A Fully Differentiable Pipeline for the Optimization of TAMBO

Zlatan Dimitrov, Hamza Hanif, Jay Agarwal, Izan Flórez García, Jeffrey Lazar, Pietro Vischia, Tommaso Dorigo

z.dimitrov308@proton.me

About me: Zlatan Dimitrov

Sep 2025 –
PhD Student, GATE Institute | MODE

- Uncertainty quantification, surrogate models, differentiable programming

Sep 2024 – Sep 2025
Researcher, Sofia University St. Kliment Ohridski

- Machine learning applications in electromagnetic simulations

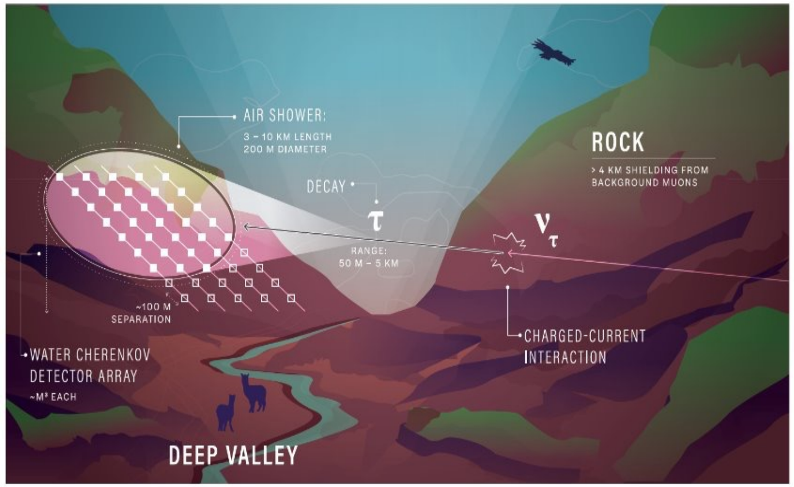
2022 – 2024
MSc, Electromagnetics, Fusion and Space Engineering, KTH Royal Institute of Technology

- Electromagnetics and Machine learning (Ericsson AB/KTH)

2018 – 2022
BSc, Information Technology / Electrical Engineering, TU Vienna / Aalto University

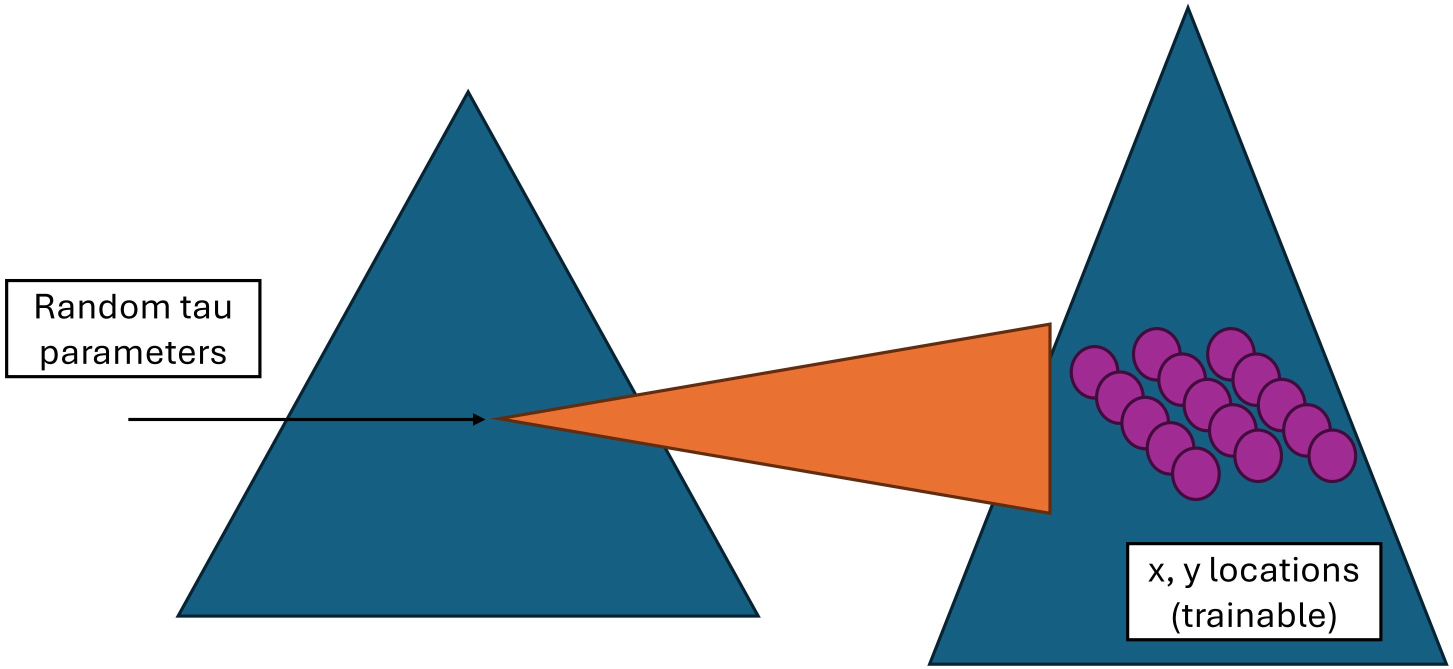
The TAMBO experiment

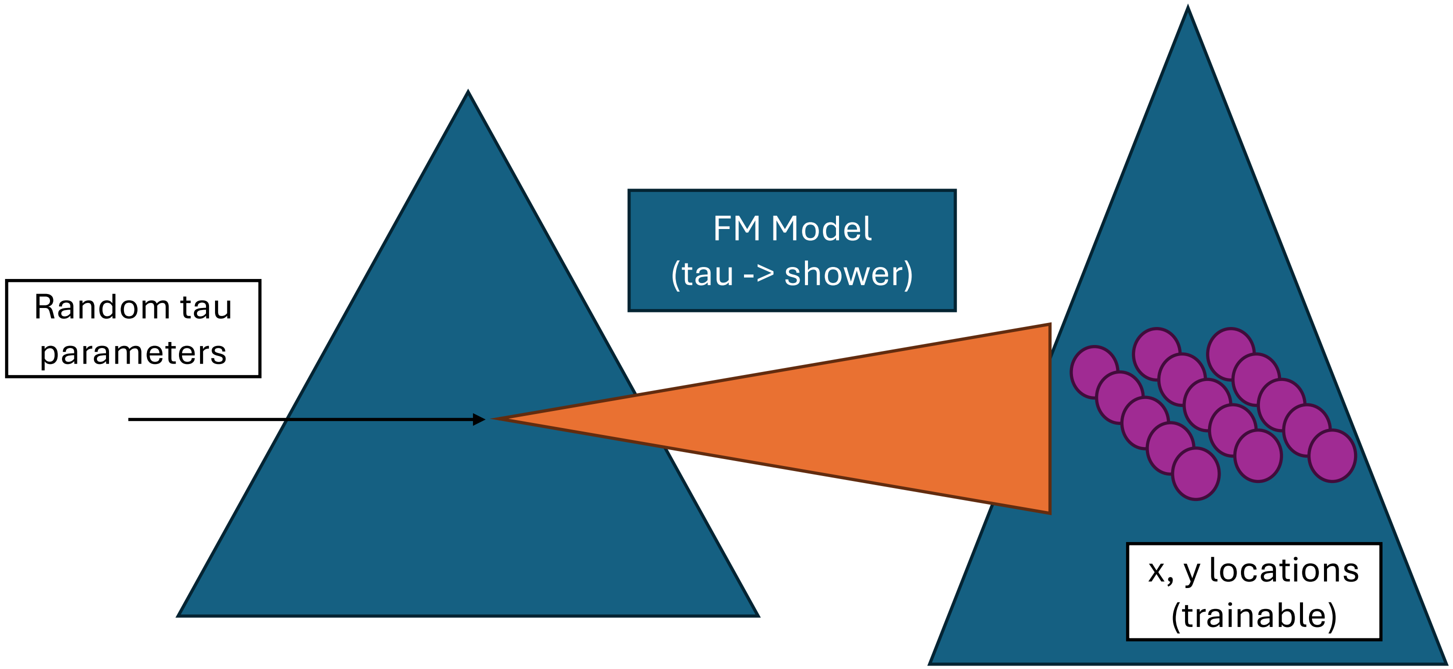
- TAMBO (Tau Air Shower Mountain-Based Observatory)
- Earth-skimming tau neutrinos
- Detection of particle showers resulting from **tau leptons interactions** with the **dense rock** in the valley
- 100 detection units, which need to be positioned efficiently on the walls of the Colca Valley in the Peruvian Andes.
- This presentation focuses on the **use of surrogate models and differentiable programming** in the optimization of the detector array.



TAU AIR-SHOWER MOUNTAIN-BASED OBSERVATORY (TAMBO) • COLCA VALLEY, PERU

[1] Pavel Zhelnin, Ibrahim Safa, Andres Romero-Wolf, and Carlos A. Argüelles. TAMBO: Searching for Astrophysical Tau Neutrinos in the Andes . PoS ICHEP2022 566.
[2] “IceCube Neutrino Observatory Home,” IceCube, Dec. 08, 2025. https://icecube.wisc.edu/


Random tau
parameters
x, y locations
(trainable)


FM Model
(tau -> shower)
Random tau
parameters
x, y locations
(trainable)

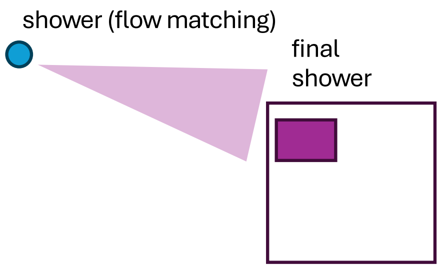
Flow-matching based surrogate model

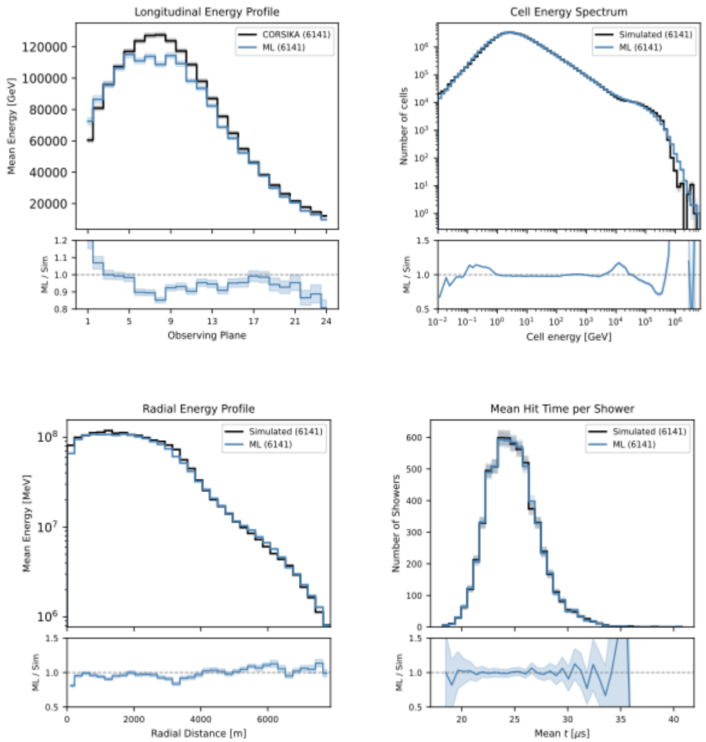
Architecture

- Parallel models for secondary species: proton, muon, electron
- Based on the AllShowers architecture [1]
    - transformer-based with block masks for custom attention window sizes,
    - generative architecture is based on flow matching
- Trained on 50k electromagnetic and 50k hadronic CORSIKA simulations

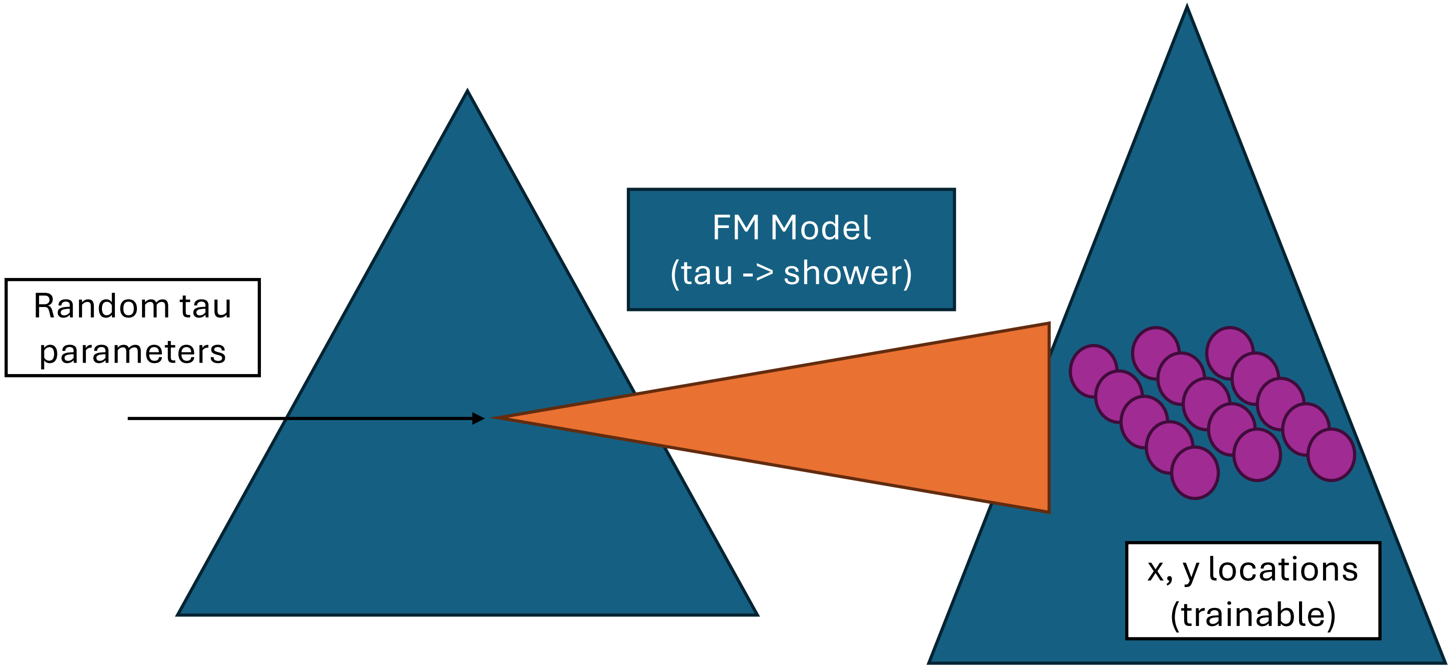
Output

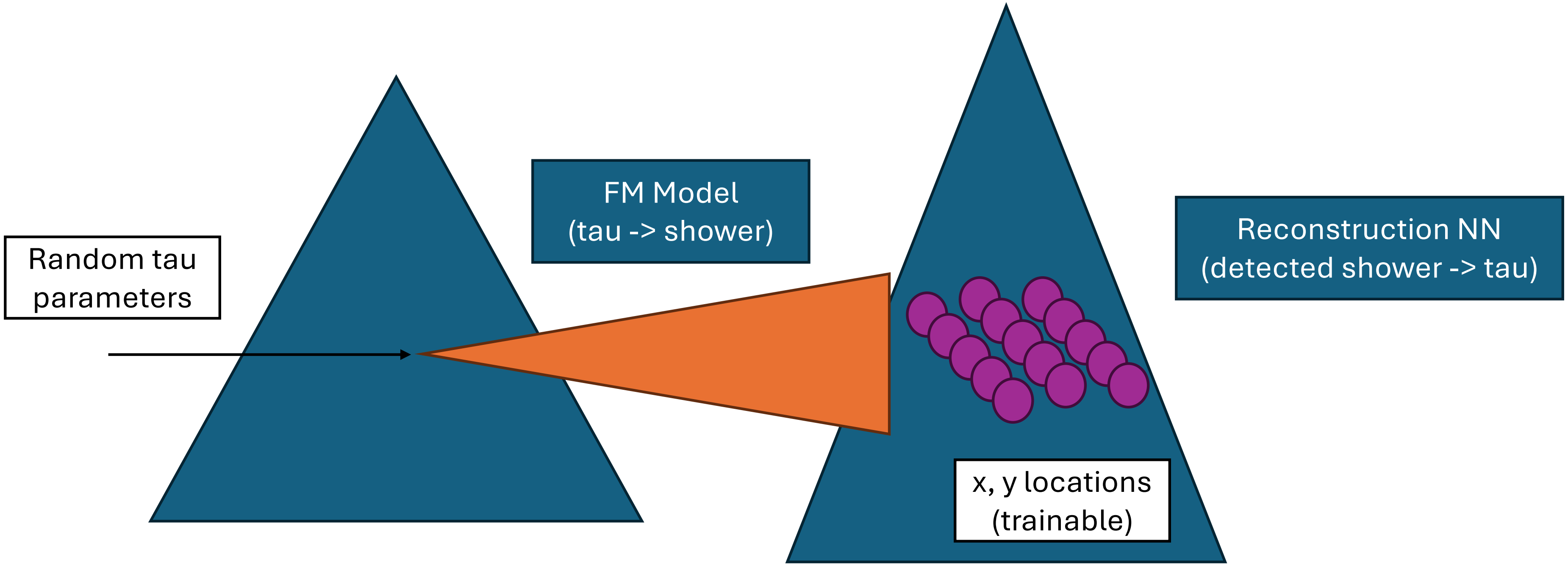
- Particle cloud representing the shower's secondary particles
    - x, y, z coordinates; Energy and time



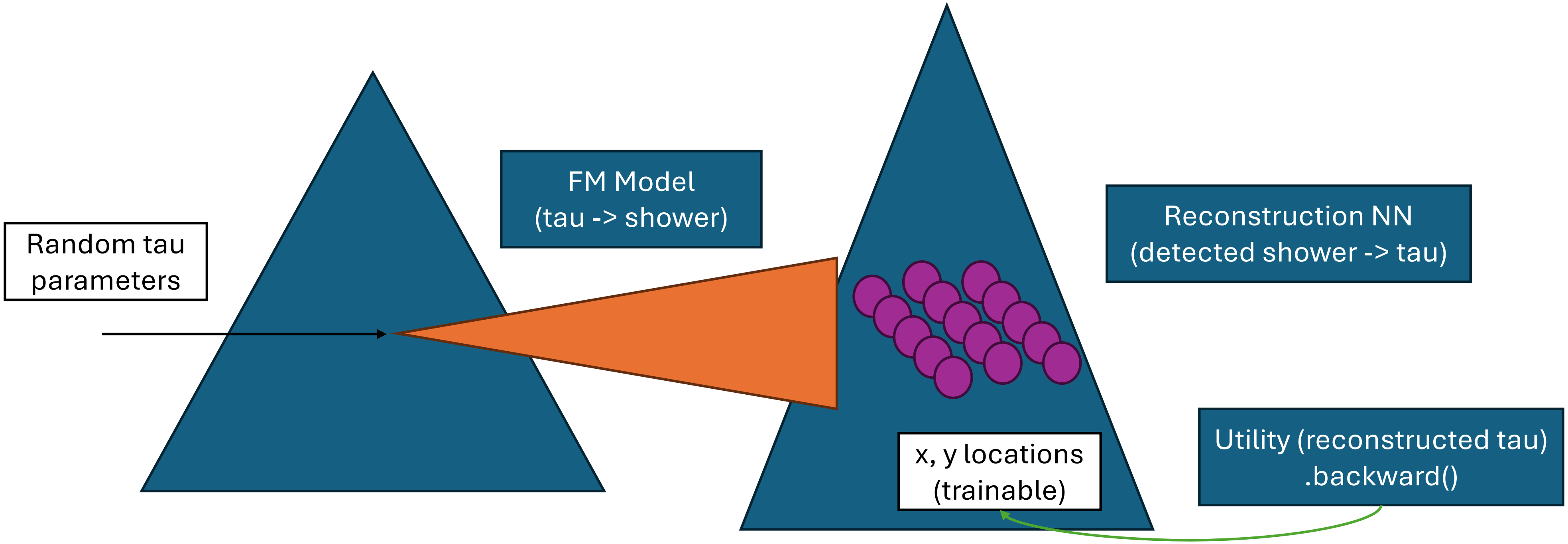


[1] T. Buss, H. Day-Hall, F. Gaede, G. Kasieczka, and K. Krüger, "AllShowers: One model for all calorimeter showers," *arXiv.org*, Jan. 16, 2026. https://arxiv.org/abs/2601.11716


Random tau
parameters
FM Model
(tau -> shower)
x, y locations
(trainable)



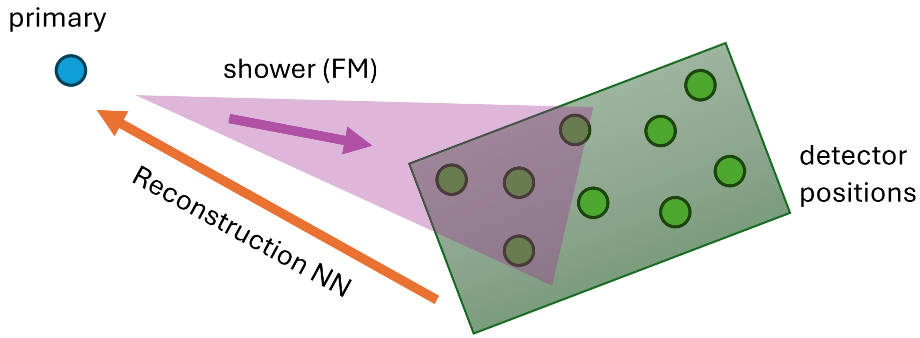
Random tau parameters
FM Model
(tau -> shower)
Reconstruction NN
(detected shower -> tau)
x, y locations
(trainable)

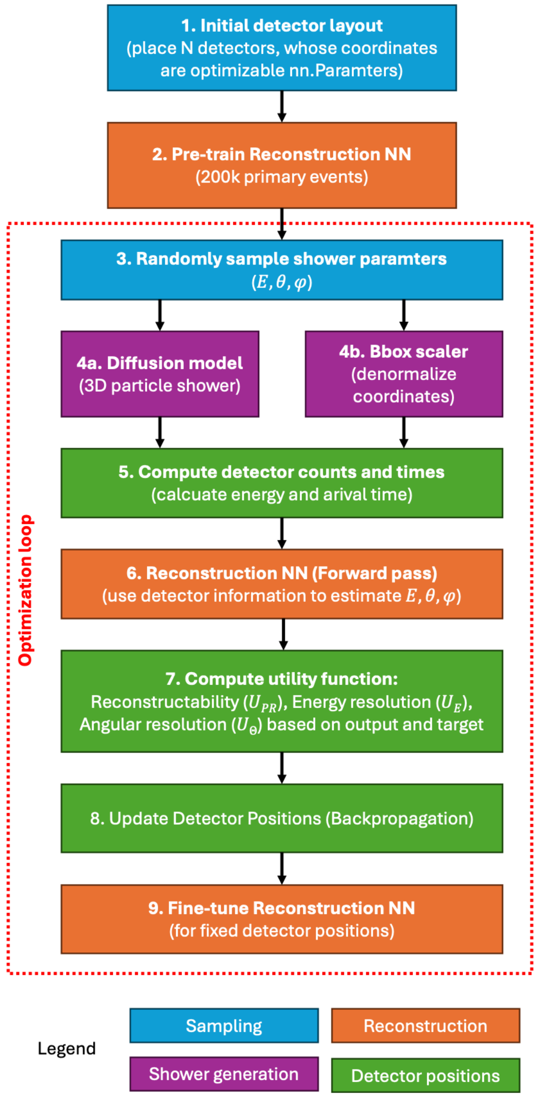

FM Model
(tau -> shower)
Random tau
parameters
Reconstruction NN
(detected shower -> tau)
x, y locations
(trainable)
Utility (reconstructed tau)
.backward()

Detector location optimization (SWGOLO)

Pre-train Reconstruction NN on 200k primary events
Loop:

- Use NN to reconstruct primary features from secondary events
- Calculate utility (U) for 500 events
- Update detector positions vie backpropagation
- **Every 5 epochs:**
    - fine-tune the Reconstruction NN on 5k samples

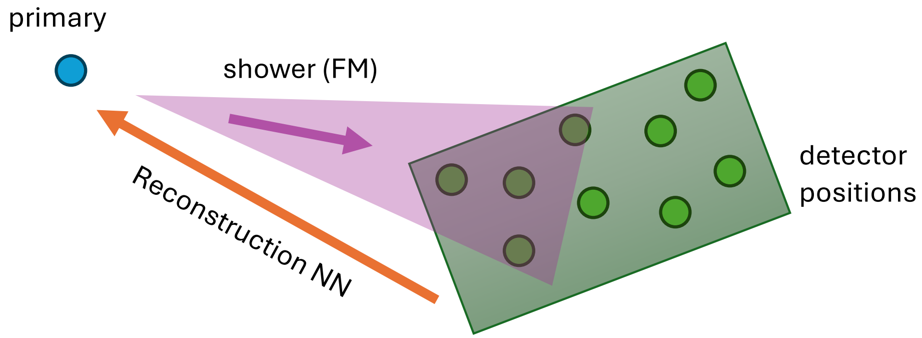


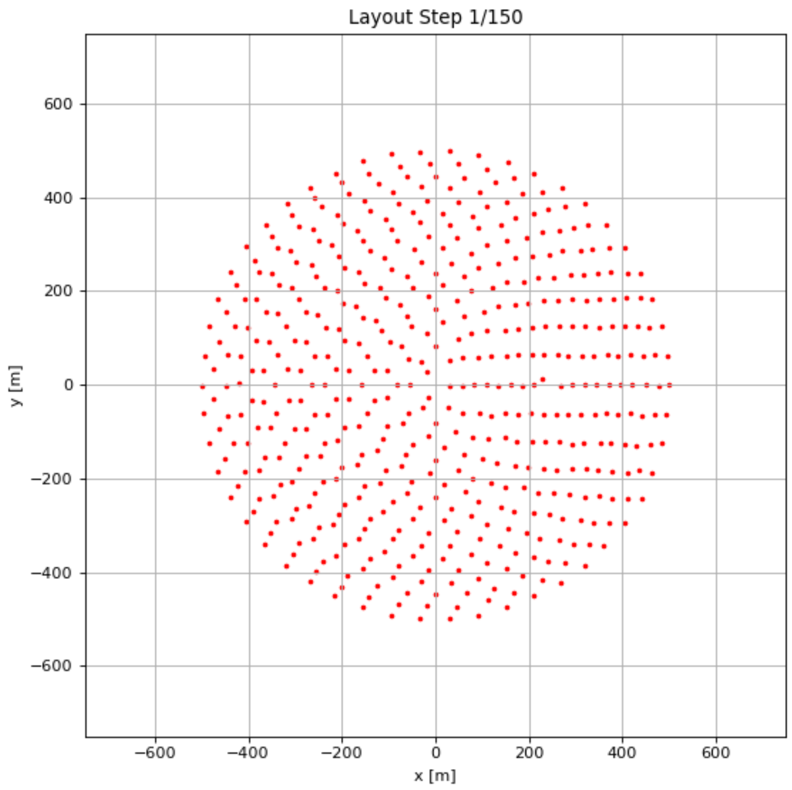


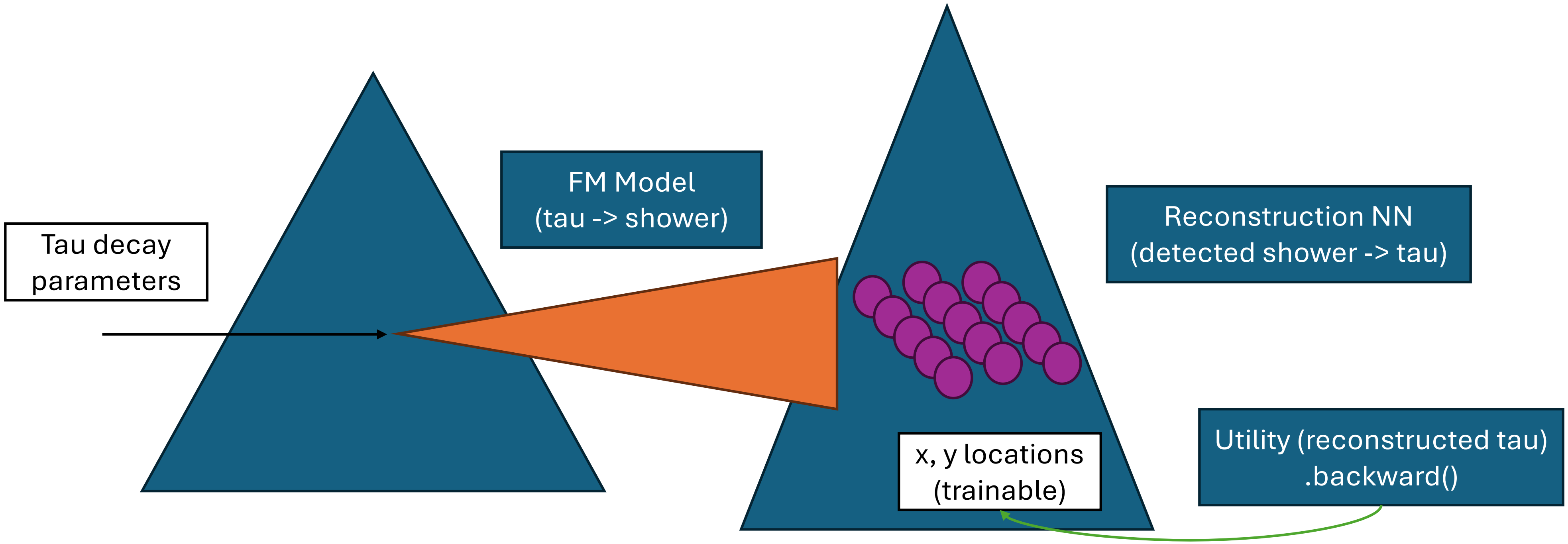
[1] Noyan, U. , “Optimization of the SWGO array with gradient descent algorithms,” *GitHub*. https://github.com/tdorigo/SWGOLO/tree/main

Stuck gradients problem

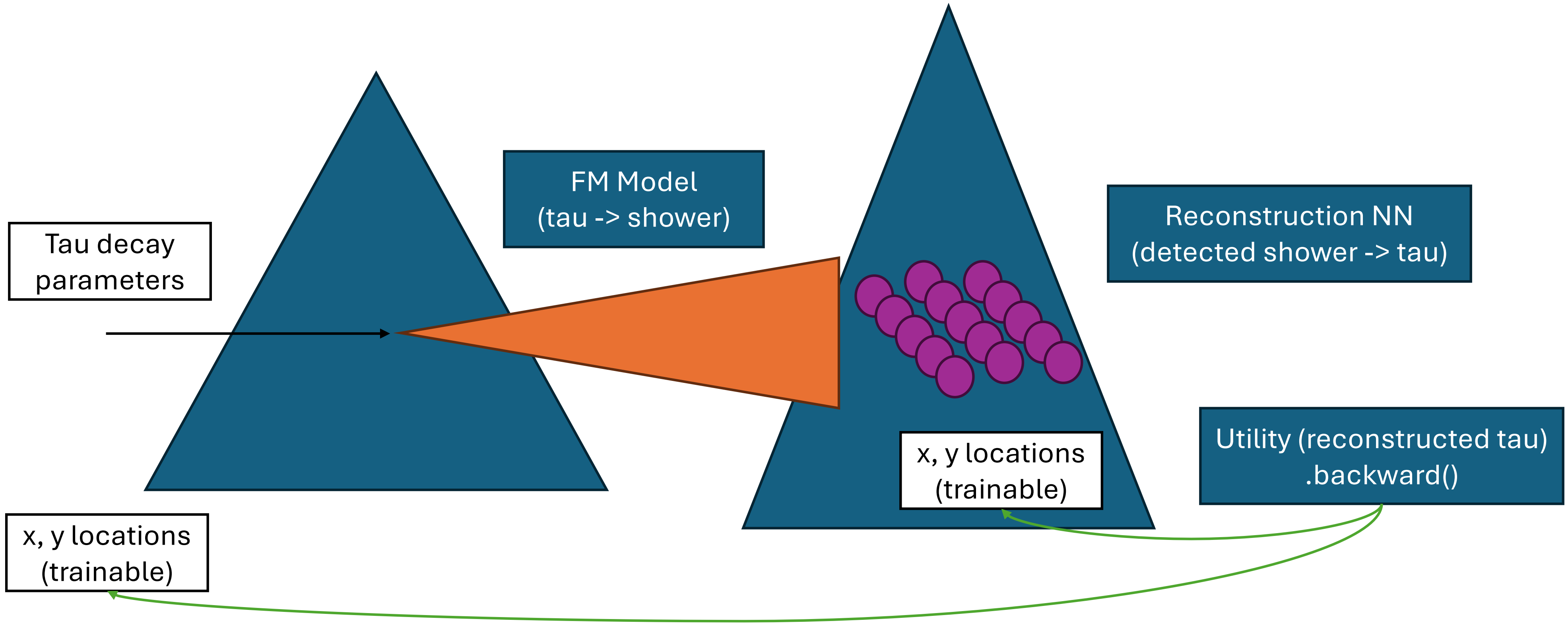
- Currently the locations update
- However, those updates are not well directed and the overall utility stays low

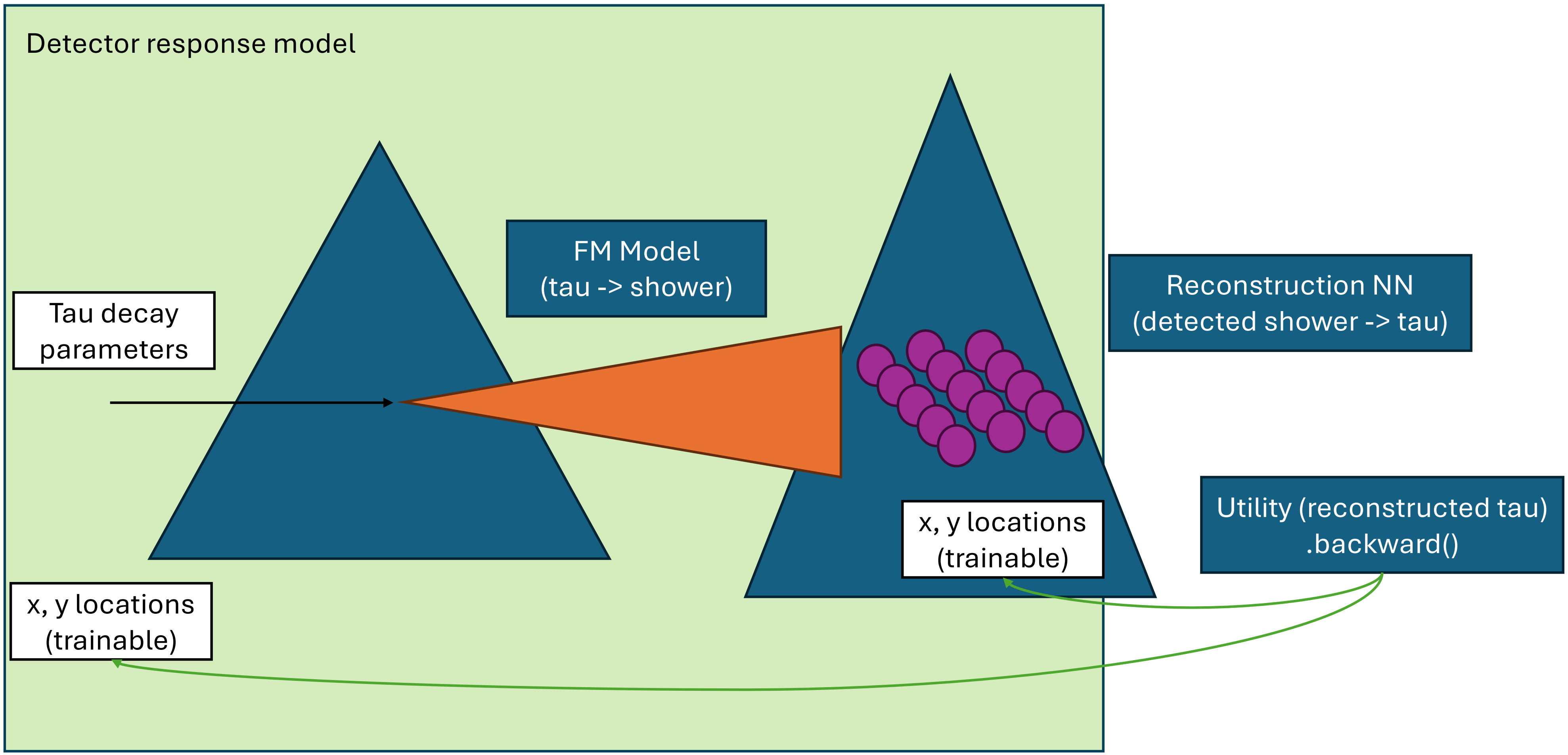


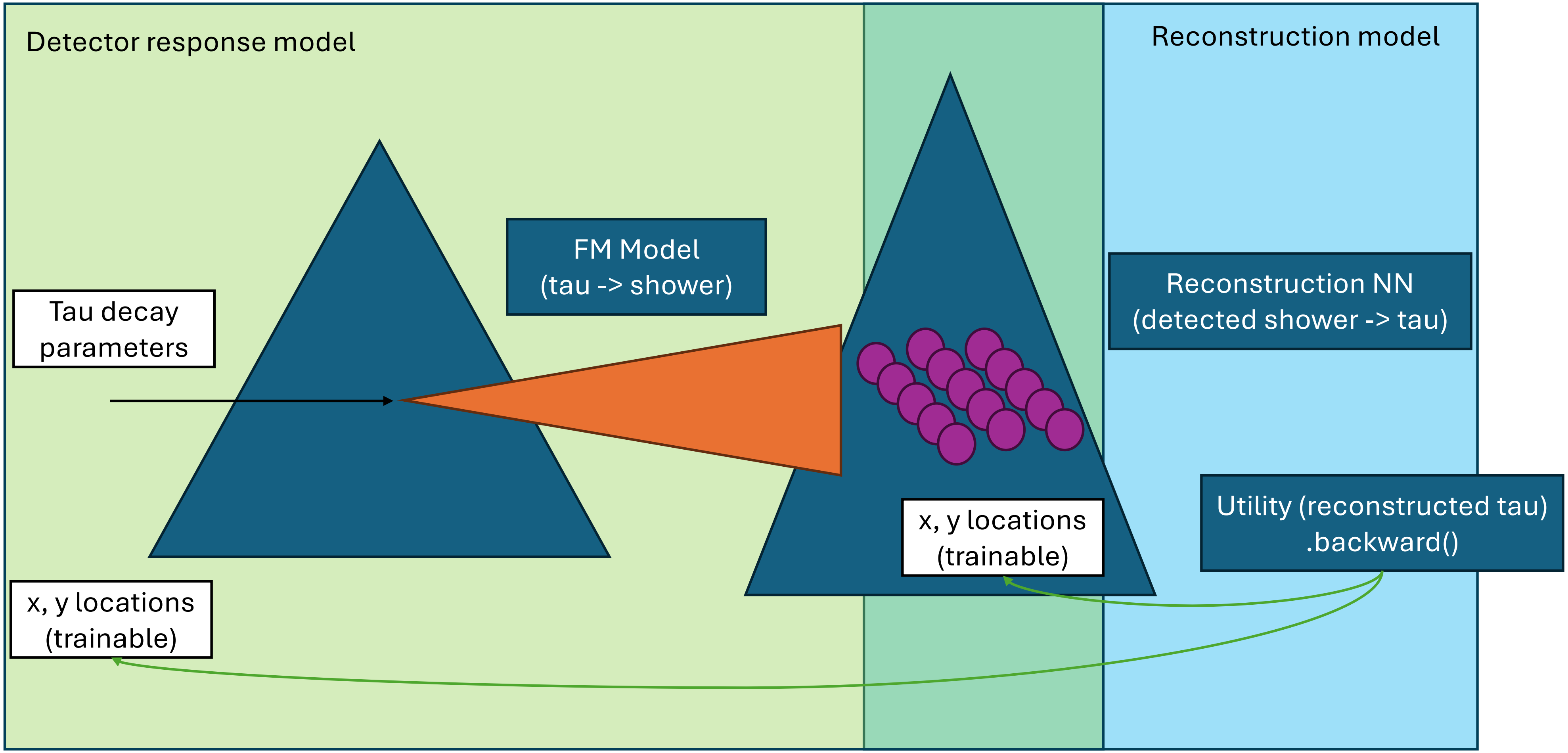


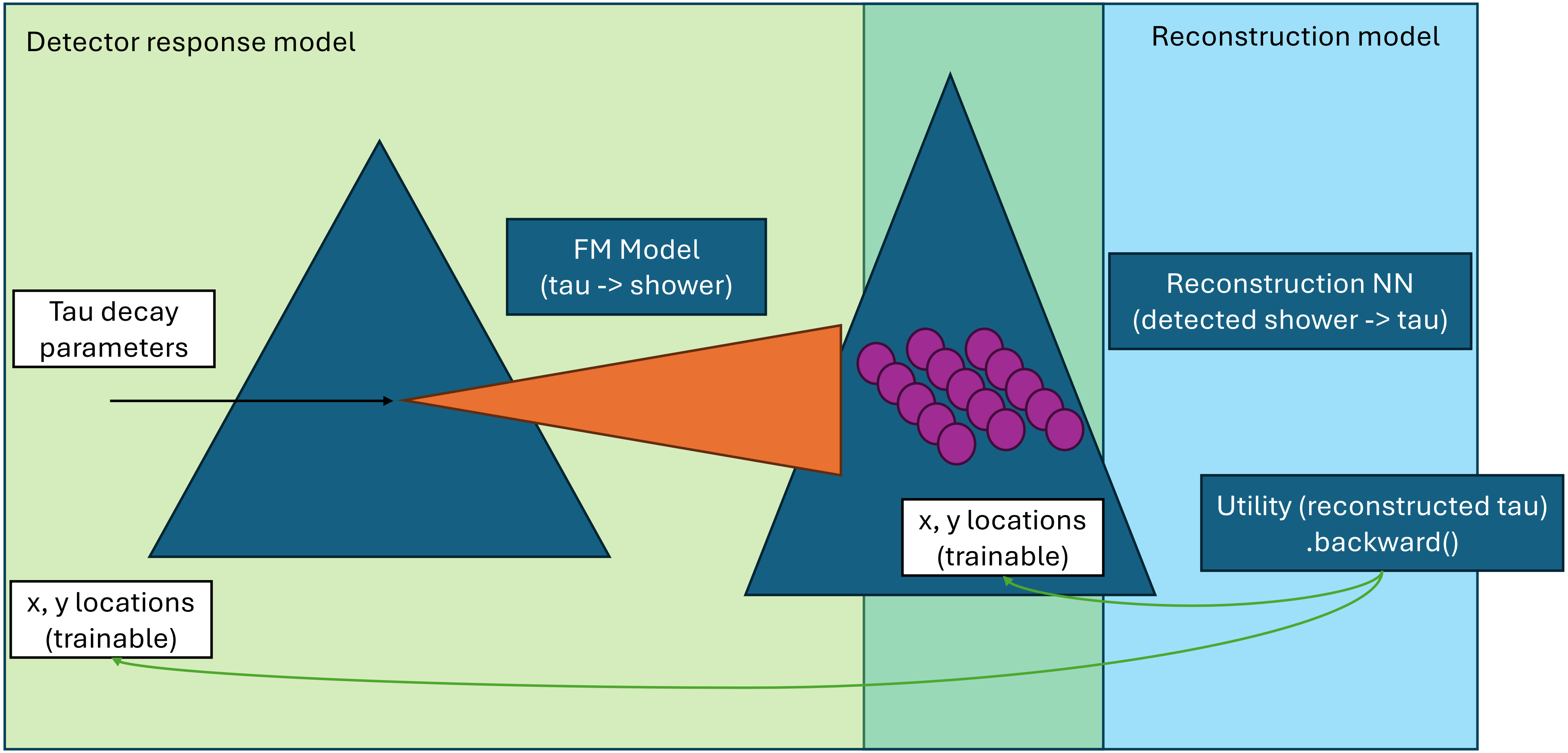


Tau decay parameters
FM Model
(tau -> shower)
Reconstruction NN
(detected shower -> tau)
x, y locations
(trainable)
Utility (reconstructed tau)
.backward()


FM Model
(tau -> shower)
Reconstruction NN
(detected shower -> tau)
Tau decay
parameters
x, y locations
(trainable)
Utility (reconstructed tau)
.backward()
x, y locations
(trainable)


Detector response model
FM Model
(tau -> shower)
Tau decay
parameters
Reconstruction NN
(detected shower -> tau)
x, y locations
(trainable)
Utility (reconstructed tau)
.backward()
x, y locations
(trainable)

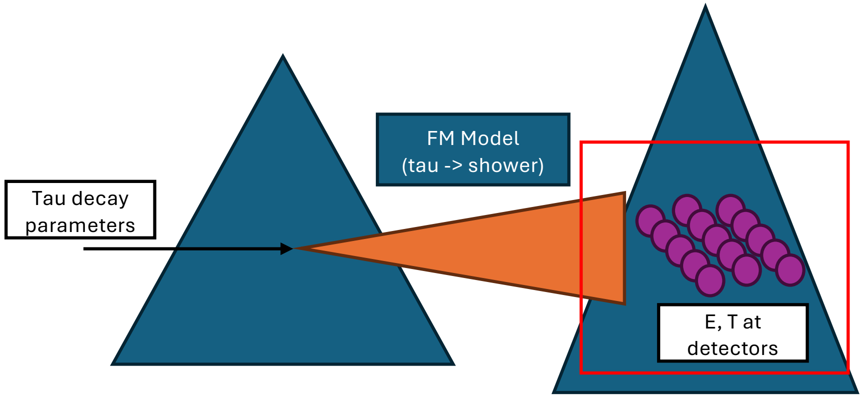

Detector response model
Reconstruction model
FM Model
(tau -> shower)
Tau decay
parameters
Reconstruction NN
(detected shower -> tau)
Utility (reconstructed tau)
.backward()
x, y locations
(trainable)
x, y locations
(trainable)



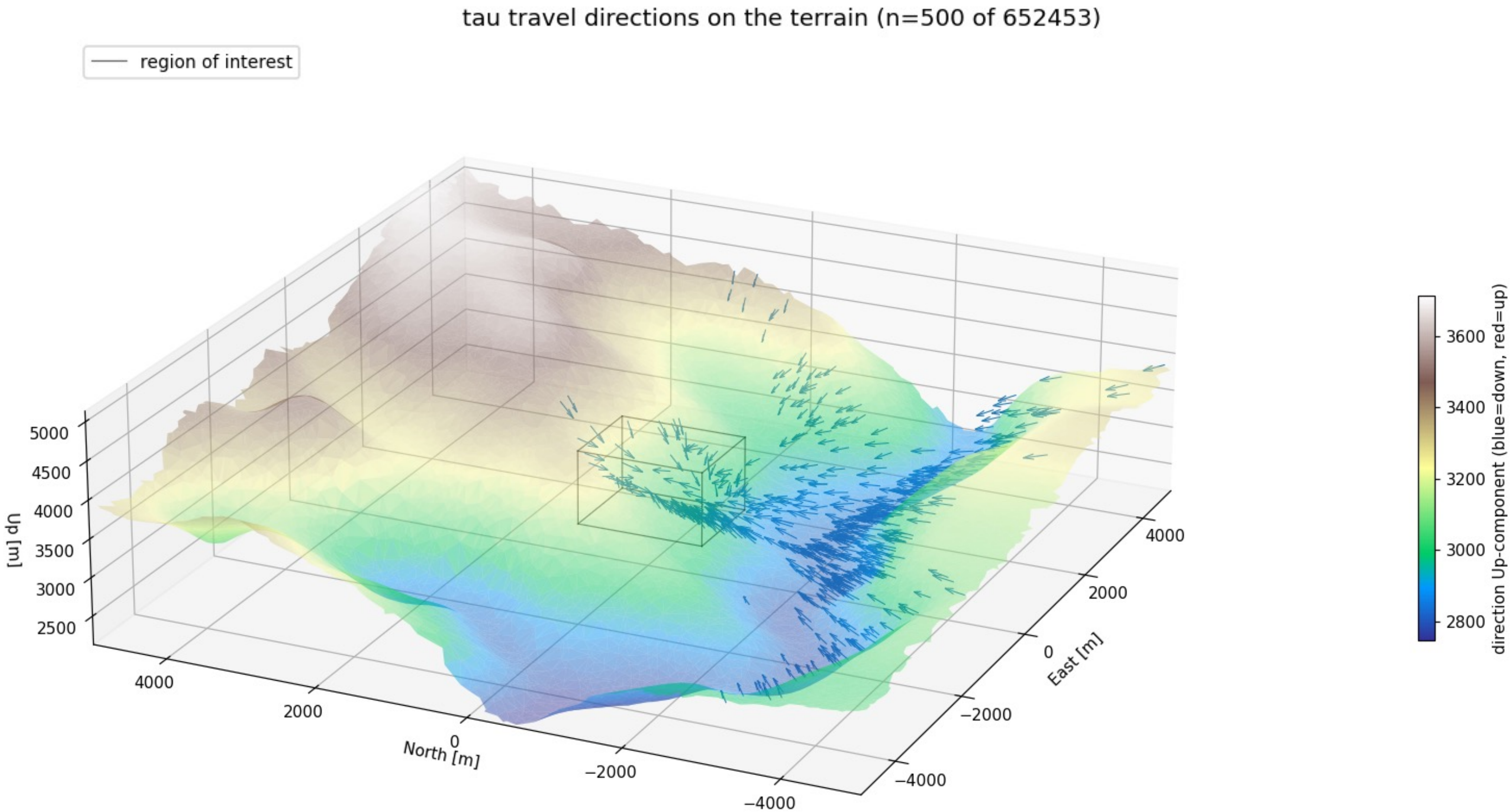
[1] X. T. Nguyen et al., “Differentiable Surrogate for Detector Simulation and Design with Diffusion Models,” arXiv.org, Jan. 09, 2026. https://arxiv.org/abs/2601.07859v1

Stage 1: Data preparation

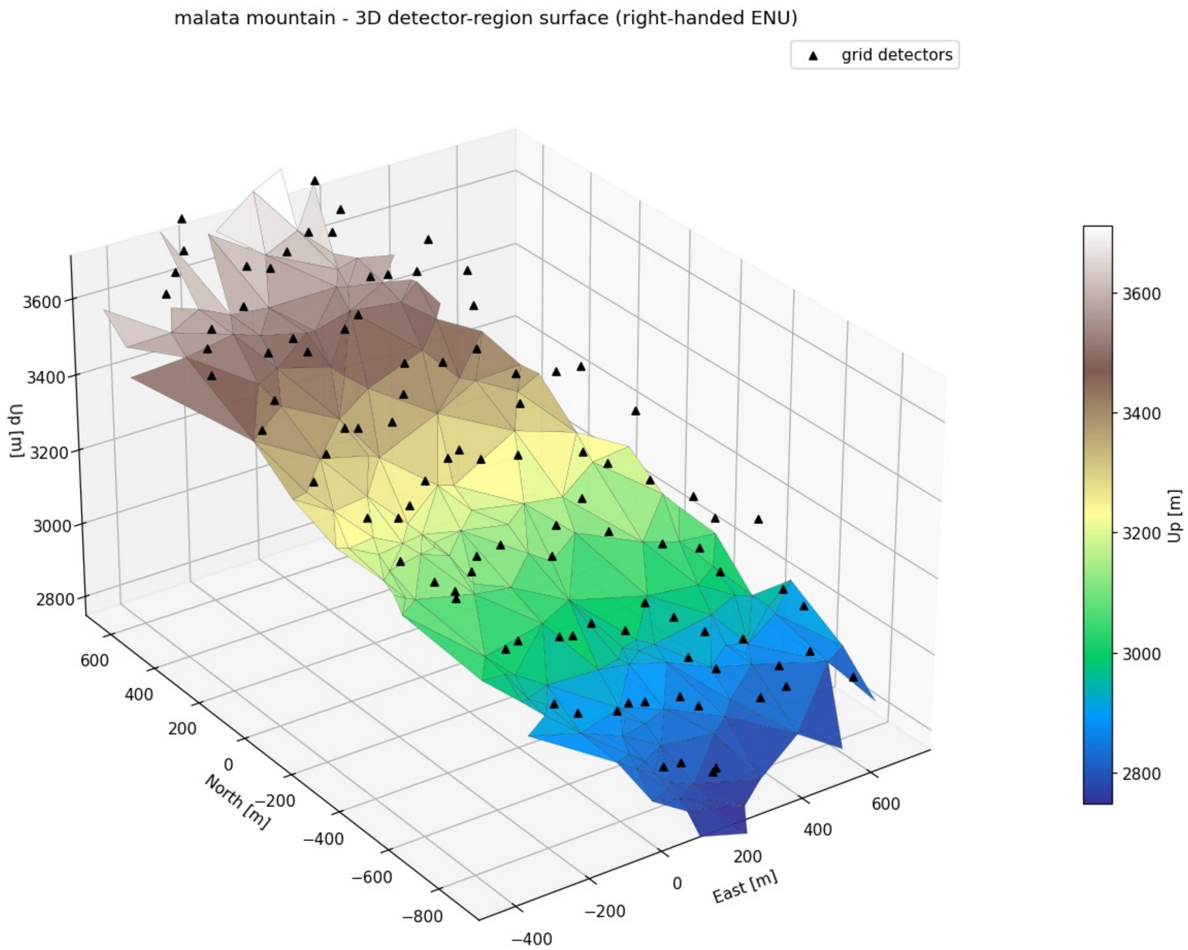
- Prepare 750k tau decay locations + energies
- Trigger surrogate model to generate a space shower, based on the
    - Direction of the primary particle (nx, ny, nz)
    - Energy of primary particle
- This shower is then shifted to the primary decay location
- Match each shower with 5 detector layouts sampled randomly using Gaussian, uniform, lattice hypercube sampling



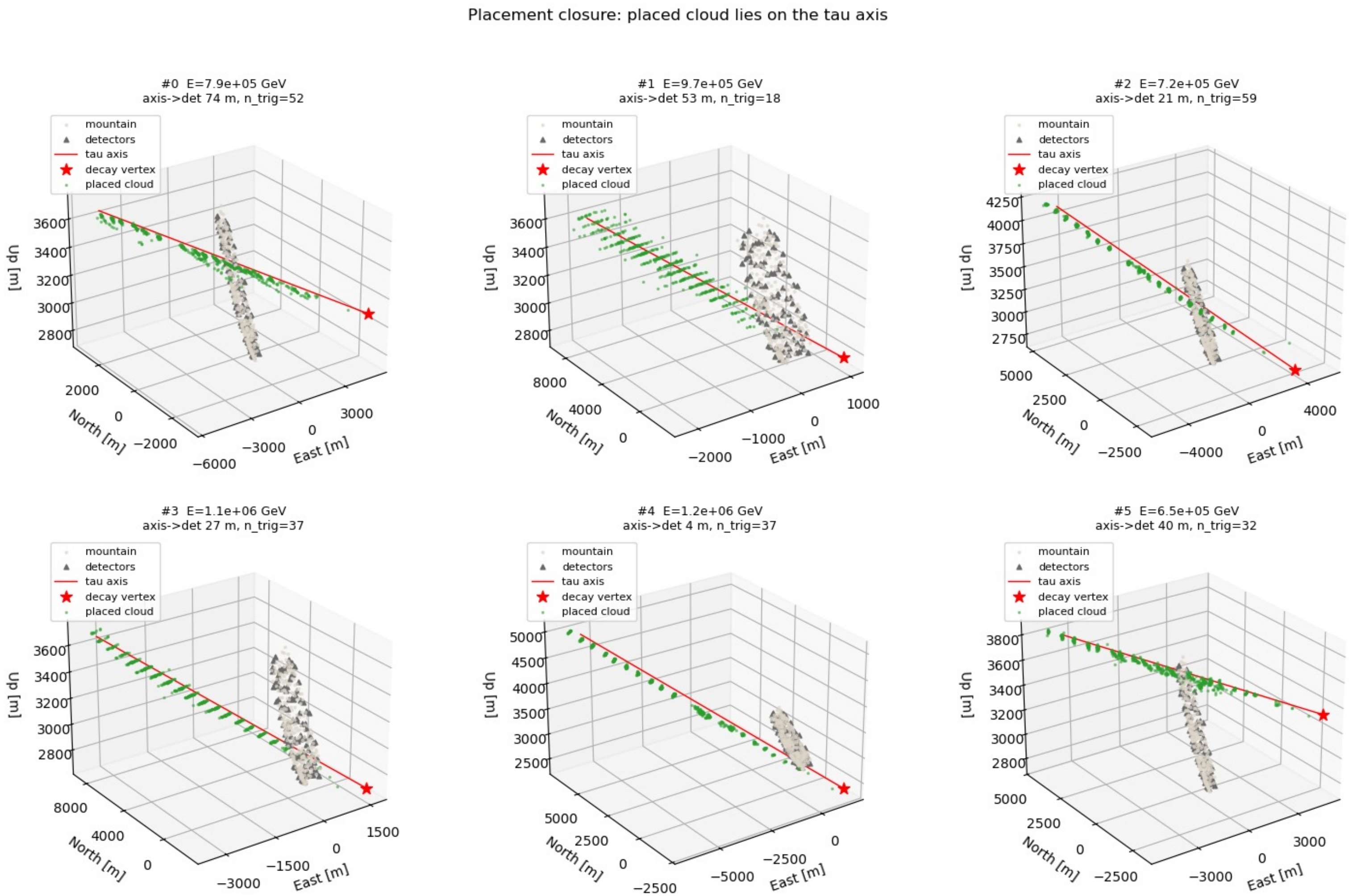
Stage 1: Data preparation – tau directions



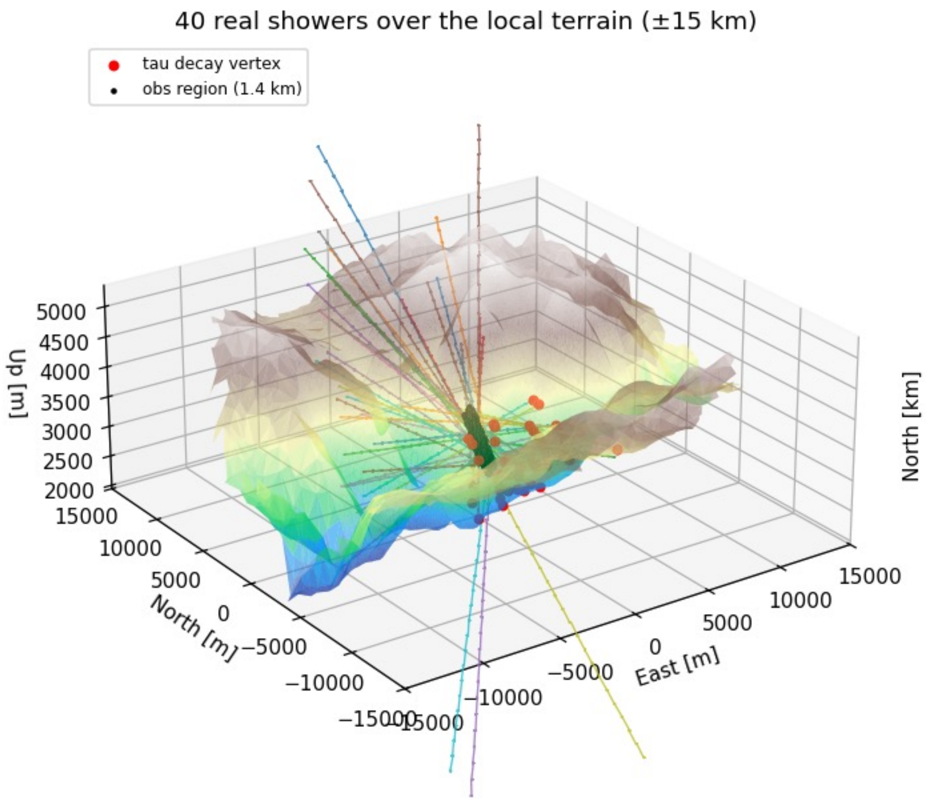
Stage 1: Data preparation – detector locations

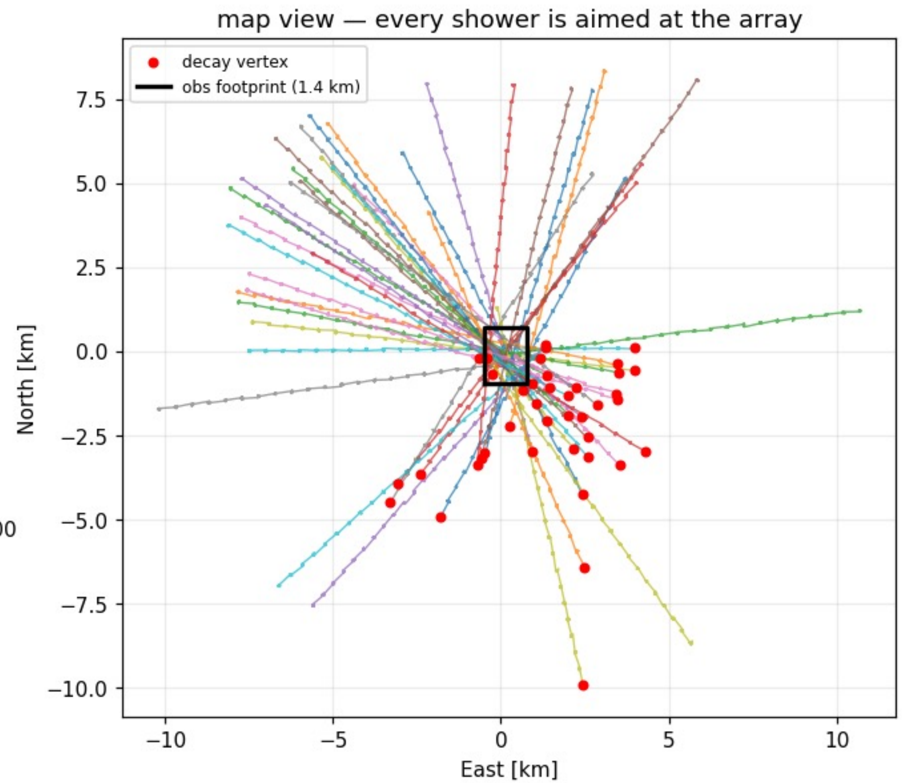


Stage 1: Data preparation – shower placement

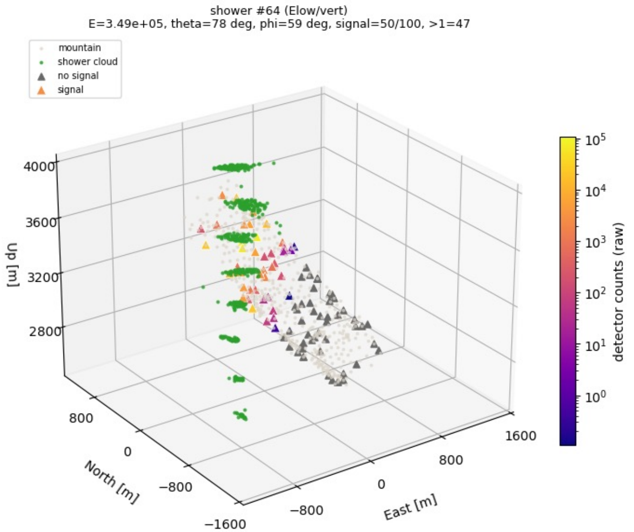


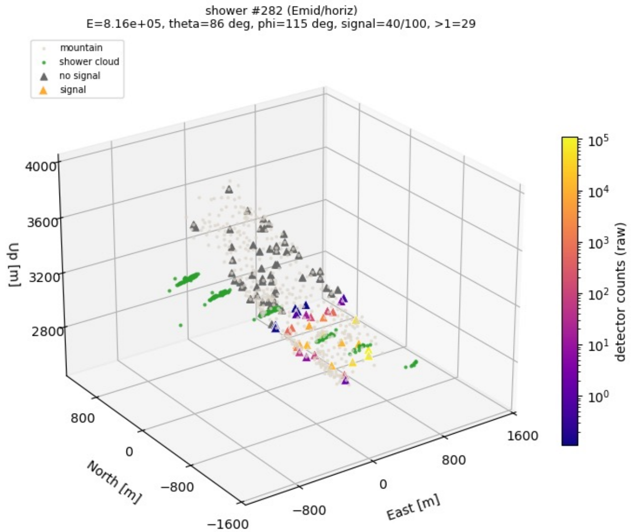
Stage 1: Data preparation – shower placement (2)

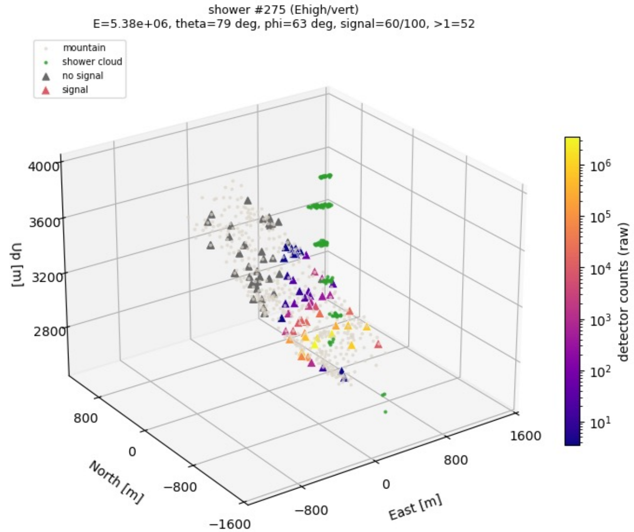




Stage 1: Data preparation – detector activation

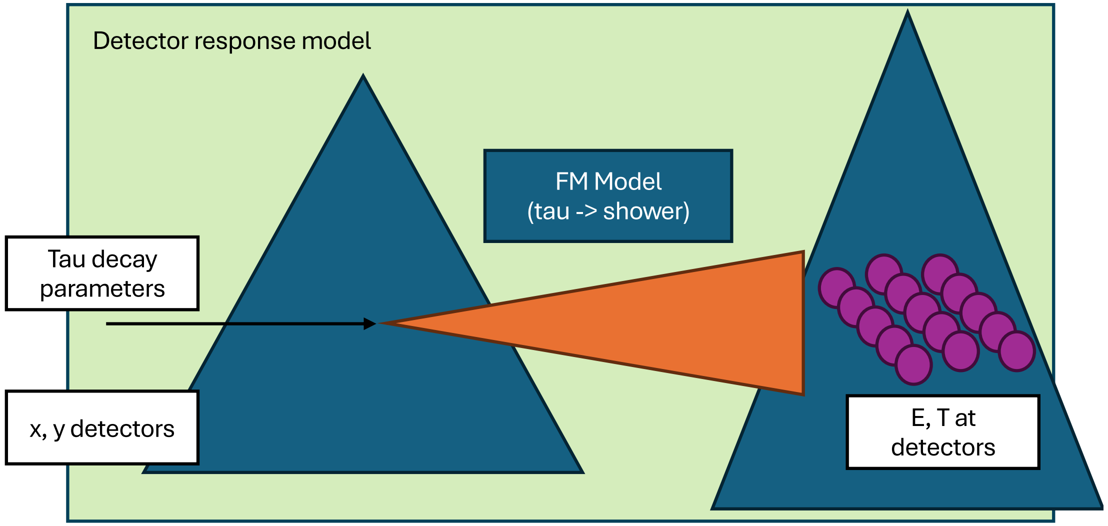


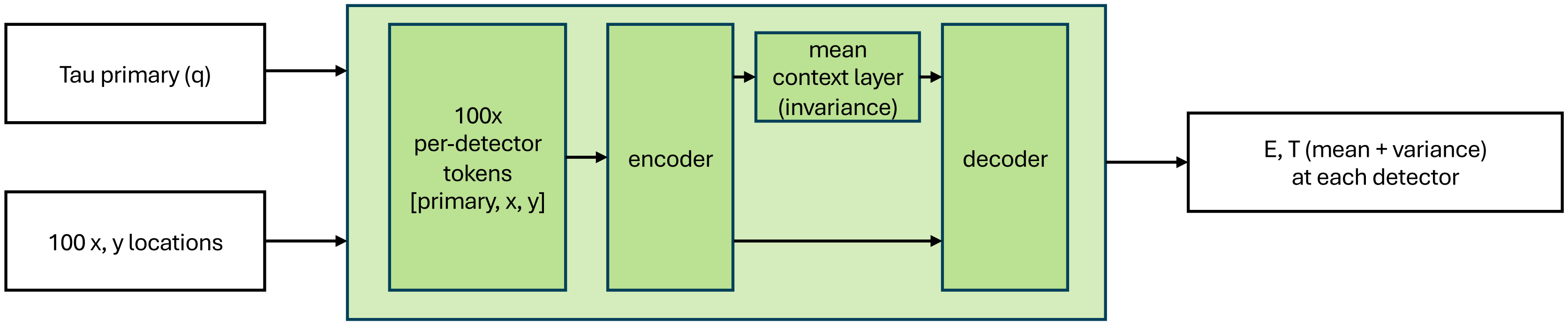




Stage 2: Detector response

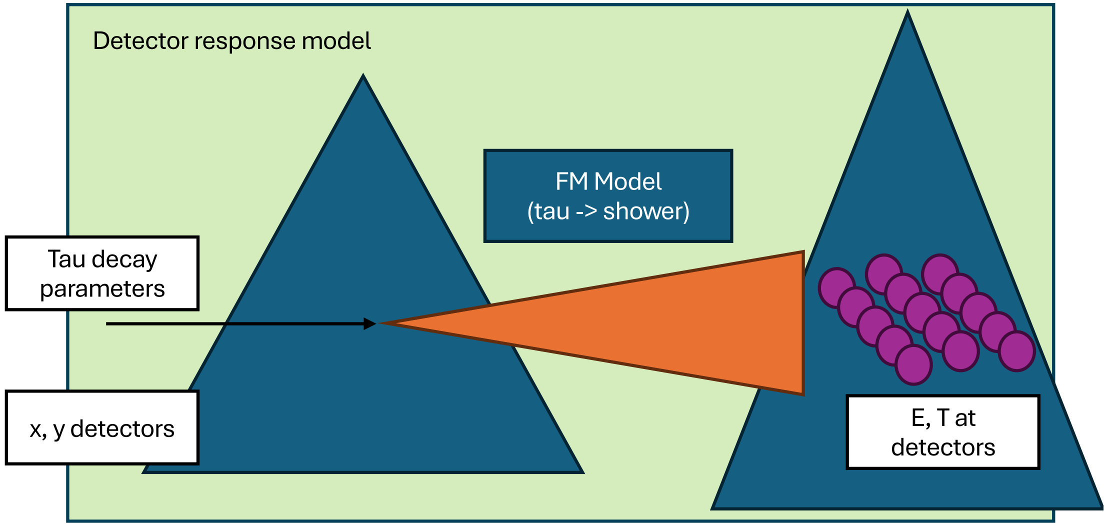
- Deepsets NN to get detector response
- Inputs:
  - x, y coordinates of detectors,
  - primary particle characteristics
- Output
  - E and T at each detector
- Detector locations permuted during training
- Parallel NNs for each particle type:
  - Outputs for each detector:
  - sum of energies
  - count weighted mean of arrival times



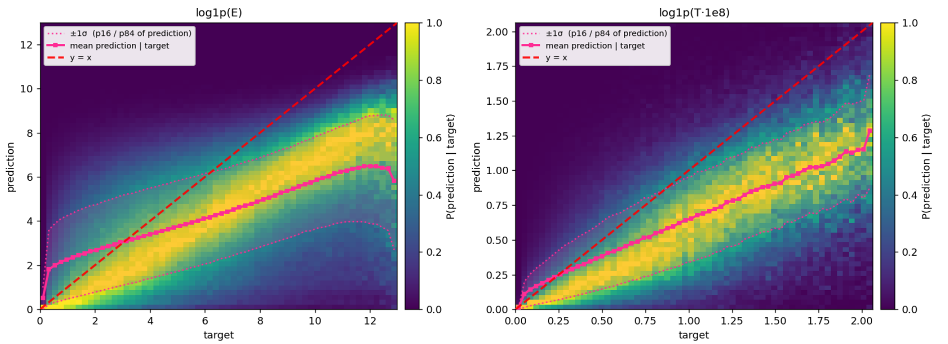


Stage 2: Detector response

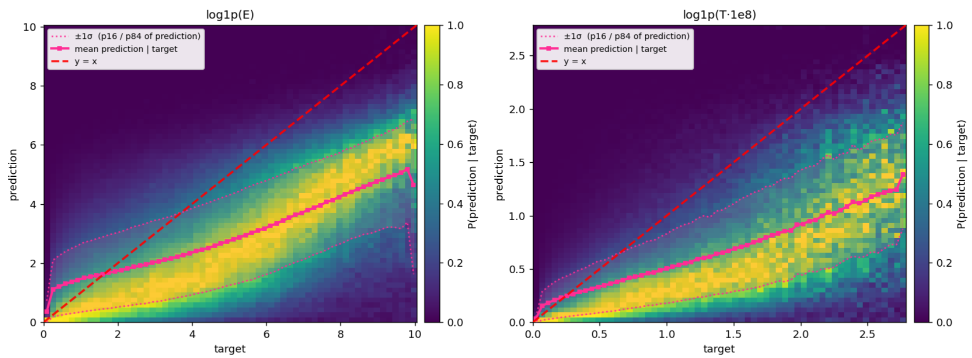
- Training error converges to the stochiasticity level in each shower
- Experiemnts are run with a mean + variance NN to achieve sochiastic behavior of the surrogate
- This approach would enable the further stages to be trained more robustly



Deepsets conditional density — hit detectors (79% of 28,608,000)

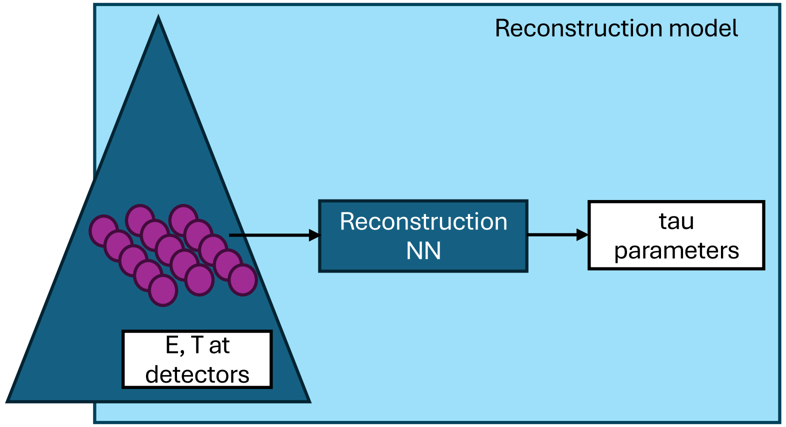


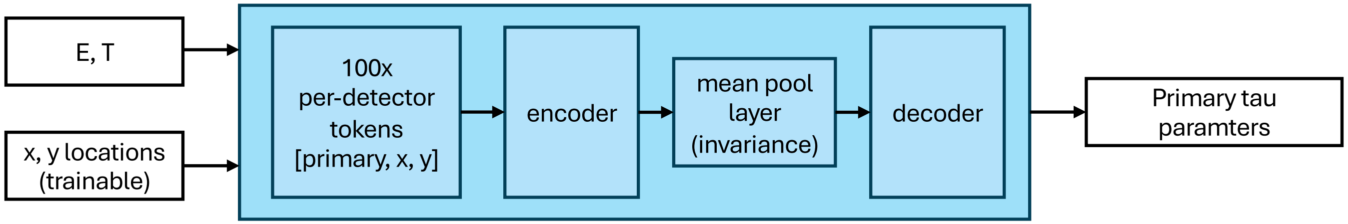
Deepsets conditional density — hit detectors (75% of 28,608,000)

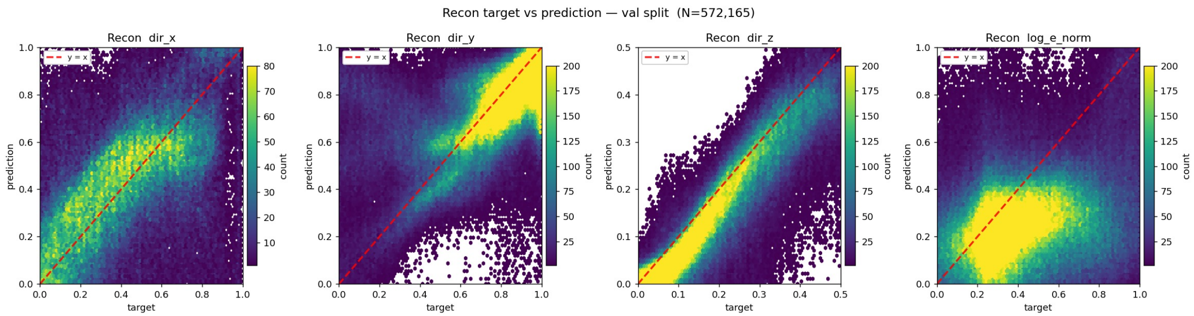


Stage 3: Reconstruction

- Input
  - Location, energy, time of arrival at each detector
- Output
  - Regenerated primary particle characteristics

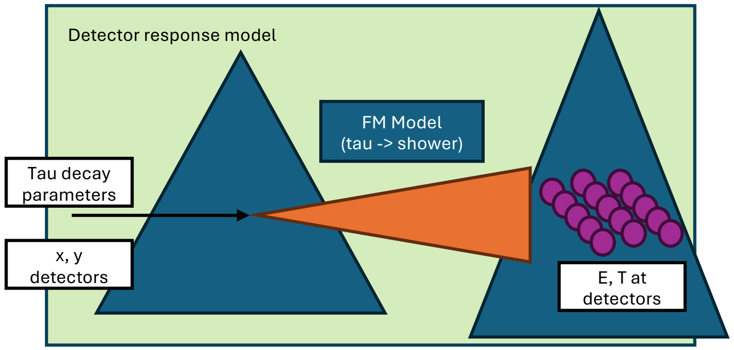






Stage 4: Detector array optimization – Normalized detector activation

- Utility
  - defined as total detected energy
  - normalized by detector overlap
  - Detectors represented as Gaussian in space with sigma=50m
  - Based only on the detector response model
- Setup
  - 2 chains initialized at different detector locations: center + grid
  - 100k primary particles
  - Move detector locations based on autodiff
  - Optimize first using Adam then LBFGS
- Imperfections in detector response model produces lower absolute numbers



$$U_{\text{distinct}} = \frac{1}{B}\sum_b \frac{1}{N_{\text{det}}}\sum_d \frac{c_{bd}}{m_d}, \tag{1}$$

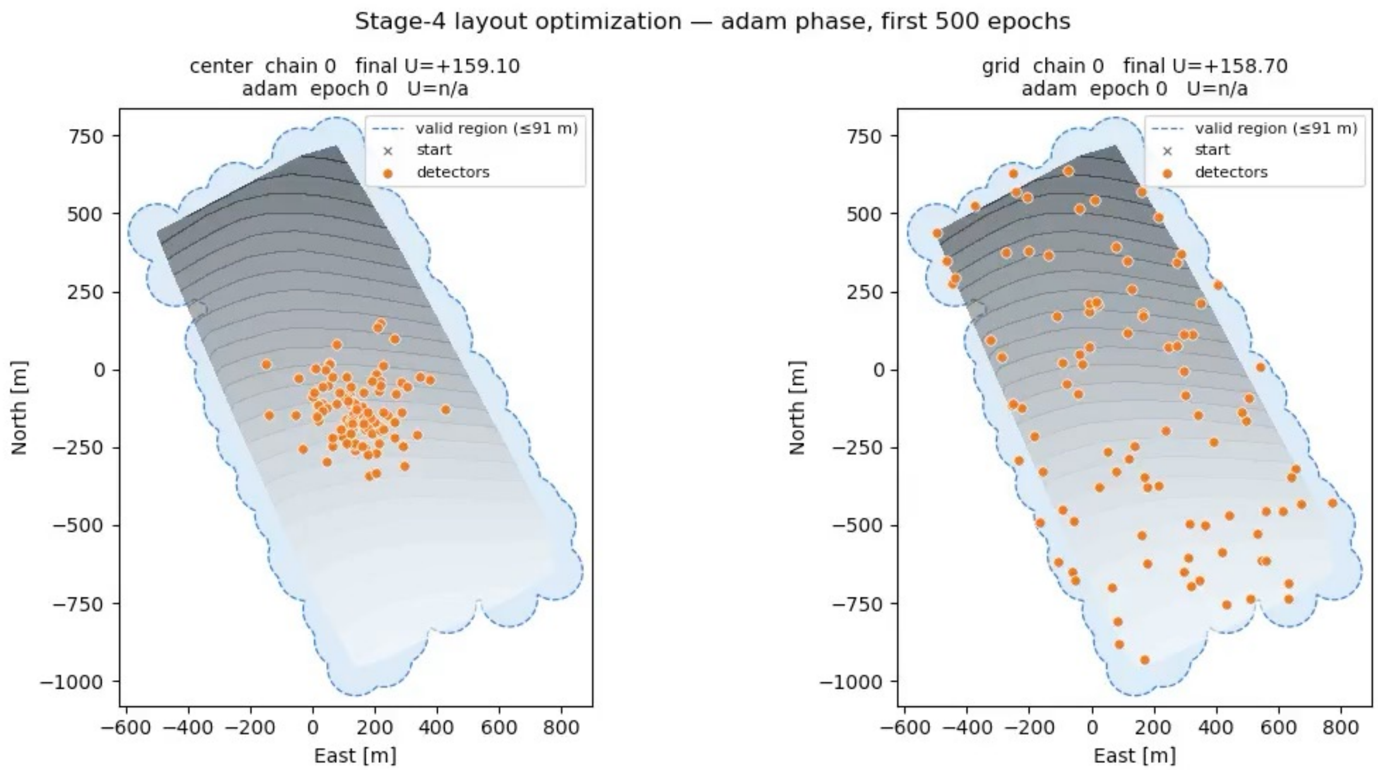
$$m_d = \sum_j \exp\left(-\frac{\|r_d - r_j\|^2}{2\sigma^2}\right), \tag{2}$$

$$\sigma = 50\ \text{m}, \quad \mathbf{r}_d = (x_d, y_d) \tag{3}$$

c_{bd} energy-weighted particle flux at detector d from shower b

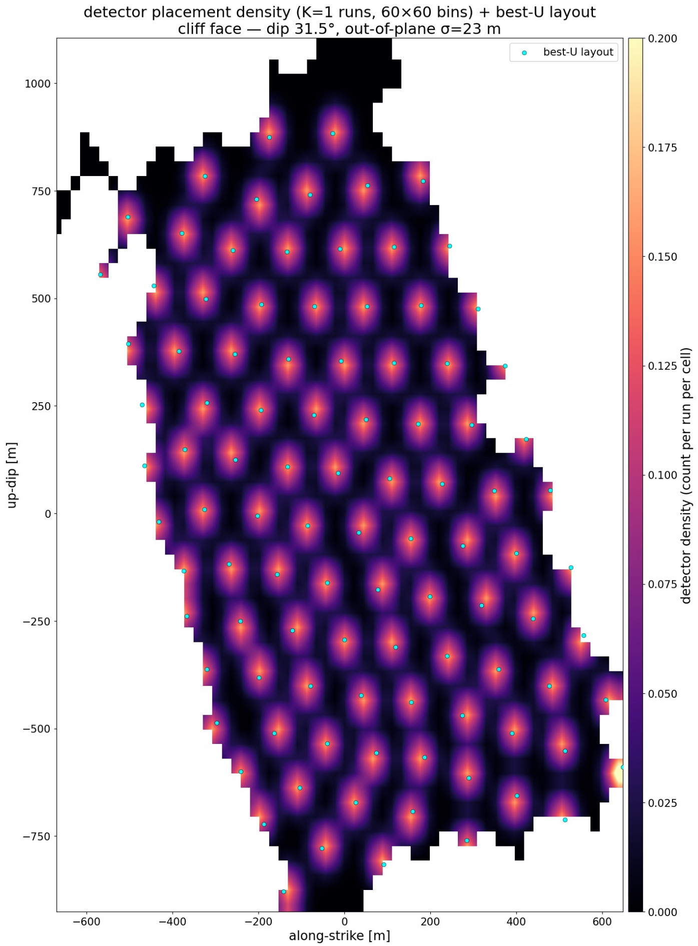
Utility (distinct)	Detector response model	Flow matching surrogate
Grid initialization		
Baseline grid	122.12	2497.12
Optimized	158.70	3240.04
Center initialization		
Baseline center	3.45	68.82
Optimized	159.08	3276.75

Stage 4: Detector array optimization – Composite utility (2)

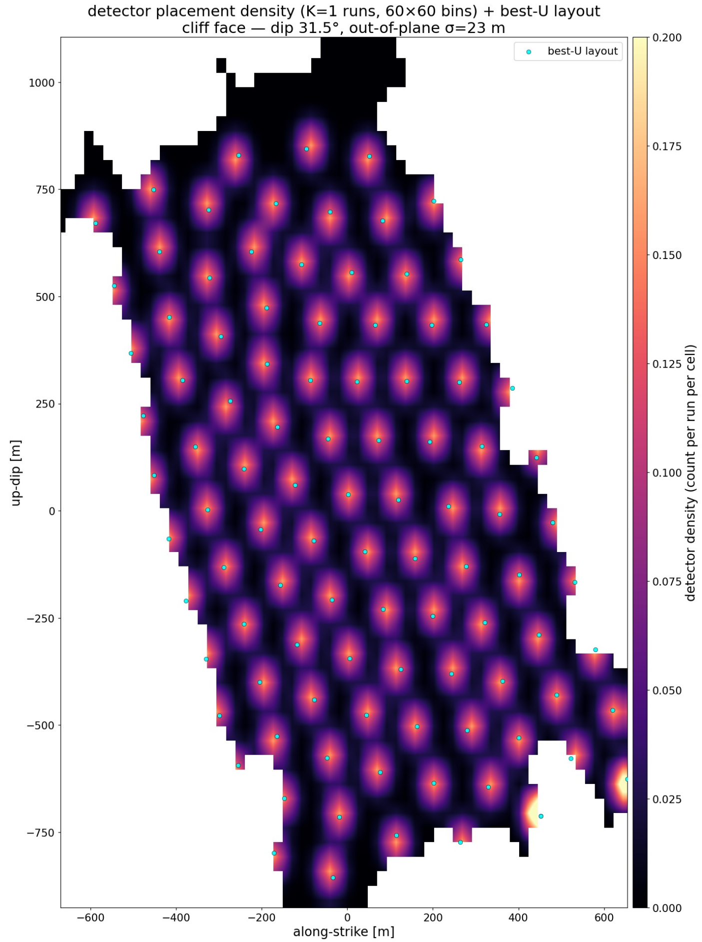


Stage 4: Detector array optimization – Composite utility (3)

Centre initialization

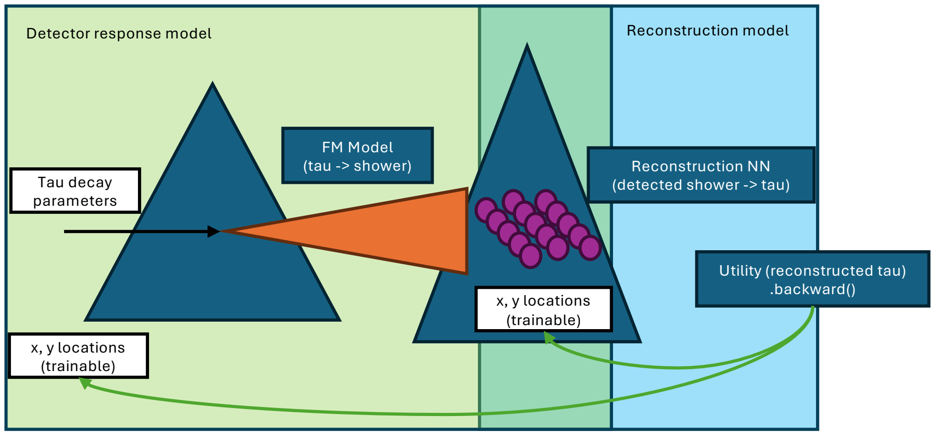


Grid initialization



Stage 4: Detector array optimization – Composite utility

- Use 2x6 chains initialized at different detector locations
- 100k primary particles
- Move detector locations based on autodiff
- Optimize first using Adam then LBFGS
- Verdict
  - Successfully optimizes the detector array on the deepsets model
  - There is a discrepancy to the unseen stochastic showers



Utility function

$$r_i = \sigma\left(\sum_j \sigma\left(\log_{10} \hat{E}_i - \lambda_{\text{layout}}\right) - \lambda_{\text{recon}}\right) \quad (1)$$

$$\mathcal{U}_E = \frac{1}{B}\sum_i \frac{r_i}{\left(\log_{10}\hat{E}_i - \log_{10} E_i\right)^2} \quad (2)$$

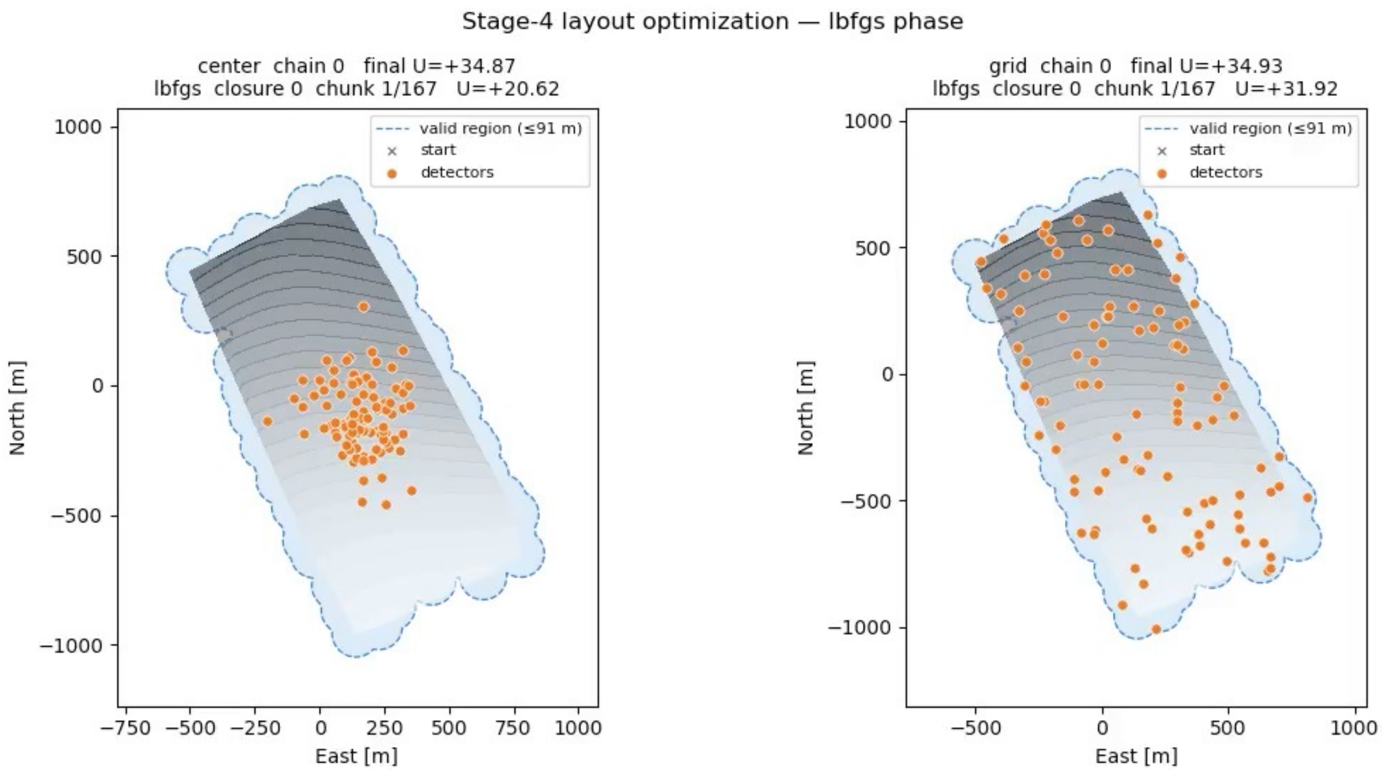
$$\mathcal{U}_\theta = \frac{1}{B}\sum_i \frac{r_i}{\left(\hat{\theta}_i - \theta_i\right)^2} \quad (3)$$

$$\mathcal{U}_\phi = \frac{1}{B}\sum_i \frac{r_i}{\left(\hat{\phi}_i - \phi_i\right)^2} \quad (4)$$

$$\mathcal{U} = \frac{w_\theta \mathcal{U}_\theta + w_\phi \mathcal{U}_\phi + w_E \mathcal{U}_E}{w_{\text{div}}} \quad (5)$$

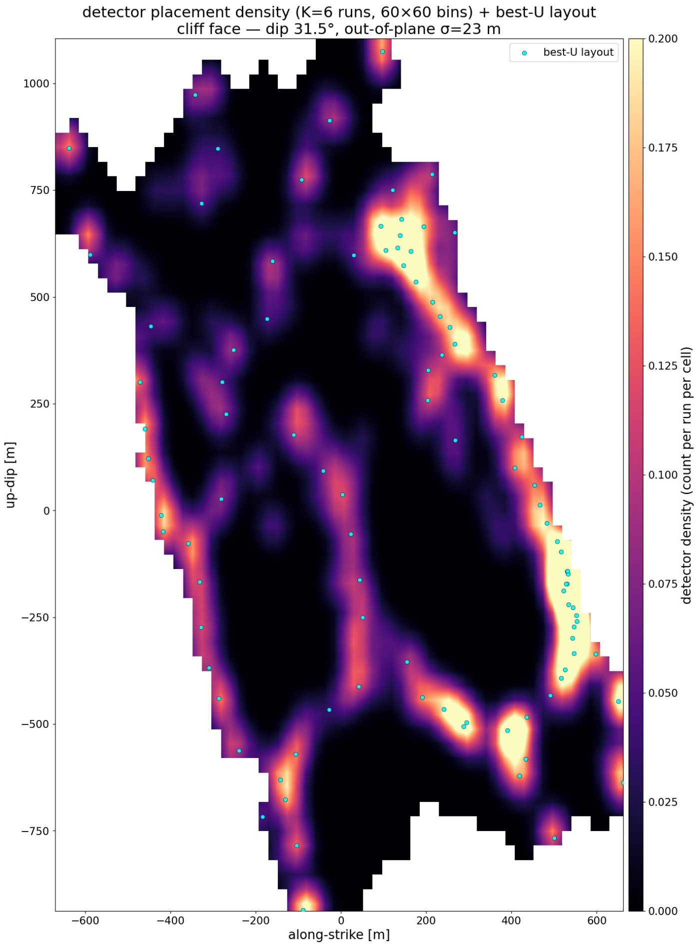
Utility	Detector response model	Flow matching surrogate
Grid initialization		
Baseline grid	32.10	23.48
Optimized	34.98	24.51
Center initialization		
Baseline center	10.77	9.58
Optimized	34.97	24.35

Stage 4: Detector array optimization – Composite utility (2)

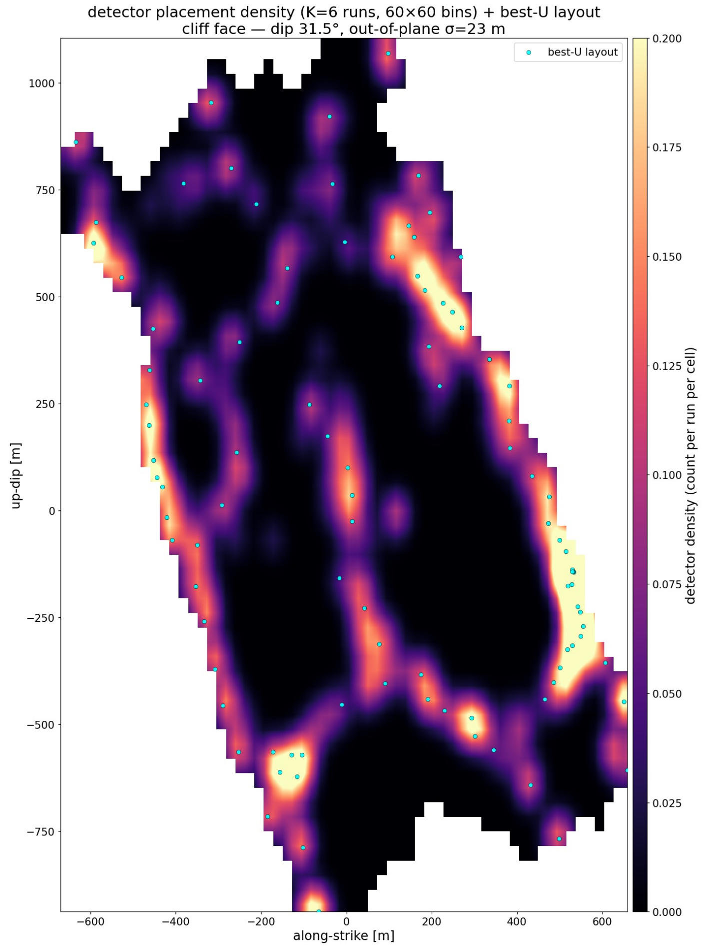


Stage 4: Detector array optimization – Composite utility (3)

Centre initialization



Grid initialization



Conclusion

- A high number of paramters need to be optimized in the Tambo observatory.
- A gradient-based method was proposed.
- A surrogate representing both architecture and primary particle paramter is used to optimize the array.
- Several optimization chains.

Next steps

- Add additional constraints about detector positions (preferred locations).

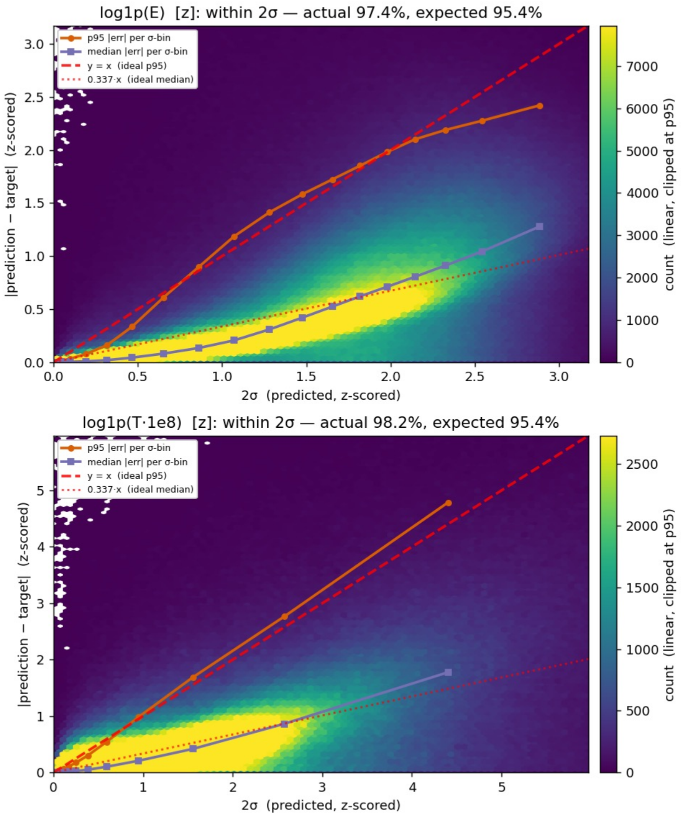
Thank you for your attention!

z.dimitrov308@proton.me

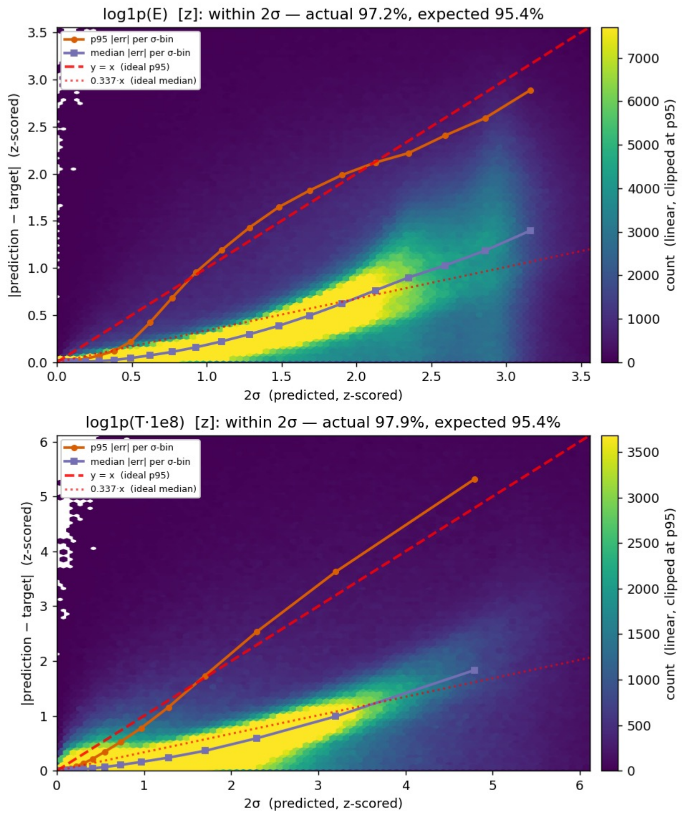
Additional slides

Stage 2: Detector response model – UQ

FNN predicted 2 σ vs realised error (electrons)



FNN predicted 2 σ vs realised error (muons)



Stage 4: Detector array optimization – Normalized detector activation

- Utility
  - defined as total detected energy
  - normalized by detector overlap
  - Detectors represented as Gaussian in space with sigma=50m
  - Based only on the detector response model
- Evaluate on composite utility comparisson

$$U_{\text{distinct}} = \frac{1}{B}\sum_b \frac{1}{N_{\text{det}}}\sum_d \frac{c_{bd}}{m_d}, \quad (1)$$

$$m_d = \sum_j \exp\left(-\frac{\|r_d - r_j\|^2}{2\sigma^2}\right), \quad (2)$$

$$\sigma = 50\text{ m}, \quad \mathbf{r}_d = (x_d, y_d) \quad (3)$$

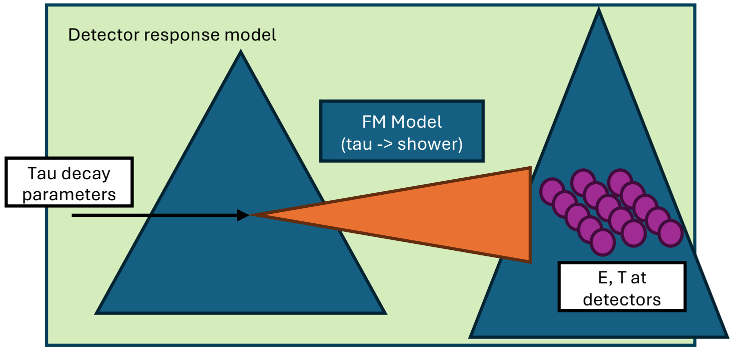
$$r_i = \sigma\left(\sum_j \sigma\left(\log_{10}\hat{E}_i - \lambda_{\text{layout}}\right) - \lambda_{\text{recon}}\right) \quad (1)$$

$$\mathcal{U}_E = \frac{1}{B}\sum_i \frac{r_i}{\left(\log_{10}\hat{E}_i - \log_{10}E_i\right)^2} \quad (2)$$

$$\mathcal{U}_\theta = \frac{1}{B}\sum_i \frac{r_i}{\left(\hat{\theta}_i - \theta_i\right)^2} \quad (3)$$

$$\mathcal{U}_\phi = \frac{1}{B}\sum_i \frac{r_i}{\left(\hat{\phi}_i - \phi_i\right)^2} \quad (4)$$

$$\mathcal{U} = \frac{w_\theta\mathcal{U}_\theta + w_\phi\mathcal{U}_\phi + w_E\mathcal{U}_E}{w_{\text{div}}} \quad (5)$$



Utility	Detector response model	Flow matching surrogate
	Grid initialization	
Baseline grid	32.10	23.48
Optimized	32.54	24.01
	Center initialization	
Baseline center	10.77	9.58
Optimized	32.63	24.06