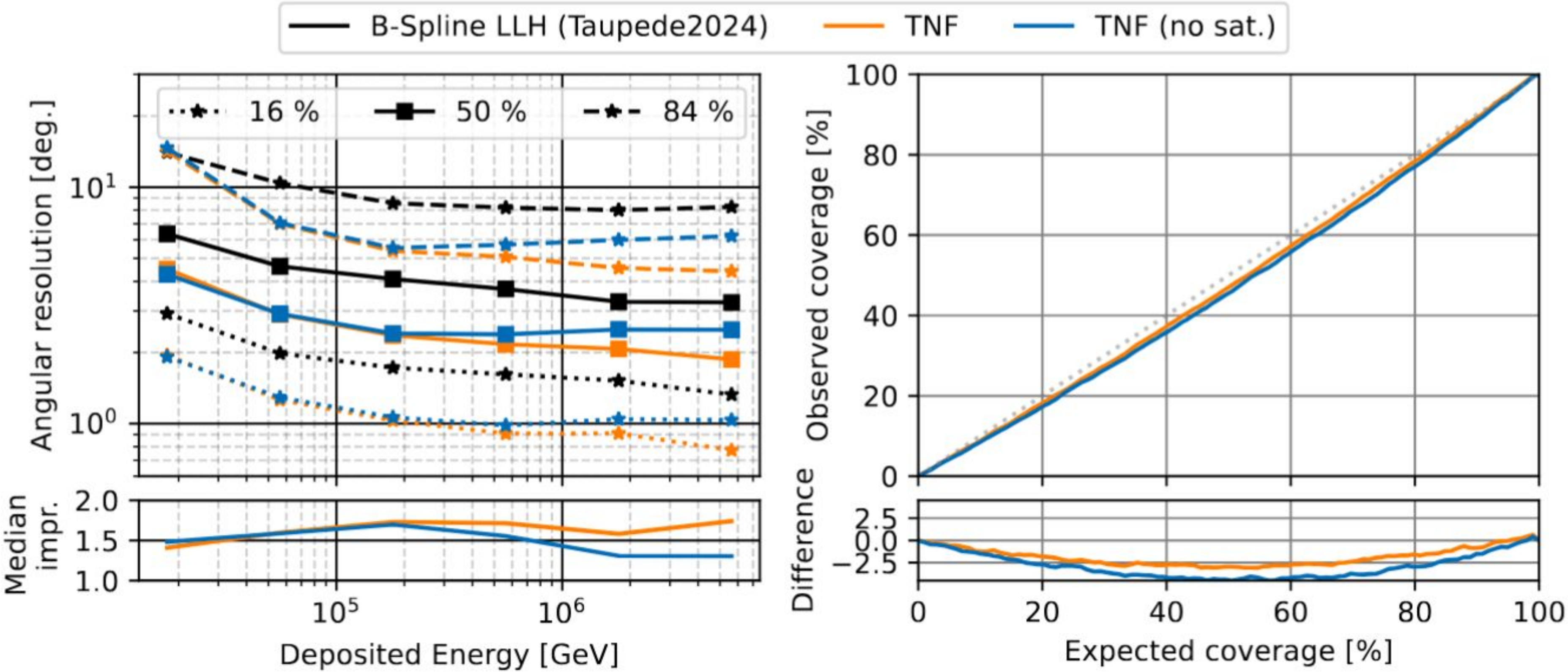
Normalizing flows: sampling + exact log-pdf evaluation **simultaneously**!

Adaptive grid in constant time (small localized posterior)



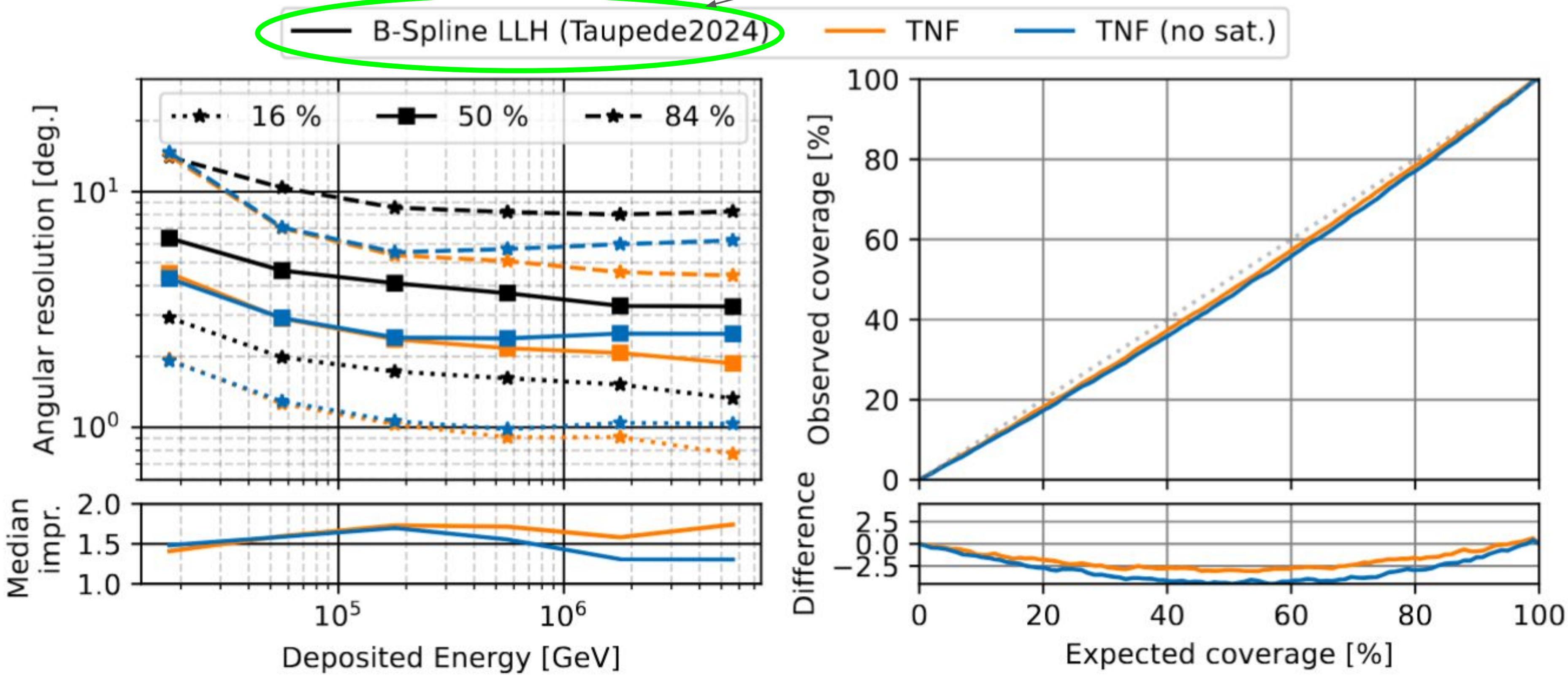
Normalizing flows: sampling + exact log-pdf evaluation **simultaneously**!

Angular resolution - showers

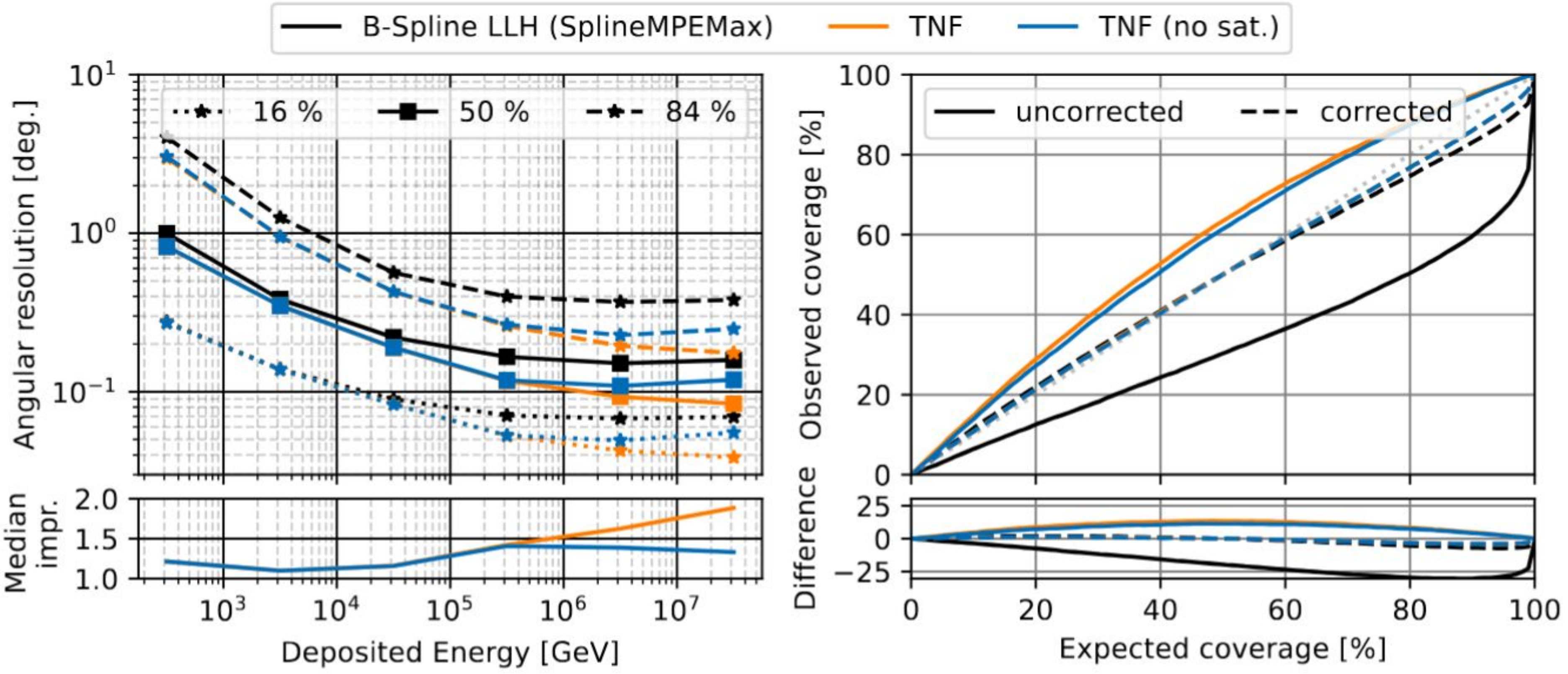


Angular resolution - shower

Used in most recent shower sample for Galactic Plane analysis (PoS-ICRC2025-1193 / 2507.08753)

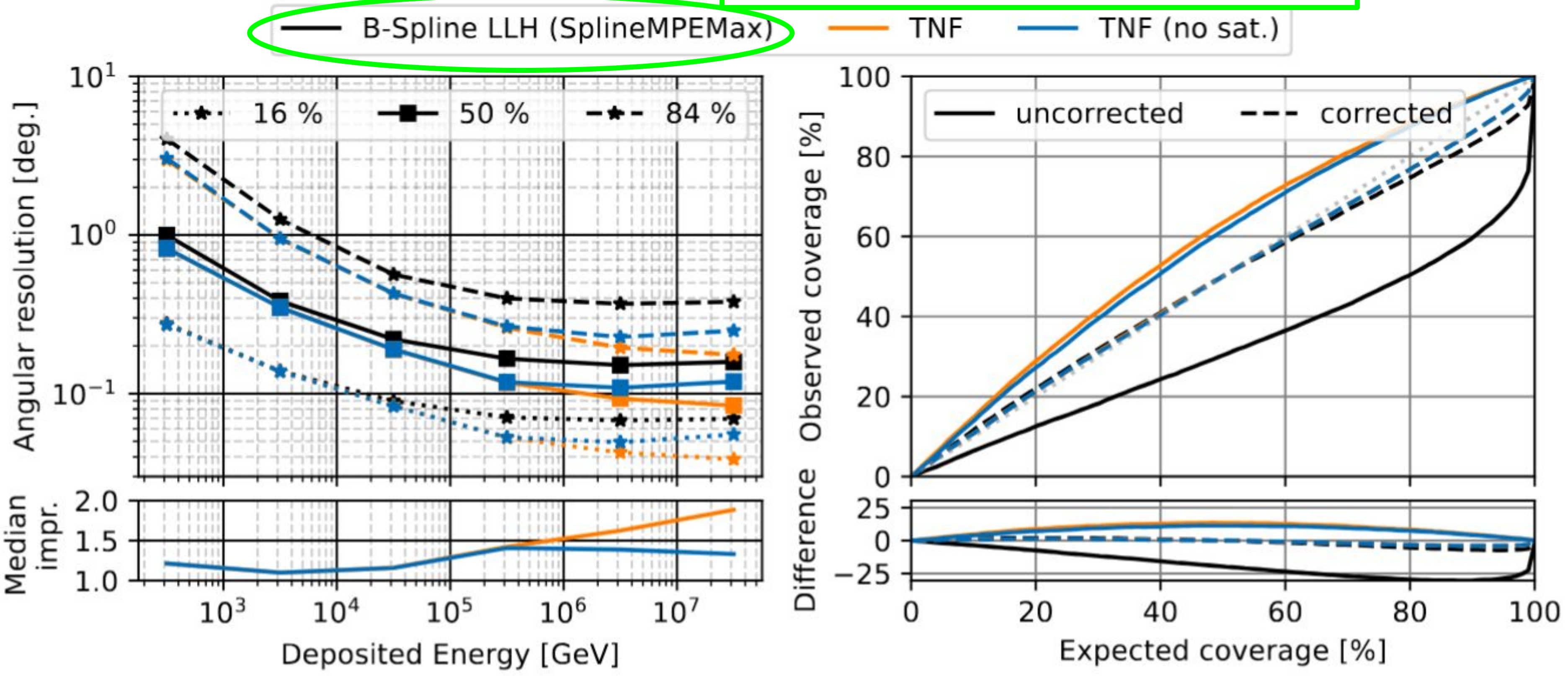


Throughgoing tracks (>300 m)

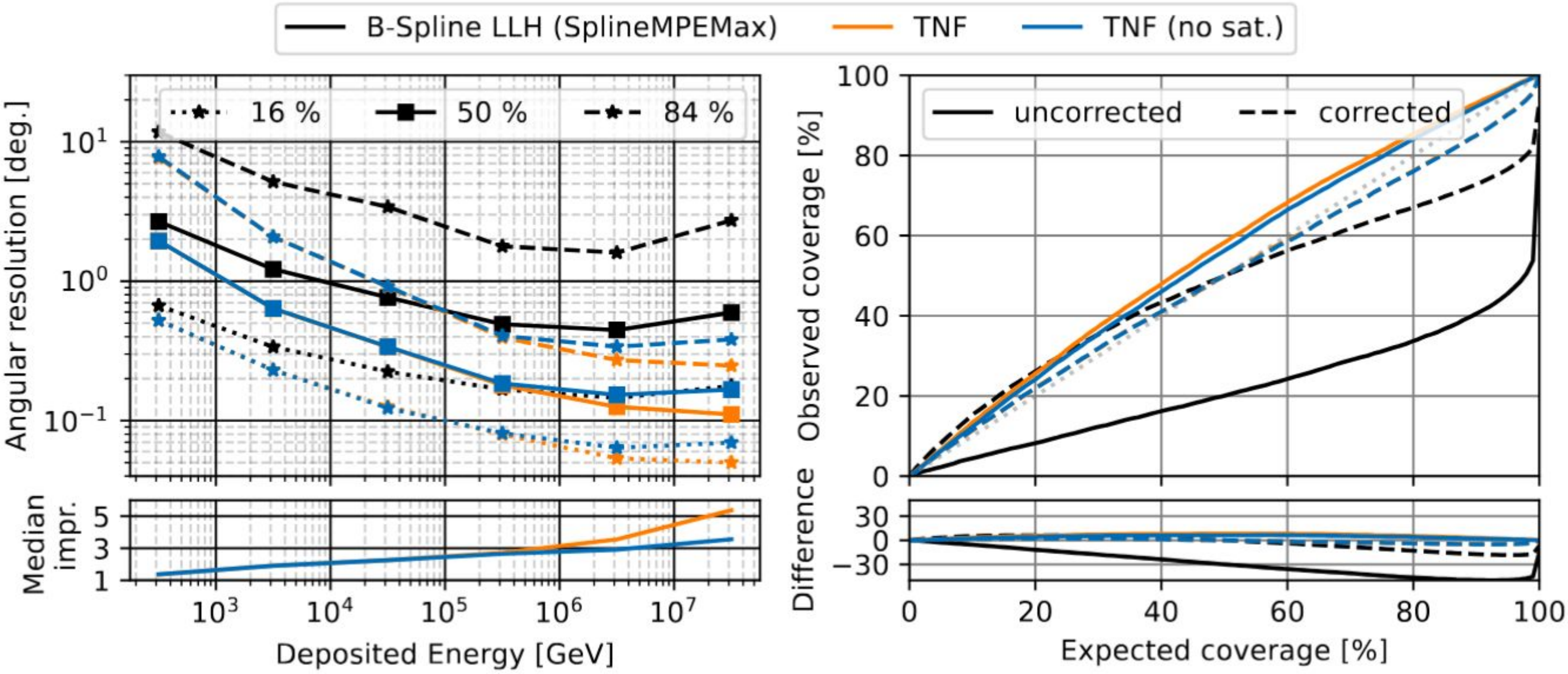


Throughgoing tracks (>300 m)

Used in all Point source results since 2013



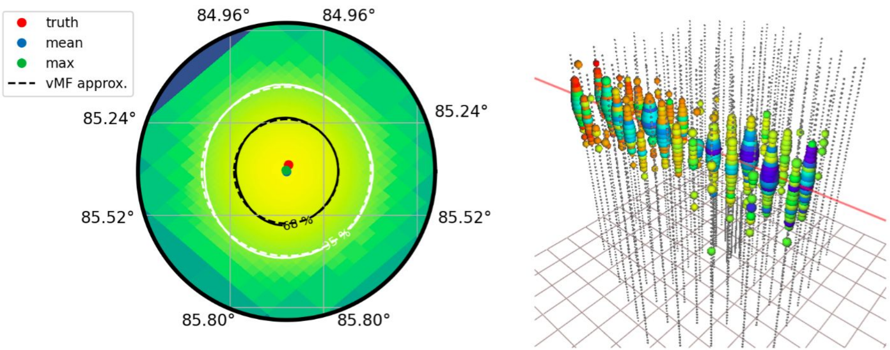
Starting tracks (>300 m)



Substantial improvement for starting tracks! Relevant for Galactic plane!

Summary

- **Transformer Normalizing Flows (TNF) are a suitable model for neutrino detectors (arXiv:2604.19846)**
- **Explicitly learn full posterior from data:** $p_\psi(\theta|\{T_1, T_2, \ldots, T_N\})$



(a) Bright horizontal track.

Summary

- Transformer Normalizing Flows (TNF) are a suitable model for neutrino detectors (arXiv:2604.19846)
- Explicitly learn full posterior from data: $p_\psi(\theta|\{T_1, T_2, \ldots, T_N\})$

  Transformer -> Can respect permutation invariance (Bayesian symmetry)
  Normalizing flow -> Flexible PDF (generalizes standard regression)

$$p(\theta|\mathbf{x}) = \frac{p(x_1;\theta)\cdot \overbrace{p(x_2;\theta)\cdot p(x_3;\theta)}^{\text{permute}}\cdot \ldots \cdot p(\theta)}{p(\mathbf{x})} = \frac{p(x_1;\theta)\cdot p(x_3;\theta)\cdot p(x_2;\theta)\cdot \ldots \cdot p(\theta)}{p(\mathbf{x})}$$

Summary

- Transformer Normalizing Flows (TNF) are a suitable model for neutrino detectors (arXiv:2604.19846)
- Explicitly learn full posterior from data: $p_\psi(\theta|\{T_1, T_2, \ldots, T_N\})$

  Transformer -> Can respect permutation invariance (Bayesian symmetry)
  Normalizing flow -> Flexible PDF (generalizes standard regression)

  For IceCube + directional application:
    - New state of the art in angular resolution: showers and tracks
    - Full posterior skymap for astronomy - irrespective of shape+size in 1s

Summary

- Transformer Normalizing Flows (TNF) are a suitable model for neutrino detectors (arXiv:2604.19846)
- Explicitly learn full posterior from data: $p_\psi(\theta | \{T_1, T_2, \ldots, T_N\})$

  Transformer -> Can respect permutation invariance (Bayesian symmetry)
  Normalizing flow -> Flexible PDF (generalizes standard regression)

  For IceCube + directional application:
  - New state of the art in angular resolution: showers and tracks
  - Full posterior skymap for astronomy - irrespective of shape+size in 1s

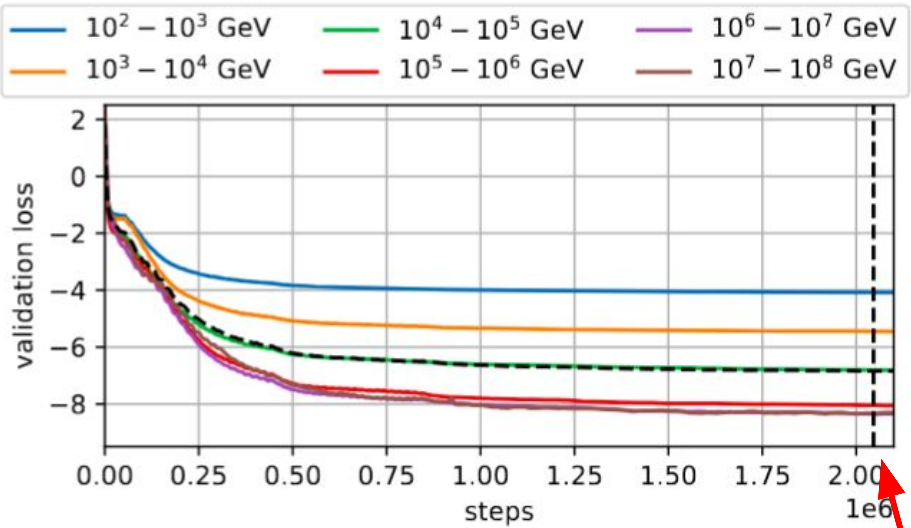
- Normalizing flow implemented in:
  https://github.com/thoglu/jammy_flows/

$$f_u(\theta) = [f_{\phi_{15}} \circ \ldots \circ f_{\phi_1}](\theta)$$

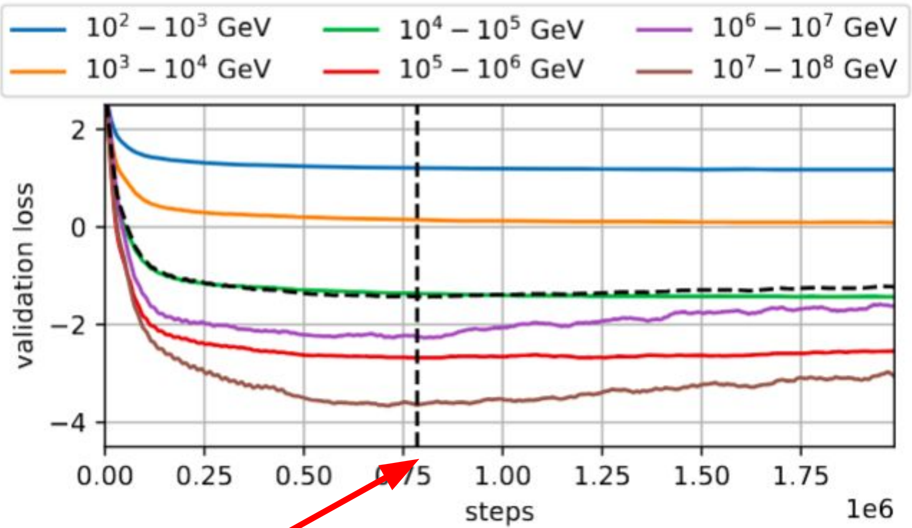
```
pdf=jammy_flows.pdf("s2", "ffffffffffffff", options_overwrite=opt_dict)
```

Backup

Example validation curves



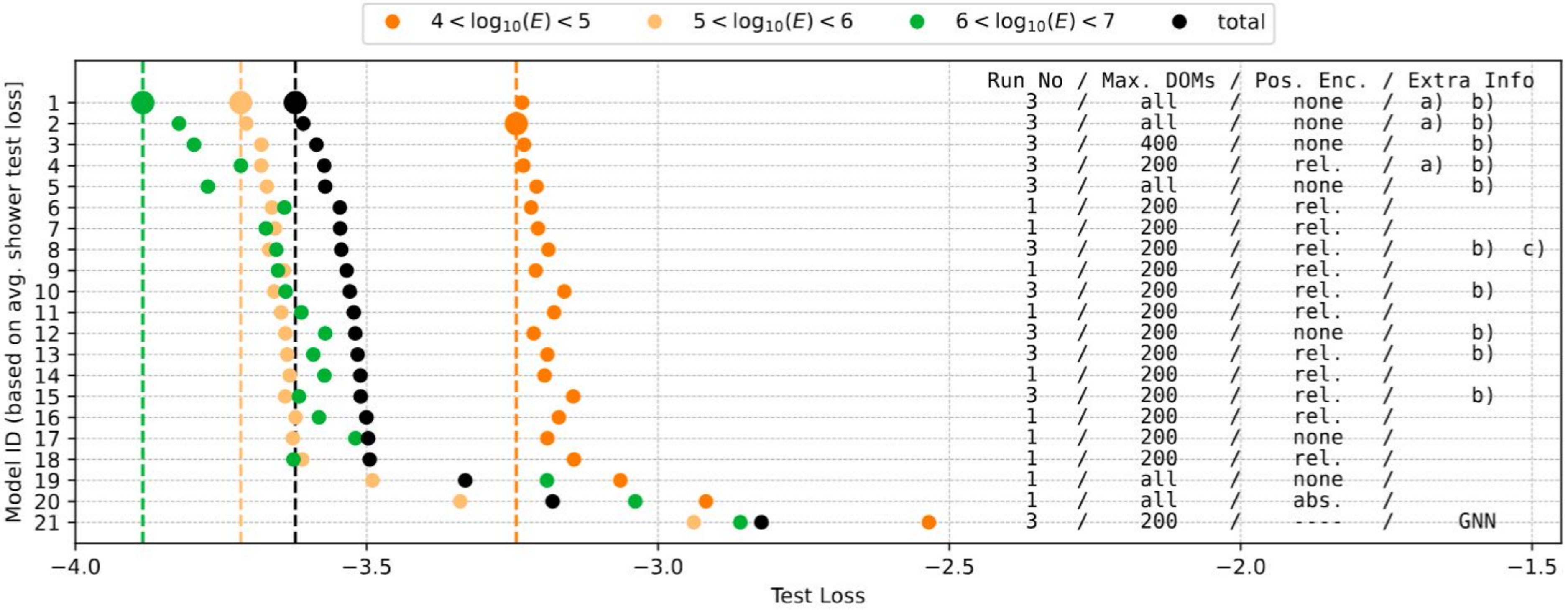
(a) Best track model.



(b) Best shower model

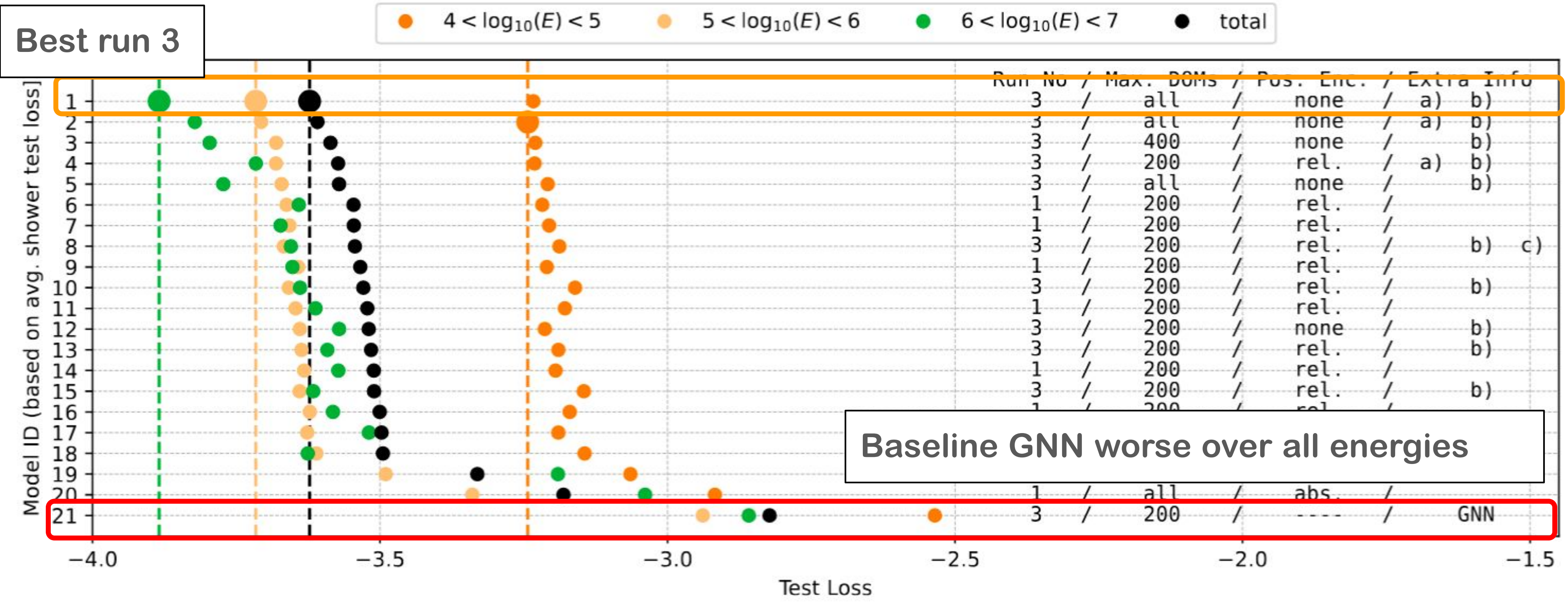
Model frozen for testing

Showers: test results on ~20 different shower model trainings



"Total" = average over individual log-energy results

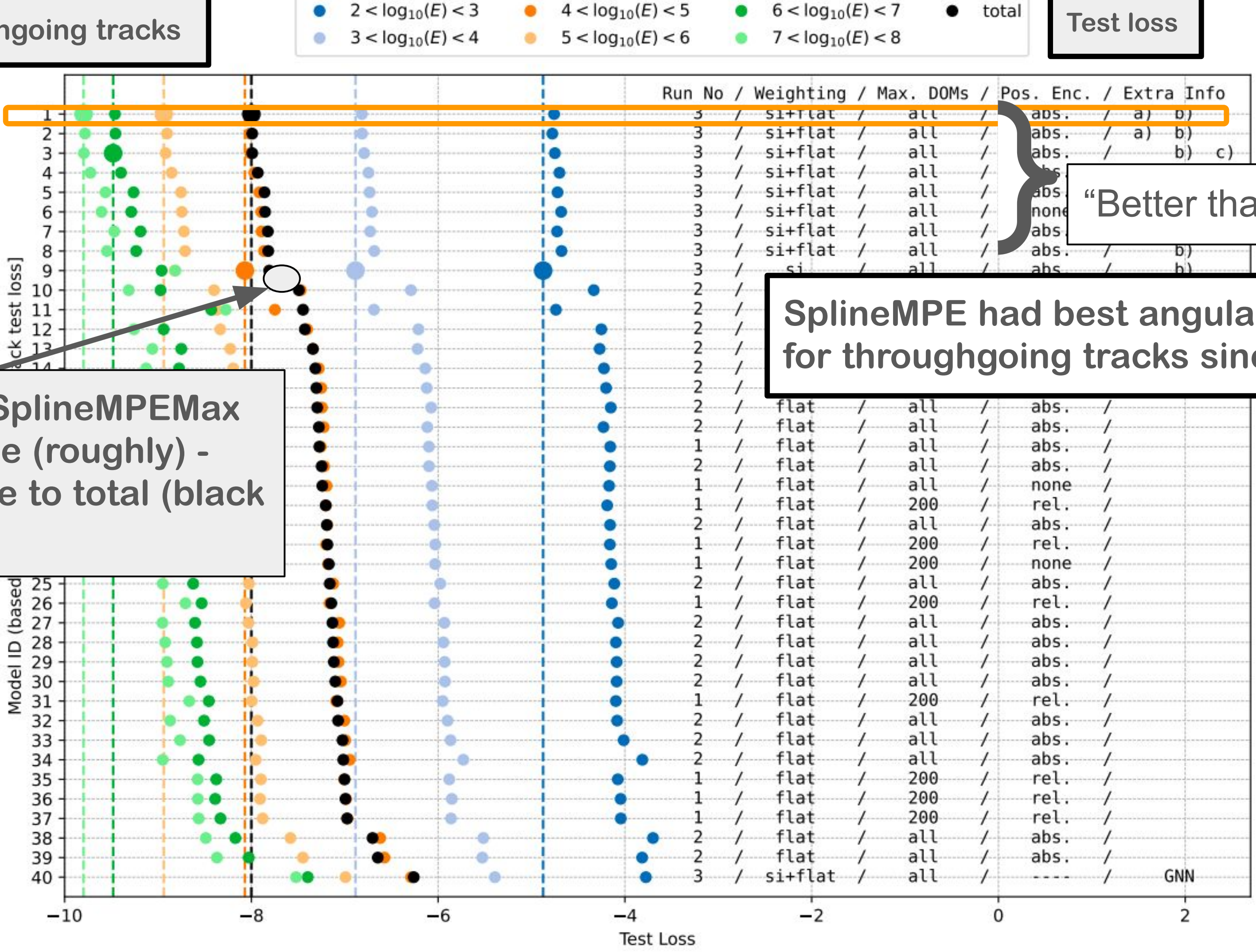
Showers: test results on ~20 different shower model trainings



“Total” = average over individual log-energy results

Throughgoing tracks

Test loss



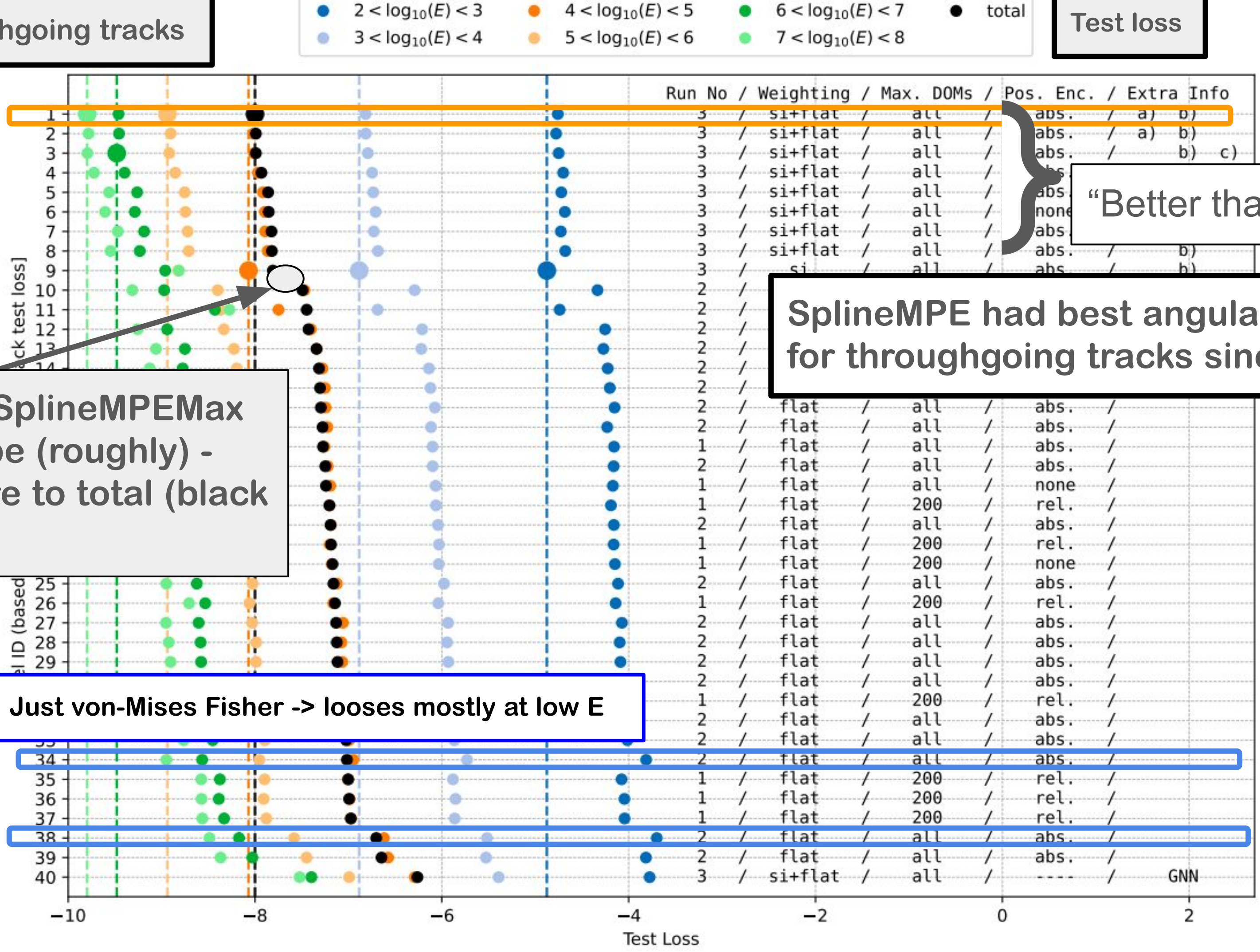
“Better than SplineMPE”

SplineMPE had best angular resolution for throughgoing tracks since 2013!

Where SplineMPEMax would be (roughly) - compare to total (black dots)

Throughgoing tracks

Test loss

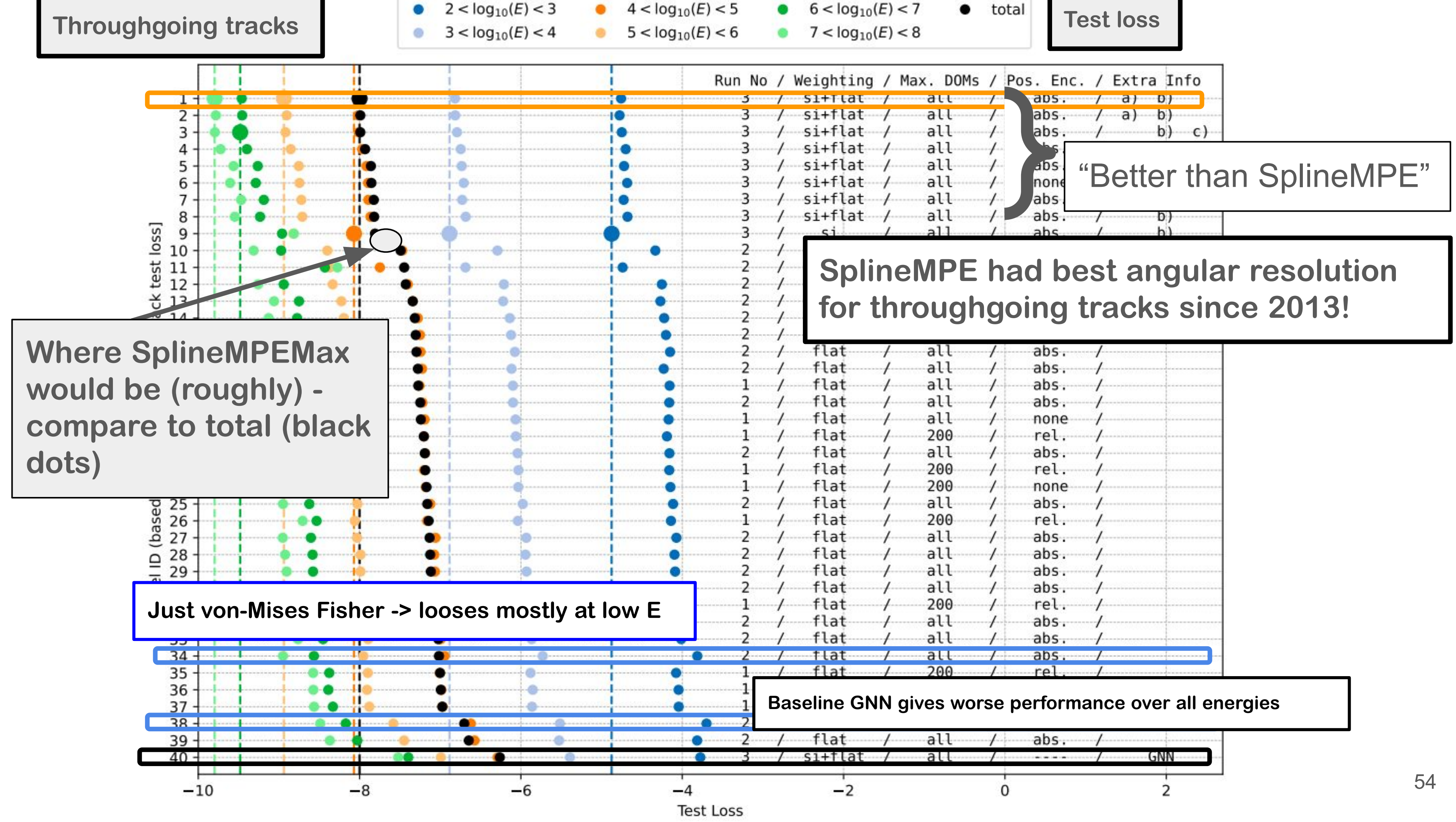


“Better than SplineMPE”

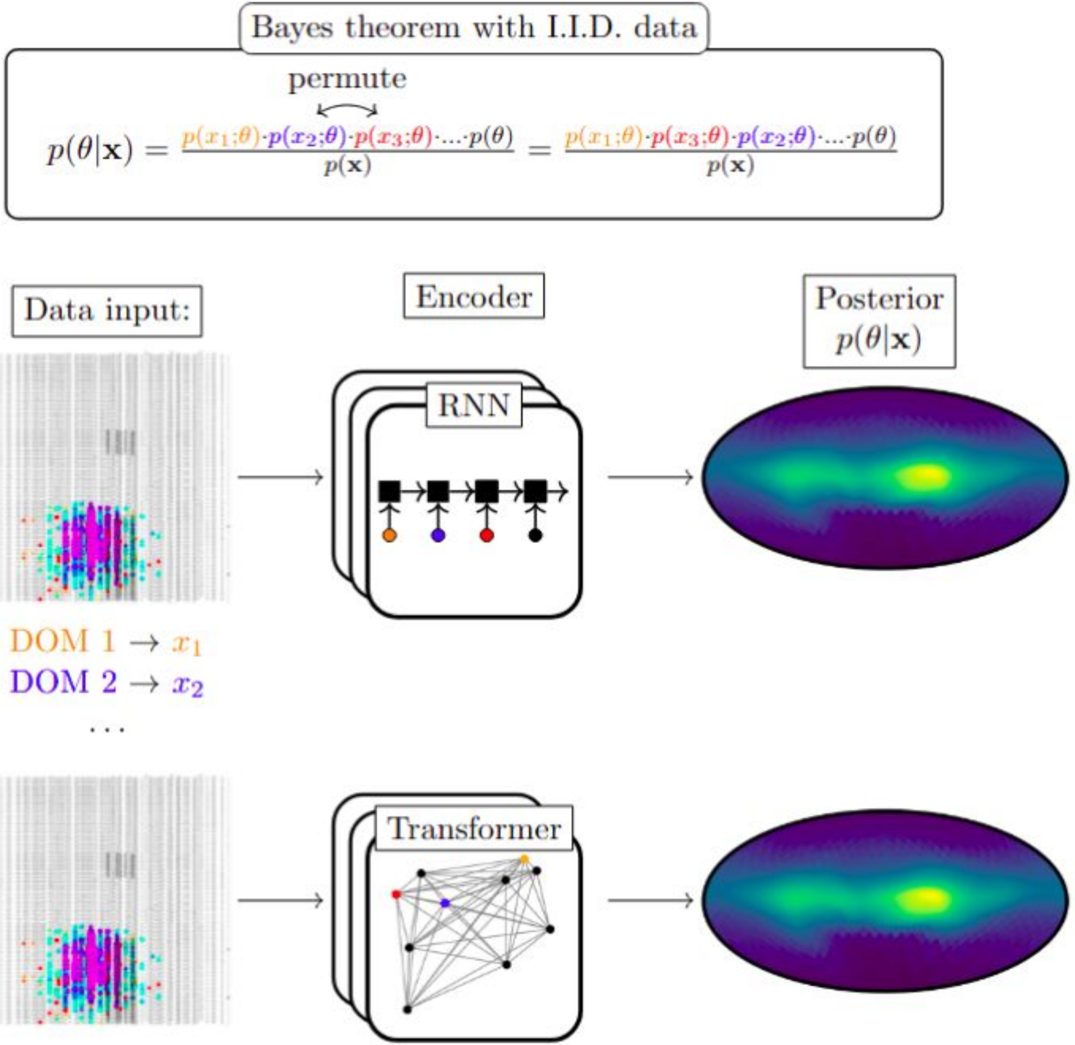
SplineMPE had best angular resolution for throughgoing tracks since 2013!

Where SplineMPEMax would be (roughly) - compare to total (black dots)

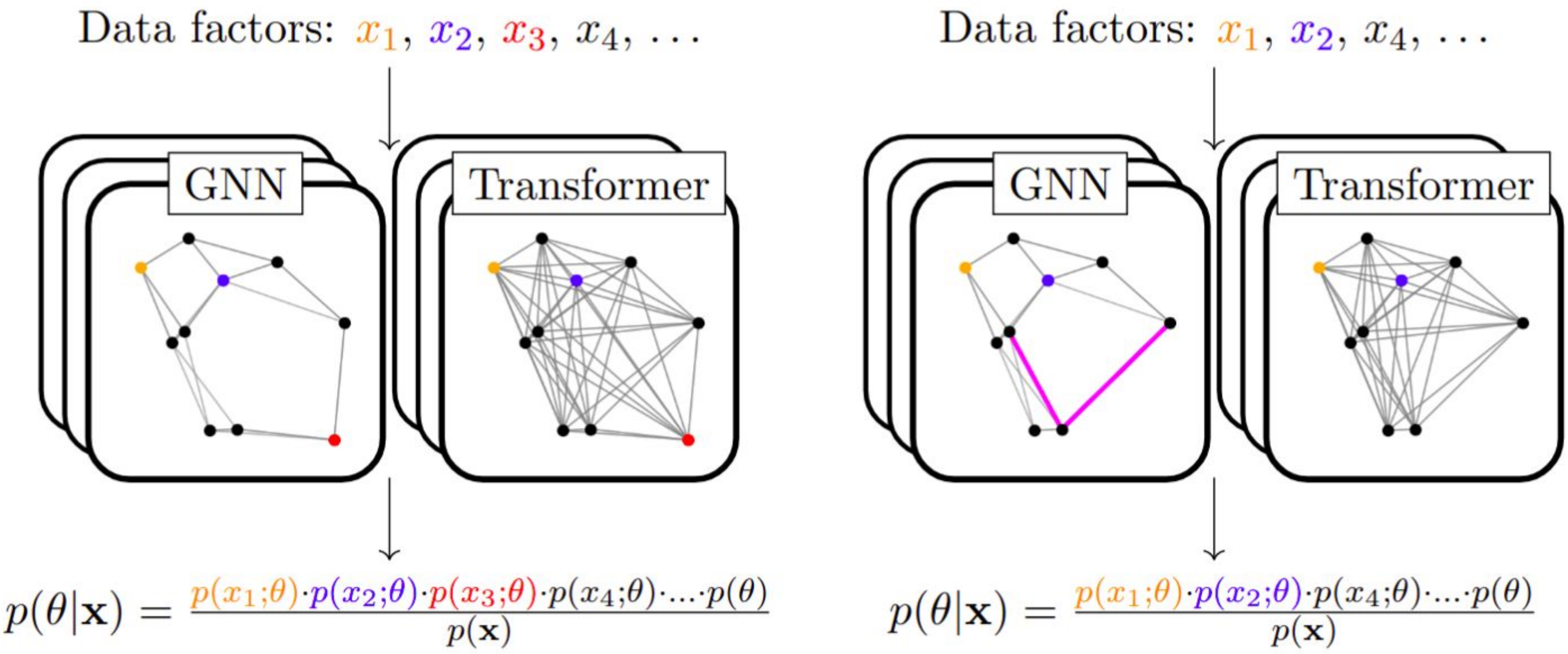
Just von-Mises Fisher -> looses mostly at low E

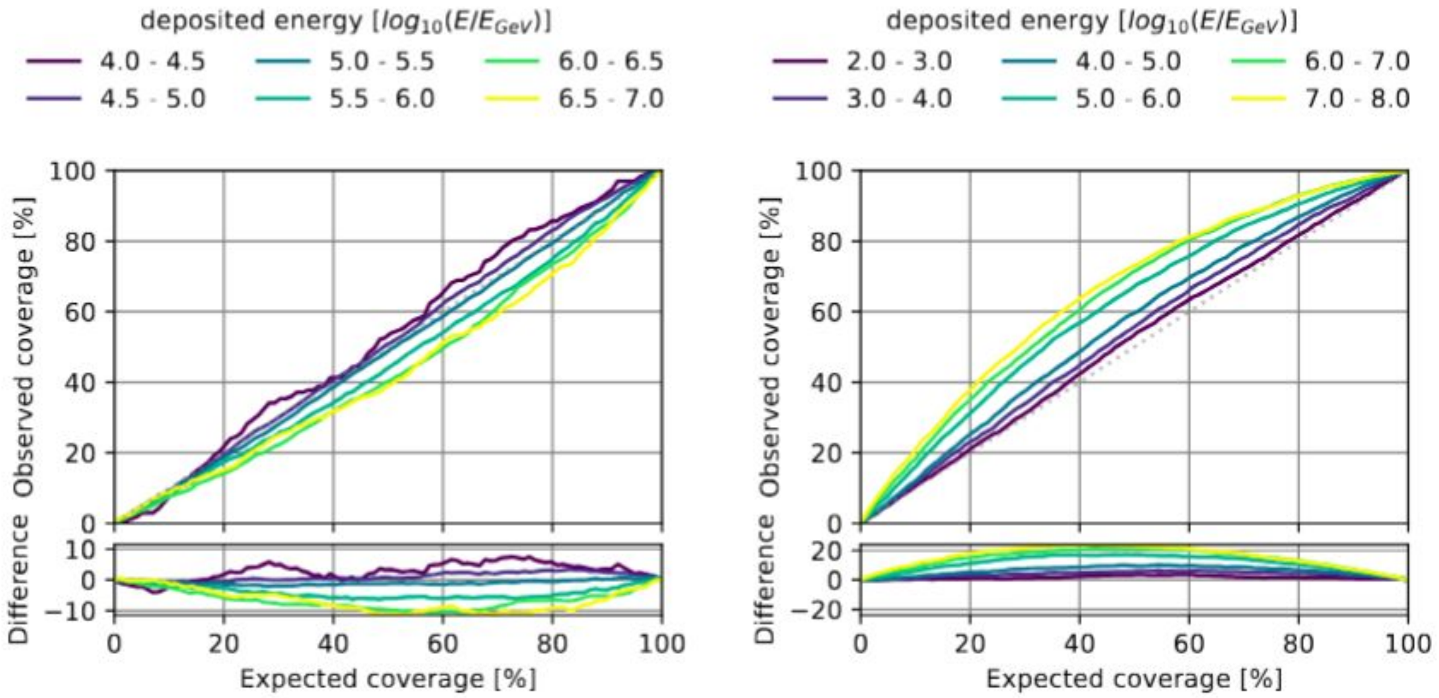

Throughgoing tracks
$2 < \log_{10}(E) < 3$
$3 < \log_{10}(E) < 4$
$4 < \log_{10}(E) < 5$
$5 < \log_{10}(E) < 6$
$6 < \log_{10}(E) < 7$
$7 < \log_{10}(E) < 8$
total
Test loss
Run No / Weighting / Max. DOMs / Pos. Enc. / Extra Info
“Better than SplineMPE”
SplineMPE had best angular resolution for throughgoing tracks since 2013!
Where SplineMPEMax would be (roughly) - compare to total (black dots)
Just von-Mises Fisher -> looses mostly at low E
Baseline GNN gives worse performance over all energies
Test Loss
−10
−8
−6
−4
−2
0
2

“Inductive bias”



“Inductive bias”



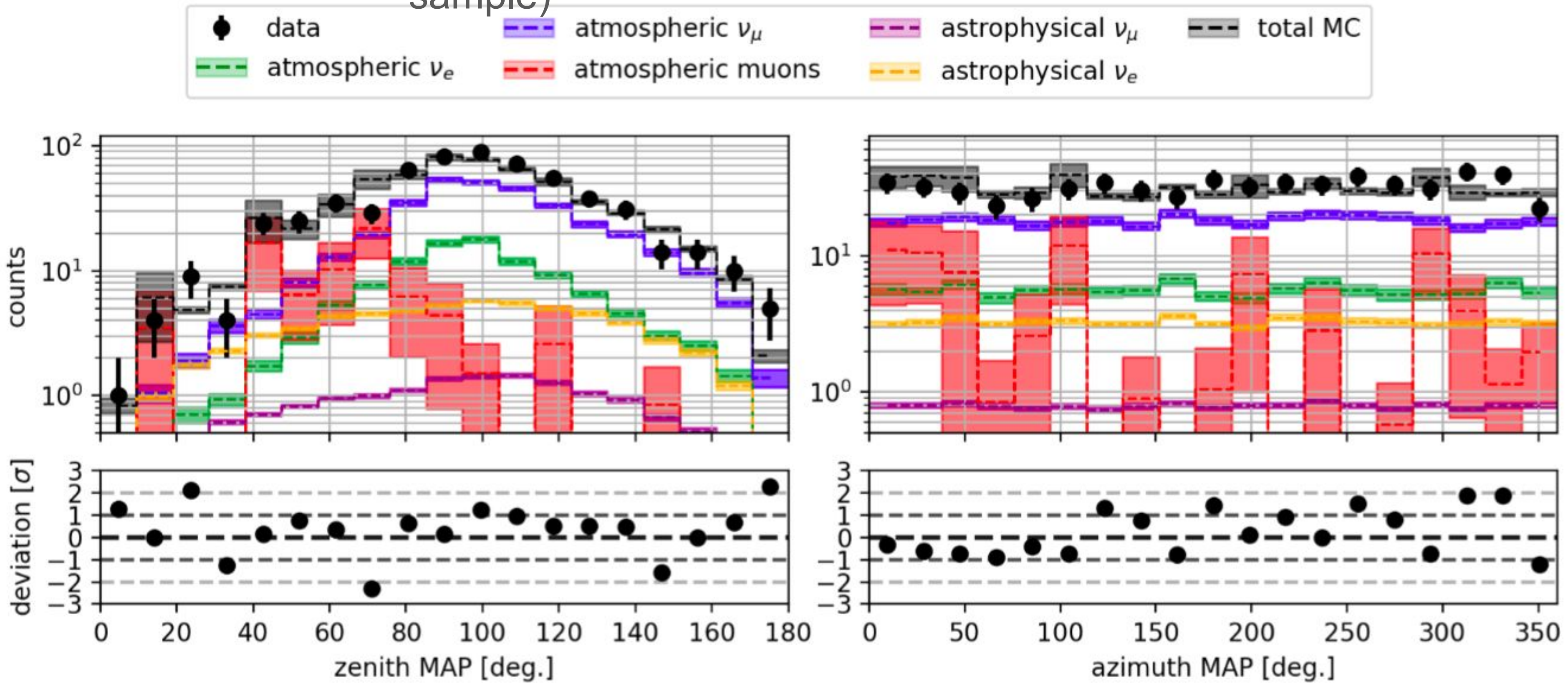


(a) Neutrino-induced showers

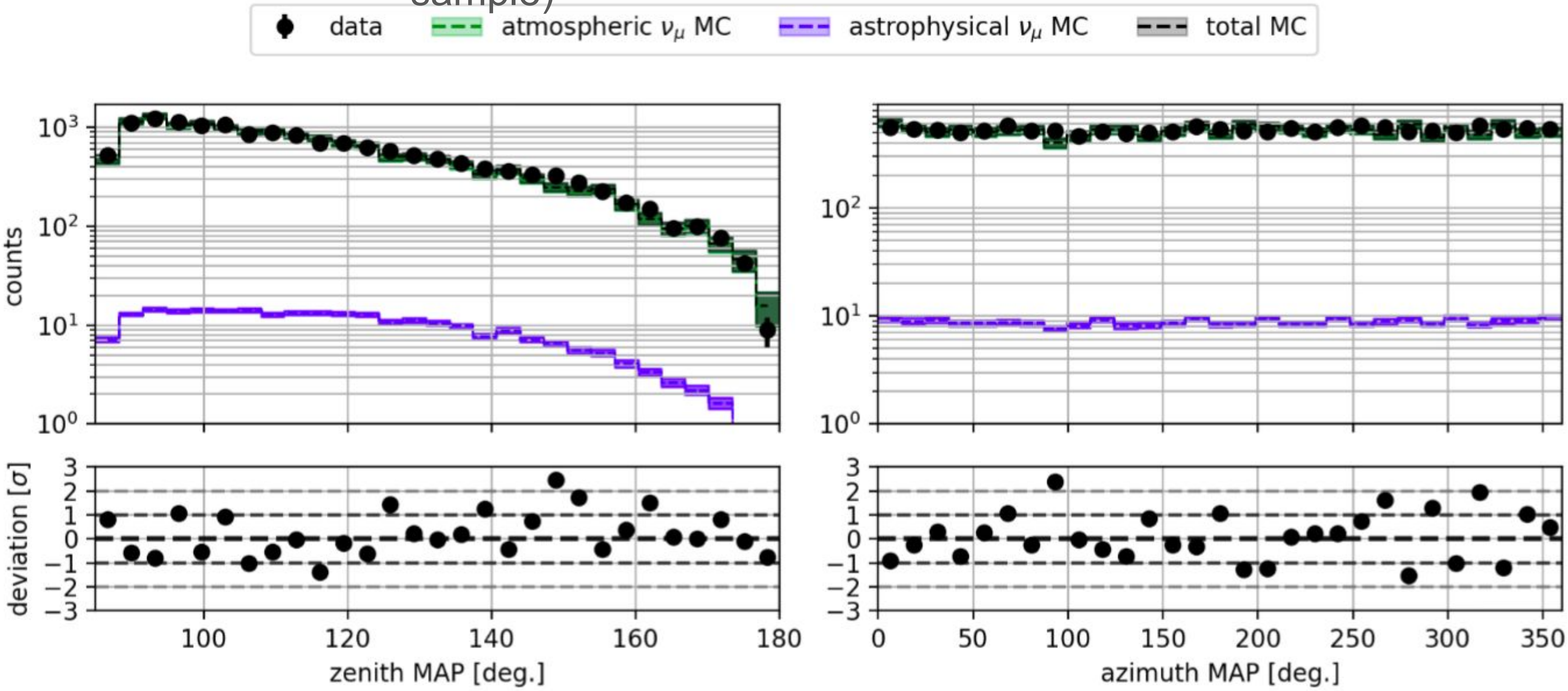
(b) Throughgoing tracks (at least 300m distance within detector)

Figure 9: Coverage split up in different deposited energy bands.

Shower zenith / azimuth (Zoe's shower sample)

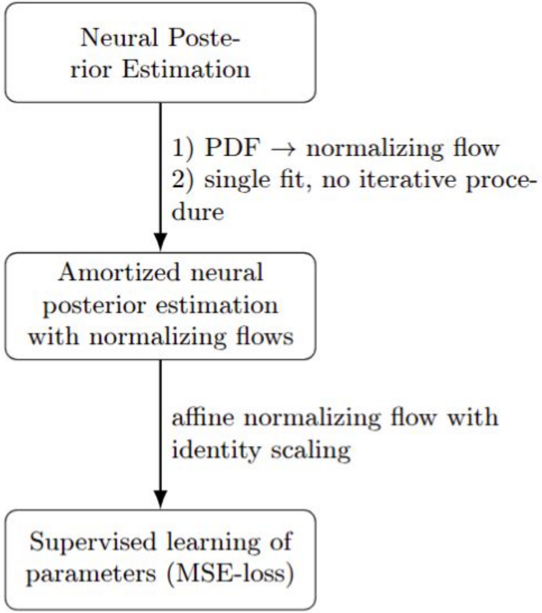


Track zenith / azimuth (northern track sample)

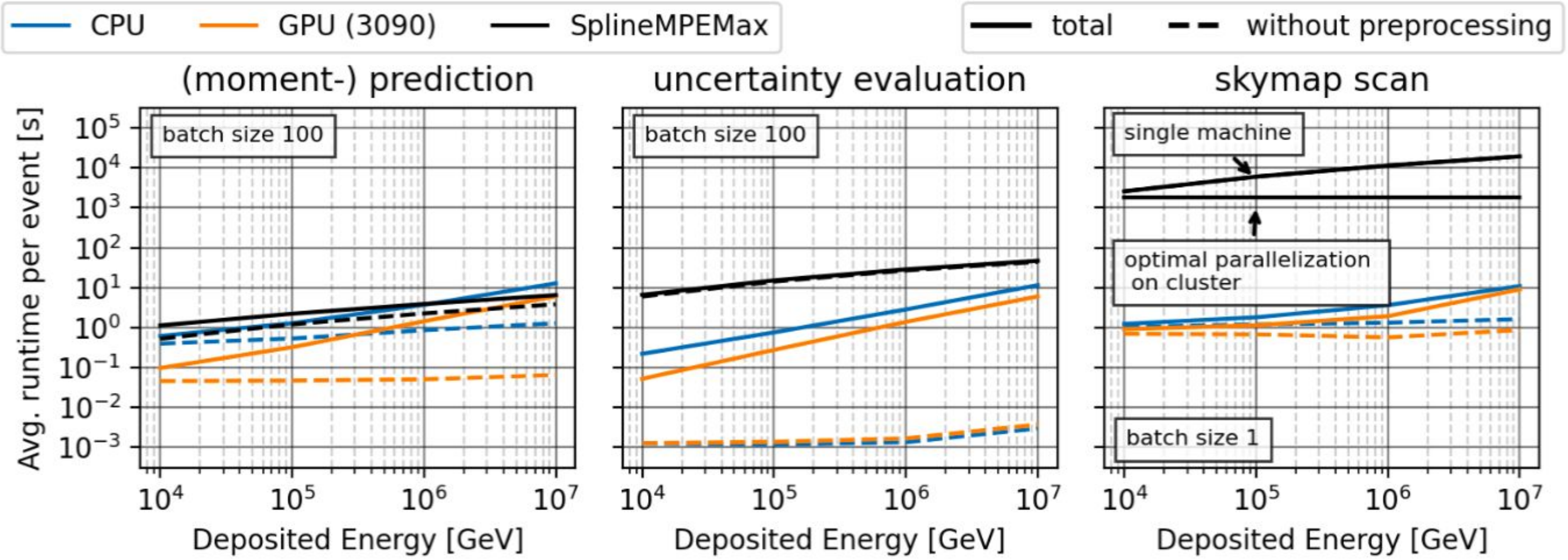


Loss function - standard cross entropy of the posterior

$$
\begin{aligned}
\psi^\star &= \arg\min_{\psi}\ D_{\mathrm{KL}}\Big(\tilde{p}(x,\theta)\ \Big\|\ q(x)\, q_\psi(\theta|x)\Big) \\
&= \arg\min_{\psi}\ \mathbb{E}_{\tilde{p}(x,\theta)}\left[\log \frac{\tilde{p}(x,\theta)}{q(x) q_\psi(\theta|x)}\right] \\
&= \arg\min_{\psi}\ \mathbb{E}_{\tilde{p}(x,\theta)}\left[D_{\mathrm{KL}}\Big(\tilde{p}(\theta|x)\ \Big\|\ q_\psi(\theta|x)\Big)\right] + \mathrm{const} \\
&= \arg\min_{\psi}\ \mathbb{E}_{\tilde{p}(x,\theta)}[-\log q_\psi(\theta|x)] \qquad \text{(drop terms constant in } \psi\text{)} \\
&\approx \arg\min_{\psi}\ \frac{1}{N}\sum^{N} -\log q_\psi(\theta_i|x_i) \equiv \arg\min_{\psi}\ \mathcal{L}_{\mathrm{NPE}}(\psi).
\end{aligned}
$$



Timing (Tracks)



Timing (Showers)