NPML, UC Irvine *2nd June 2026*

Among a few other things...

Introducing WAND: a Water-cherenkov Annotated Neutrino Dataset

César JESÚS-VALLS

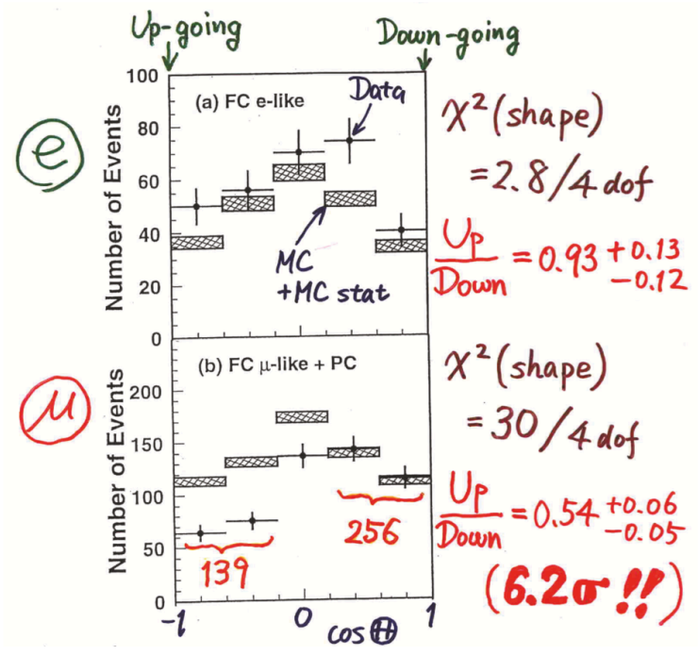
cesar.jesus@cern.ch

The Science: Neutrinos in HEP

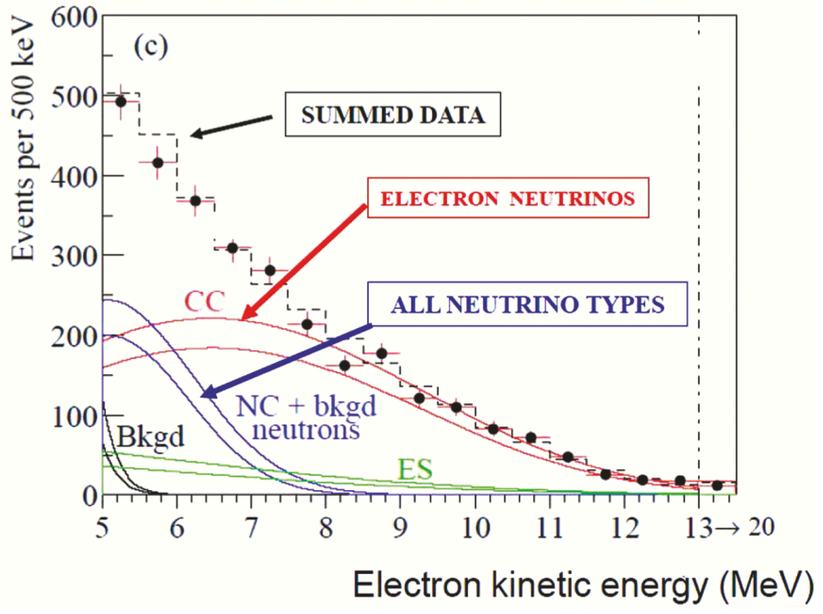
Why do we care about them in the first place?

A FEW NEUTRINO OSC 'HIGHLIGHTS'

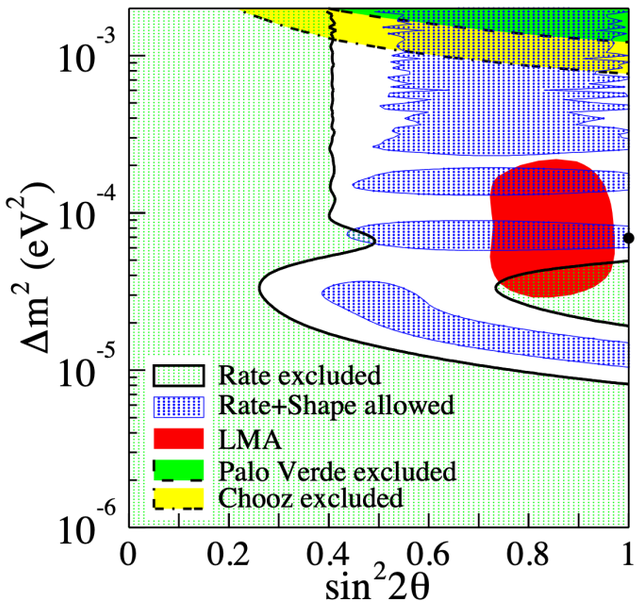
SK DEMONSTRATES ATM OSC



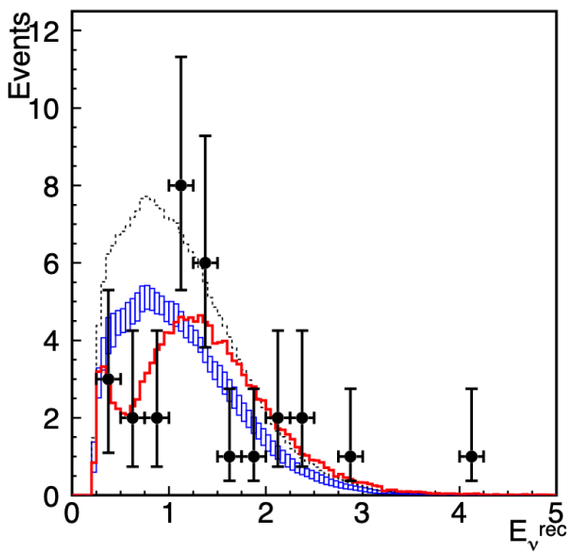
SNO SOLVES SOLAR 'ANOMALY'



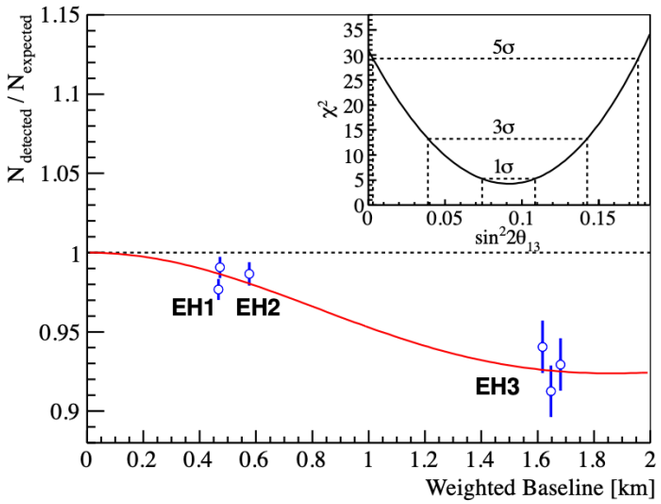
KAMLAND IDENTIFIES 'LMA'



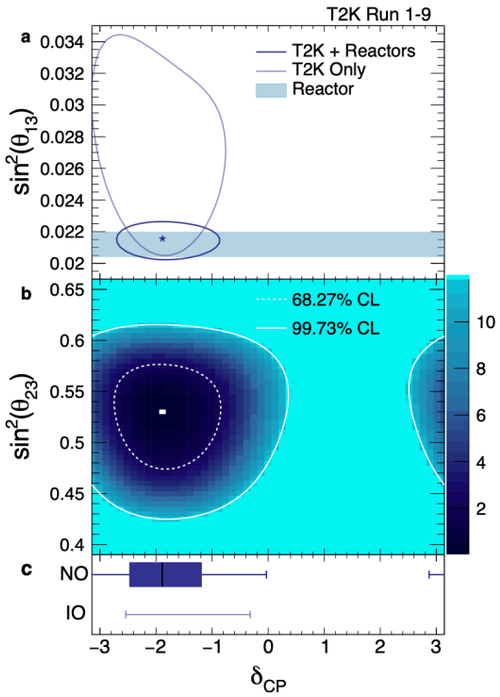
K2K FIRST OBS OF ACC ν OSC



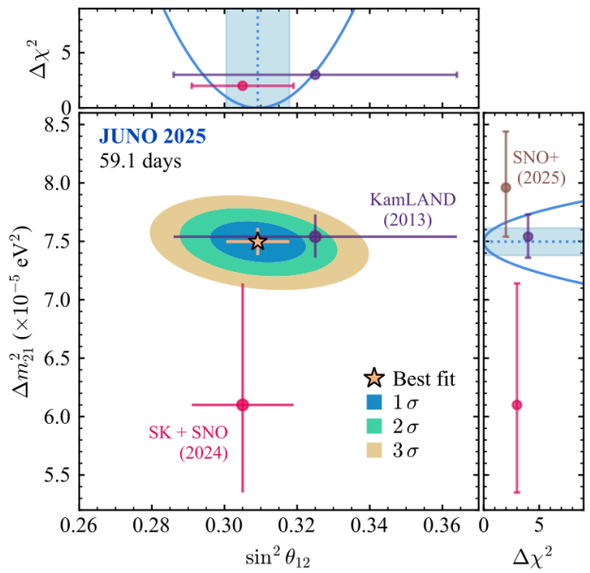
DAYA BAY ν_e DISAPPEARANCE



T2K HINTS OF CP VIOLATION



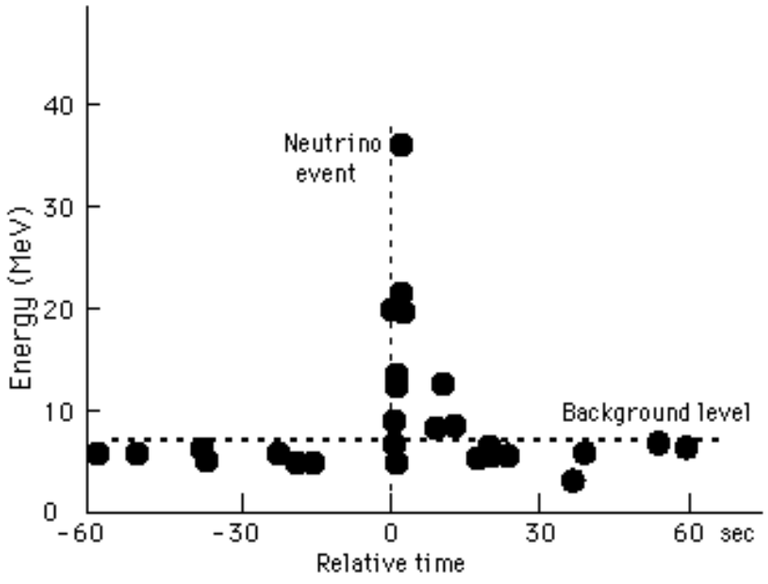
JUNO STARTS PRECISION ERA



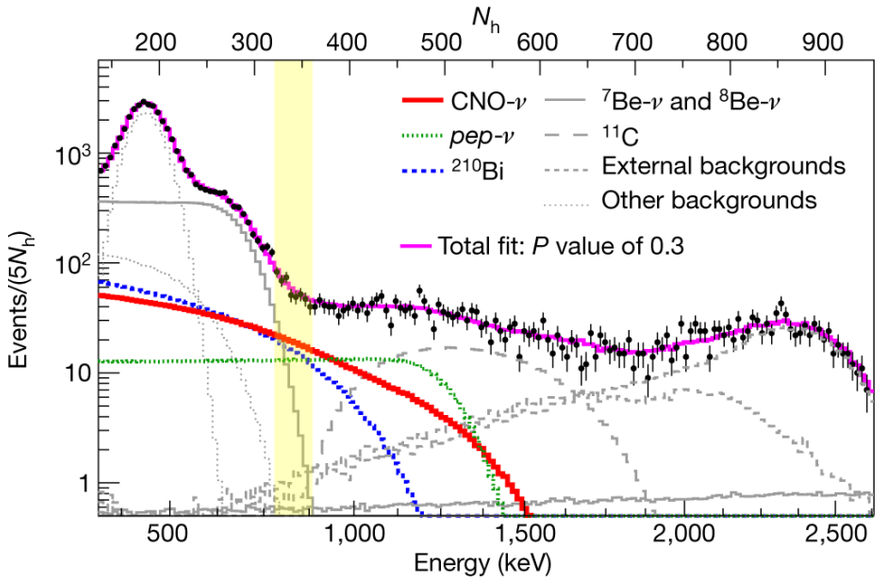
A FEW NEUTRINO ASTRO 'HIGHLIGHTS'

CERN

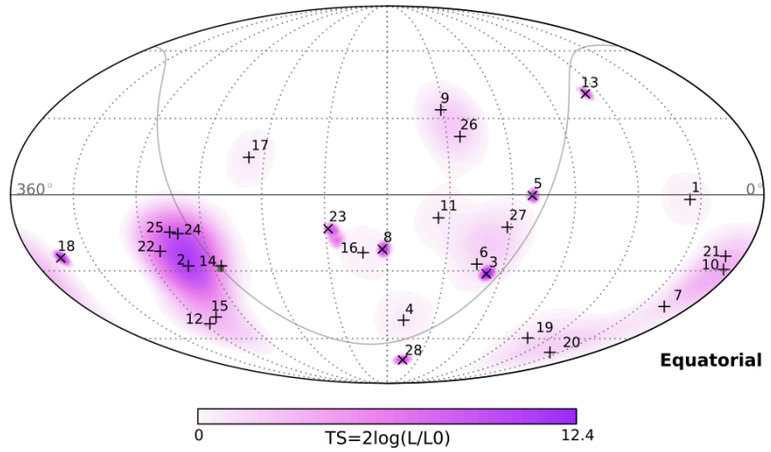
KAMIOKANDE SN 1987A



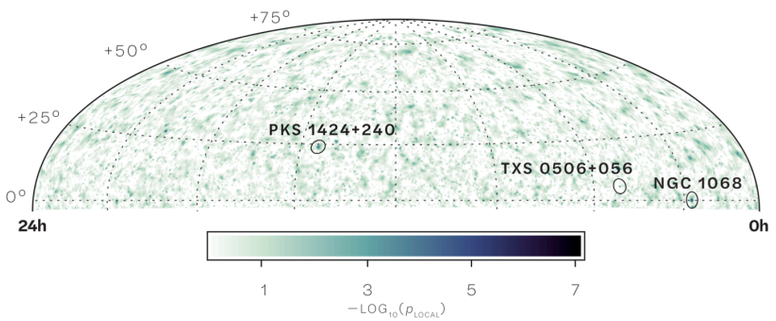
BOREXINO SOLAR SPECTROSCOPY



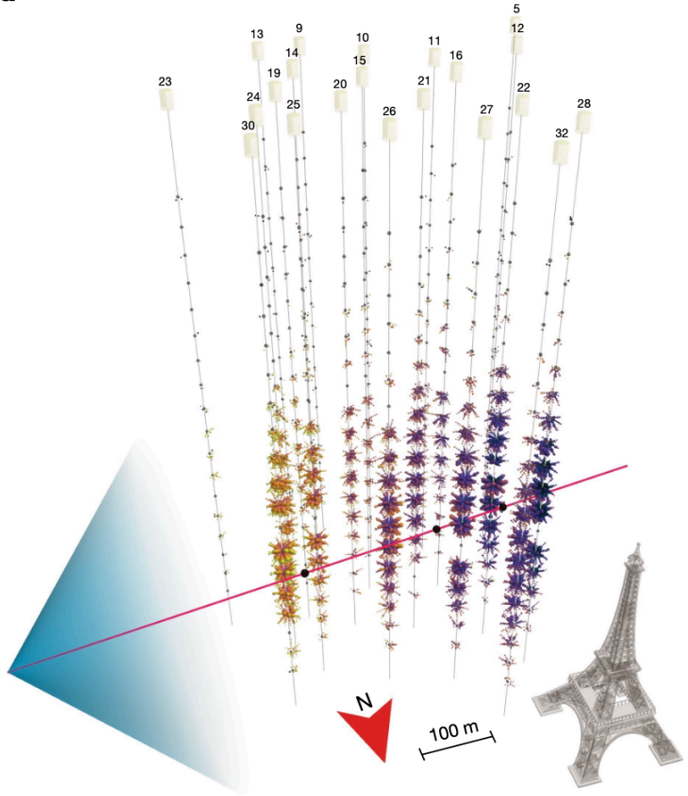
ICECUBE EXTRATERRESTRIAL ν FLUX



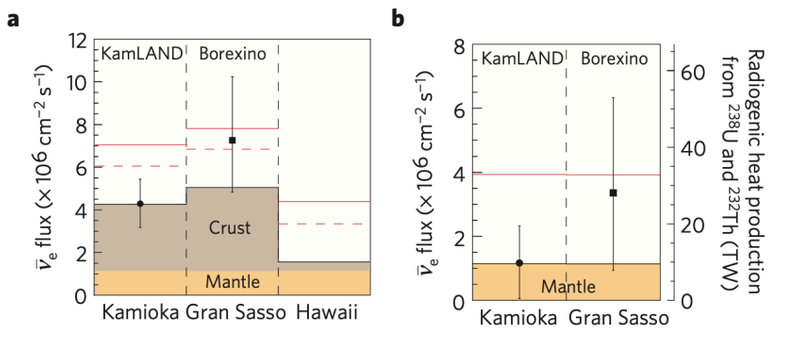
ICECUBE MULTIMESSENGER ASSOCIATATION



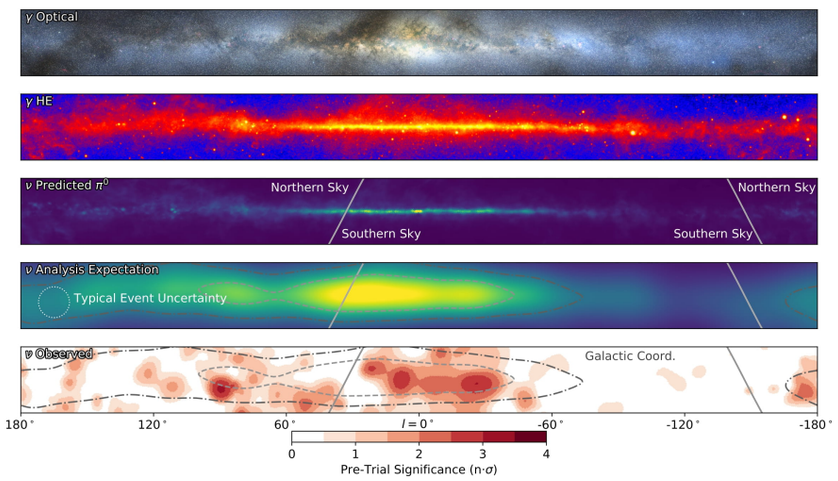
KM3NET DETECTS 220 PeV



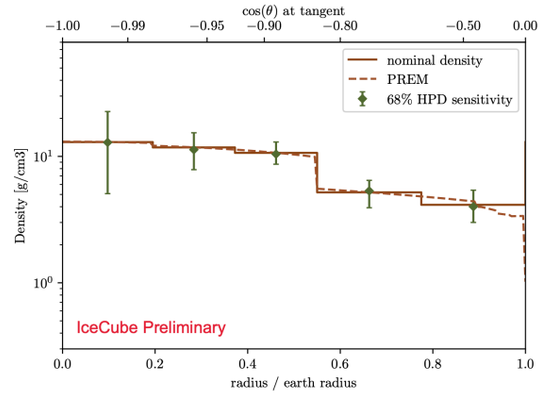
BONUS-I: Geo-neutrinos in KAMLAND/BOREXINO



ICECUBE GALACTIC PLANE ν FLUX



BONUS-II: ICECUBE EARTH TOMOGRAPHY



What do all these highlights have in common?

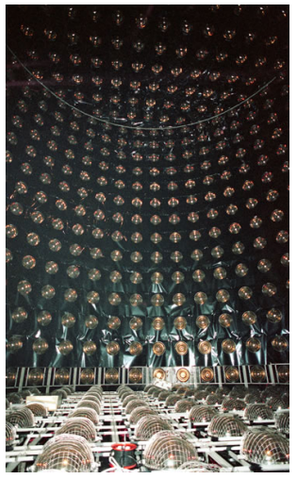
Other than neutrinos...

OPTICAL PARTICLE DETECTORS

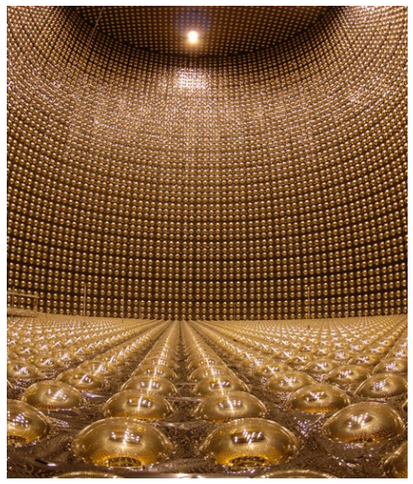
CERN

NEUTRINO PHYSICS HEAVILY RELIES IN THIS TYPE OF DETECTOR TECHNOLOGY

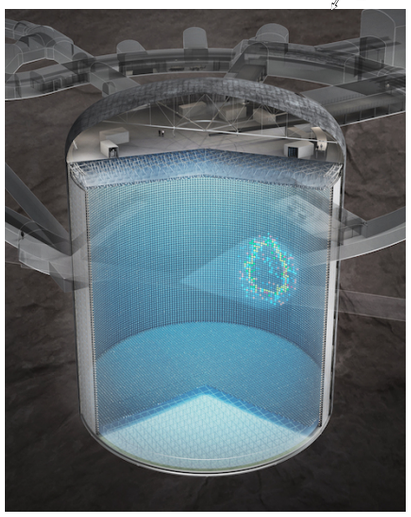
K



SUPER-K

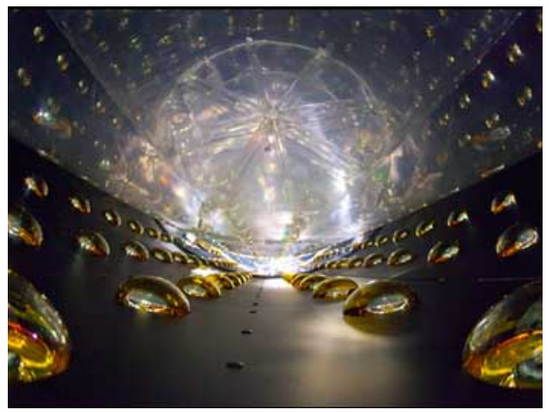


HYPER-K

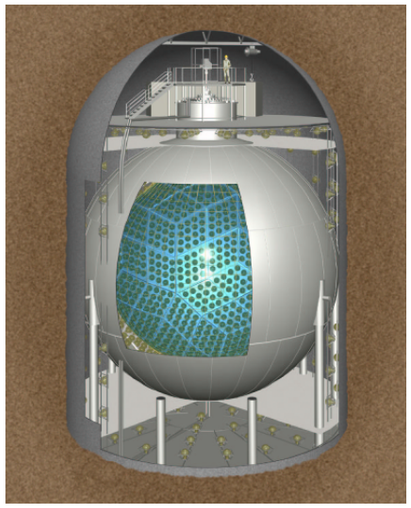


SNO

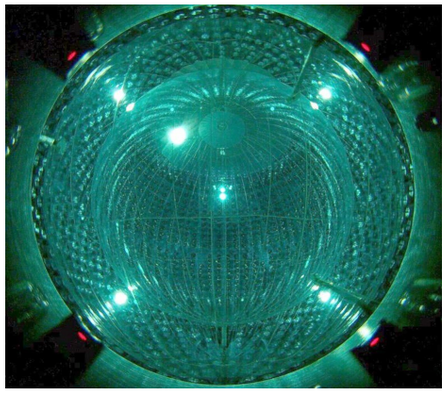
DAYA-BAY



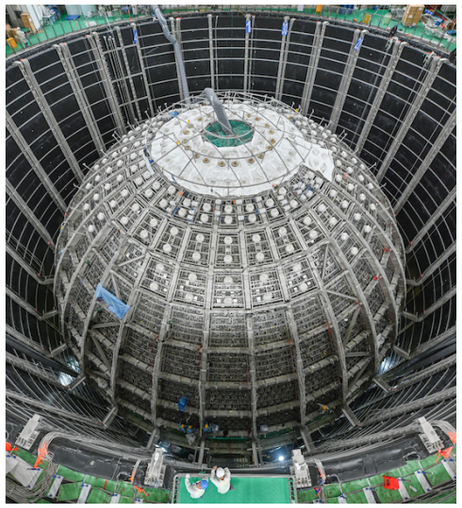
KAMLAND



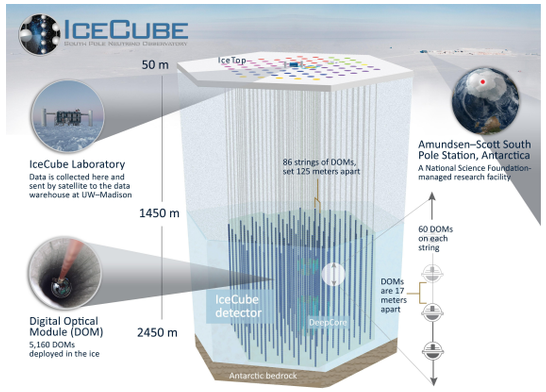
BOREXINO



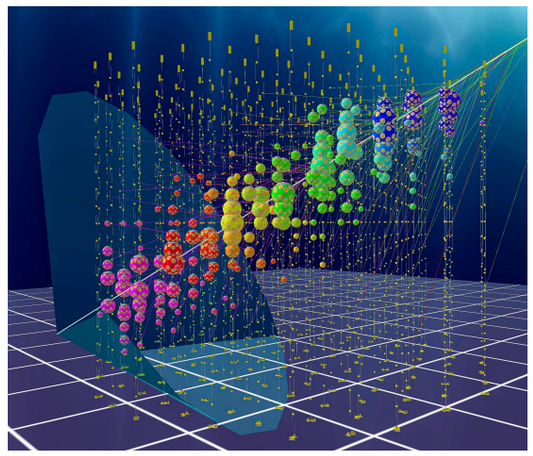
JUNO (~)



ICECUBE



KM3NET



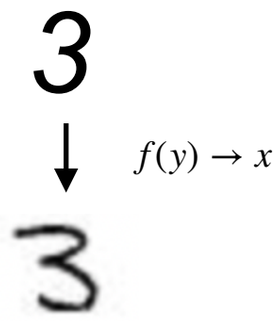
And many others experiments & demos (Double Chooz, LSND, MiniBooNE, ANNIE, WCTE, IWCD, EOS)...

Fighting black-box modelling

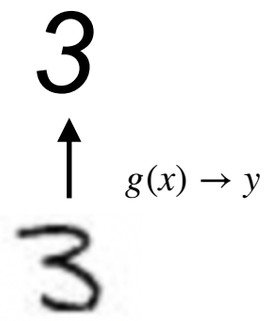
A BASIC CS EXAMPLE

Let's consider MNIST

For a given label "3", we can construct cases asking people to write the label

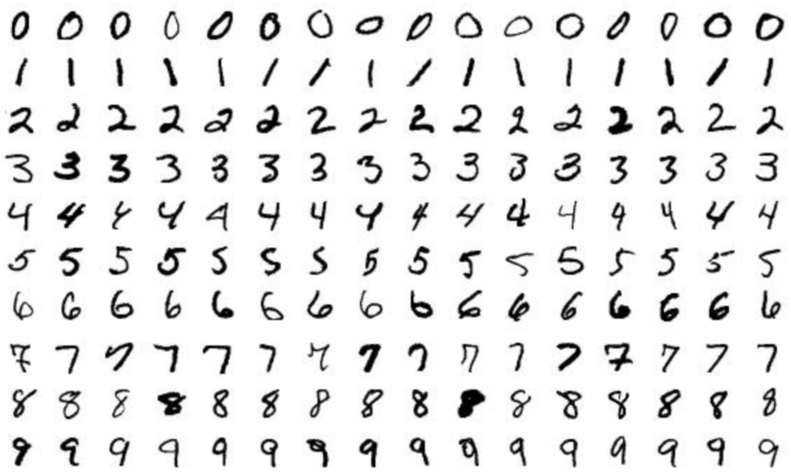


Solving the inverse problem

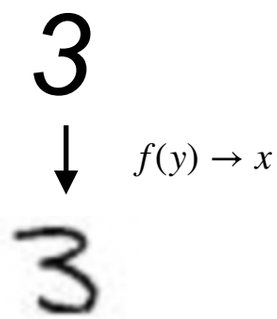


A BASIC CS EXAMPLE

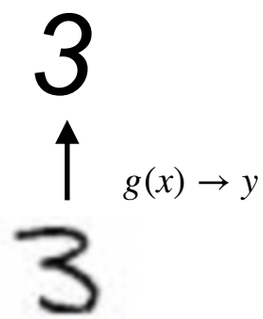
Let's consider MNIST



For a given label "3", we can construct cases asking people to write the label



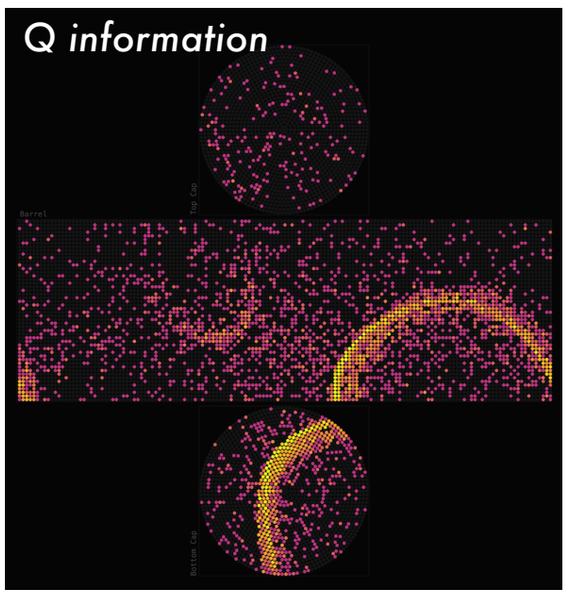
Solving the inverse problem

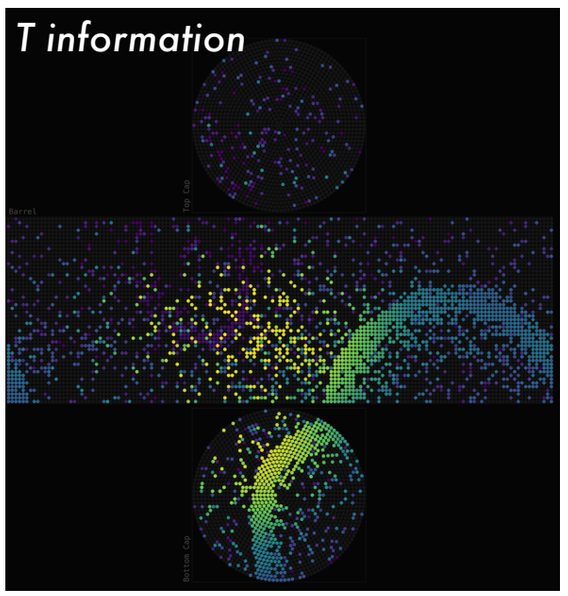


EXPERIMENTAL HIGH-ENERGY PHYSICS (HEP) AS AN INVERSE PROBLEM

Final State

Initial State (y) $\xrightarrow{f_{\vec{\theta}}(y) \to x}$





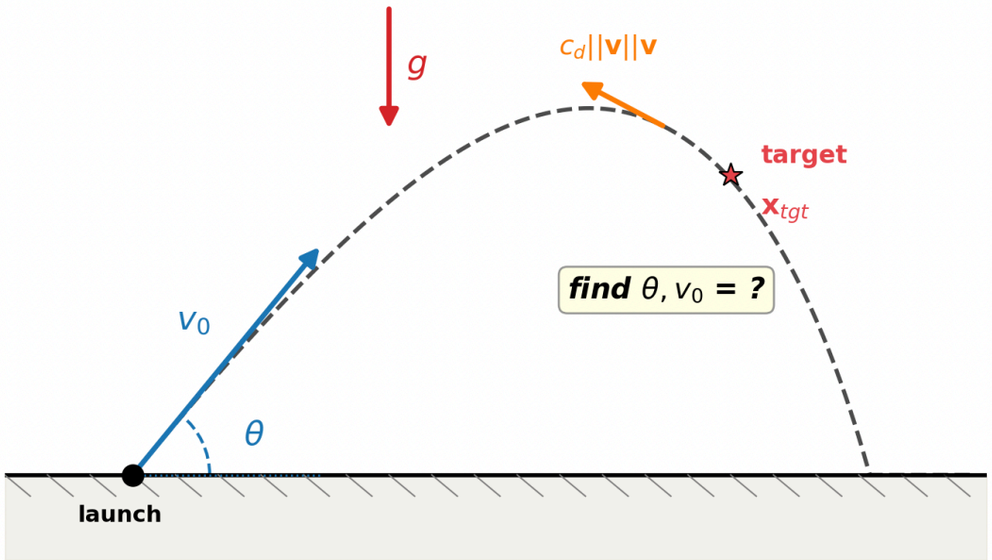
$\xrightarrow{g_{\vec{\theta}}(x) \to y}$ *Inferred Initial State (y')*

What is $\vec{\theta}$: Properties of a given material (e.g. transparency) or sensor (e.g. detection efficiency) or detector (e.g. electric field), etc.

Black Box reconstruction is NOT enough in HEP! Often we care about intermediate parameters θ!

What is a *differentiable physics simulation*?

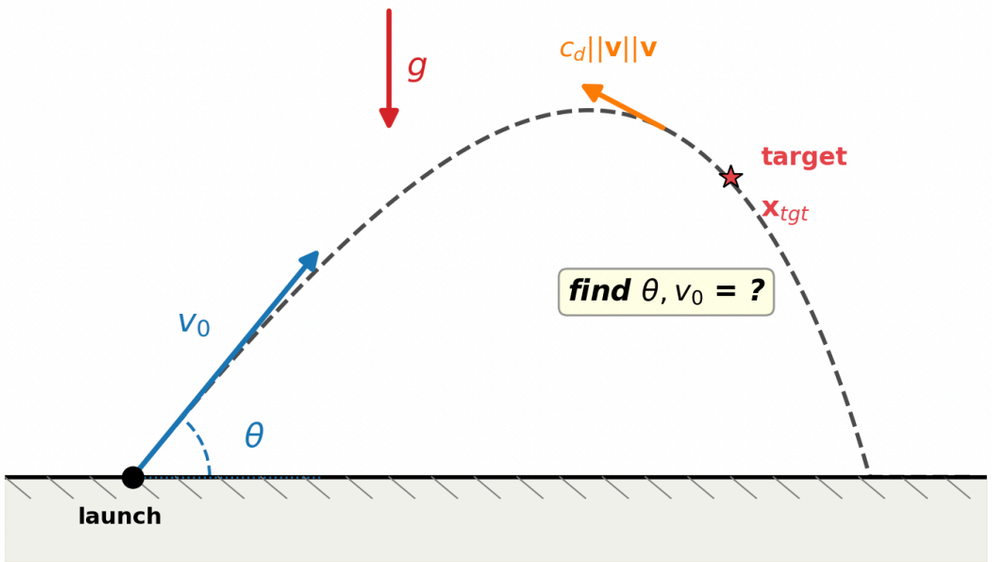
Inverse problem: optimise launch parameters to hit target



A DIFF SIM MINIMAL EXAMPLE

What is a *differentiable physics simulation*?

Inverse problem: optimise launch parameters to hit target



FORWARD

Let's first calculate trajectory taking many small steps forward

$$\mathbf{a}_t = -g\,\hat{y} - c_d\,||\mathbf{v}_t||\,\mathbf{v}_t$$

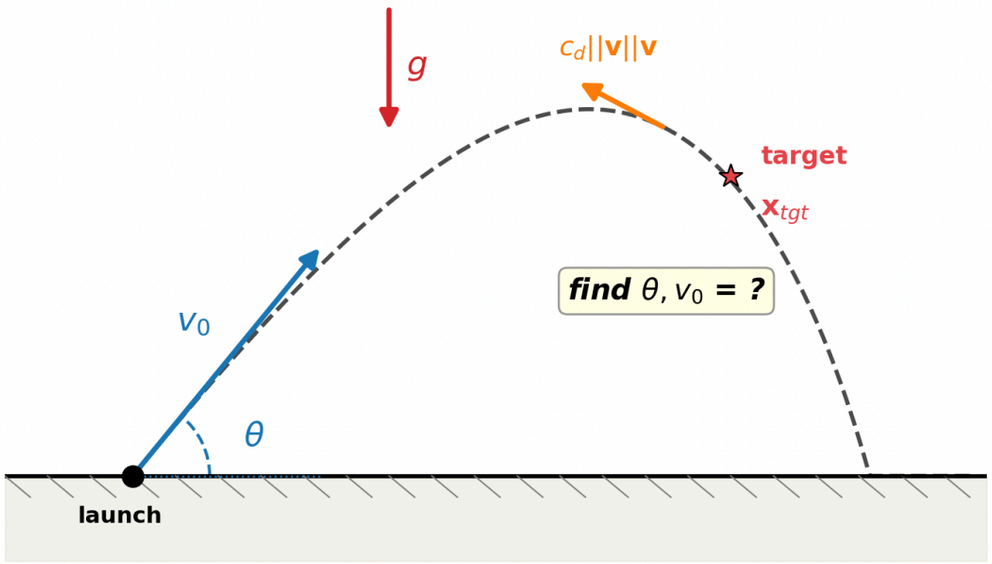
$$\mathbf{v}_{t+1} = \mathbf{v}_t + \mathbf{a}_t\,\Delta t$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v}_{t+1}\,\Delta t$$

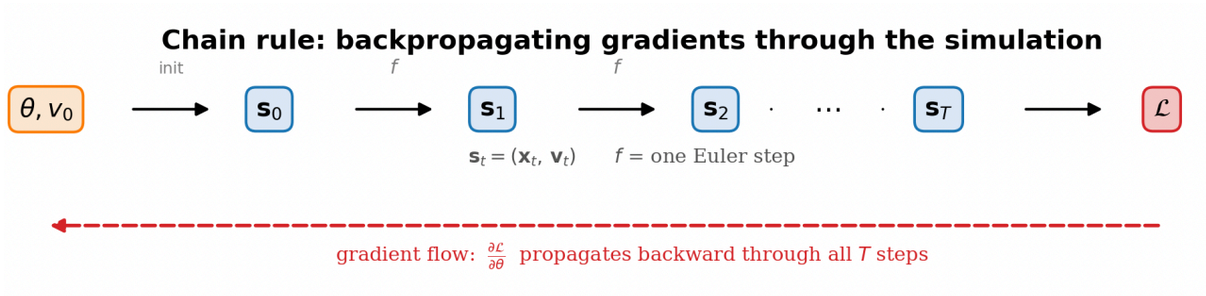
Then lets calculate how far is the closest point in the trajectory to the target, and define the loss $\mathscr{L}$

What is a *differentiable physics simulation*?

Inverse problem: optimise launch parameters to hit target



BACKWARD



Chain rule (reverse mode AD):

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \mathbf{s}_T} \cdot \frac{\partial \mathbf{s}_T}{\partial \mathbf{s}_{T-1}} \cdot \ldots \cdot \frac{\partial \mathbf{s}_1}{\partial \mathbf{s}_0} \cdot \frac{\partial \mathbf{s}_0}{\partial \theta}$$

$$= \frac{\partial \mathcal{L}}{\partial \mathbf{s}_T} \cdot \prod_{t=1}^{T} \frac{\partial f}{\partial \mathbf{s}}\Big|_{\mathbf{s}_{t-1}} \cdot \frac{\partial \mathbf{s}_0}{\partial \theta}$$

FORWARD

Let's first calculate trajectory taking many small steps forward

$$\mathbf{a}_t = -g\,\hat{y} - c_d\,||\mathbf{v}_t||\,\mathbf{v}_t$$

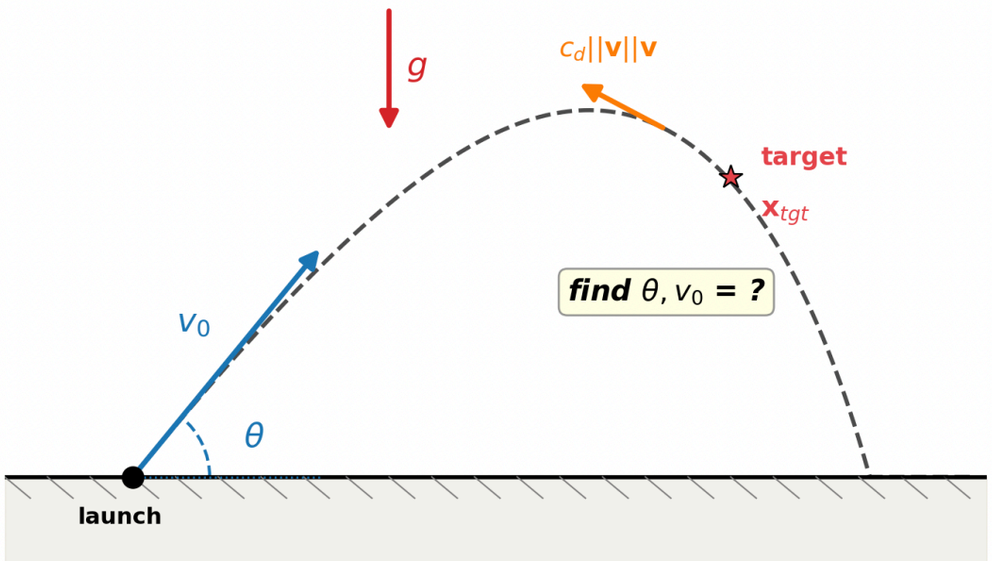
$$\mathbf{v}_{t+1} = \mathbf{v}_t + \mathbf{a}_t\,\Delta t$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v}_{t+1}\,\Delta t$$

Then lets calculate how far is the closest point in the trajectory to the target, and define the loss $\mathscr{L}$

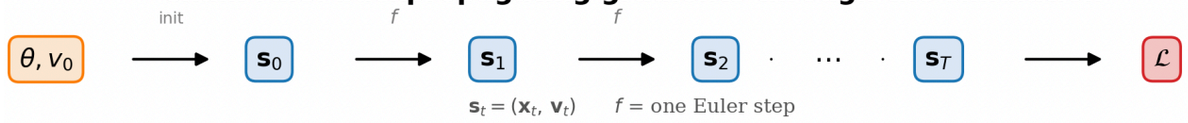
What is a *differentiable physics simulation*?

Inverse problem: optimise launch parameters to hit target



BACKWARD

Chain rule: backpropagating gradients through the simulation



Chain rule (reverse mode AD):

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \mathbf{s}_T} \cdot \frac{\partial \mathbf{s}_T}{\partial \mathbf{s}_{T-1}} \cdot \ldots \cdot \frac{\partial \mathbf{s}_1}{\partial \mathbf{s}_0} \cdot \frac{\partial \mathbf{s}_0}{\partial \theta}$$

$$= \frac{\partial \mathcal{L}}{\partial \mathbf{s}_T} \cdot \prod_{t=1}^{T} \frac{\partial f}{\partial \mathbf{s}}\Big|_{\mathbf{s}_{t-1}} \cdot \frac{\partial \mathbf{s}_0}{\partial \theta}$$

Take home message: We define the forward operation, we tune parameters leveraging AD!

FORWARD

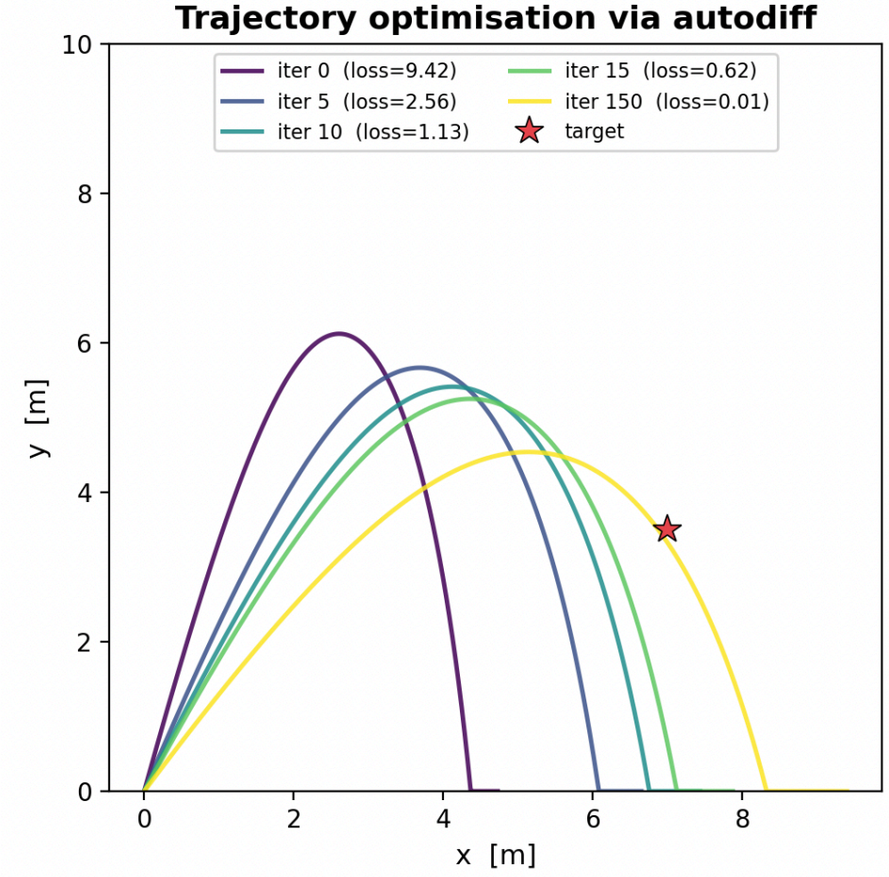
Let's first calculate trajectory taking many small steps forward

$$\mathbf{a}_t = -g\,\hat{y} - c_d\,||\mathbf{v}_t||\,\mathbf{v}_t$$

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \mathbf{a}_t\,\Delta t$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v}_{t+1}\,\Delta t$$

Then lets calculate how far is the closest point in the trajectory to the target, and define the loss $\mathscr{L}$



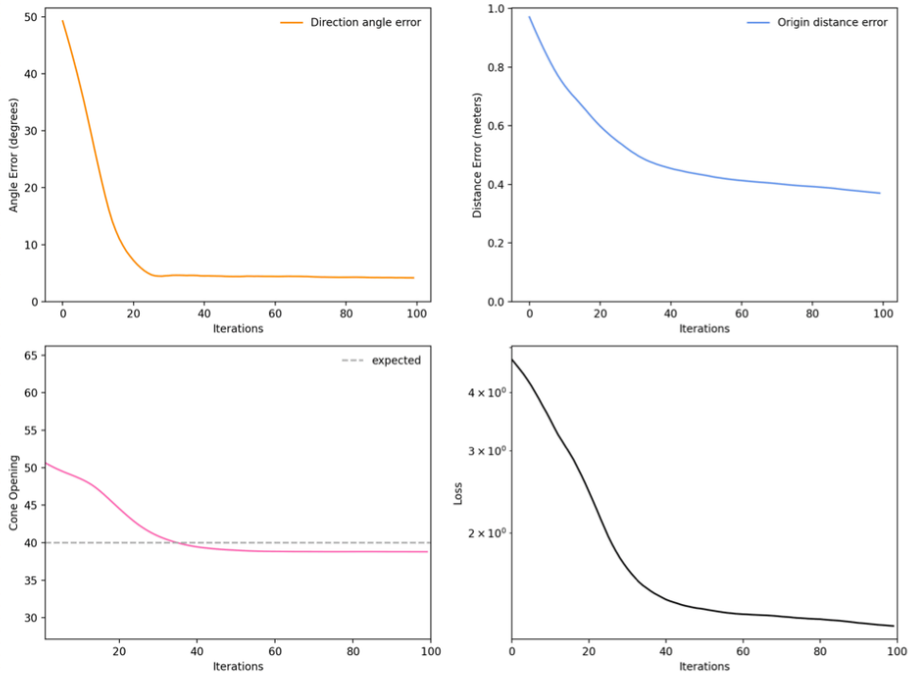
Diff Sim Tutorials

https://github.com/cesarjesusvalls/Physics_DiffSim_Tutorial

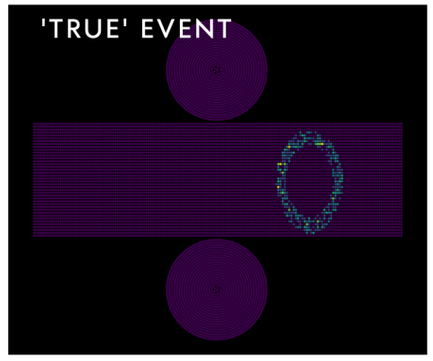
https://github.com/cesarjesusvalls/differentiable-sampling

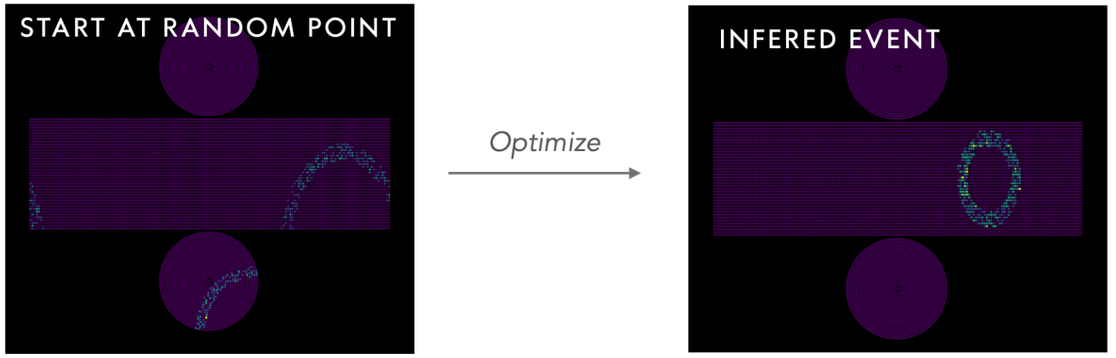
[NPML 2024 Talk]

Proof-of-concept in 2024



*Select **true** parameters: $(\vec{x}, \vec{v}, \theta_{CH})$*





2024 Conclusions

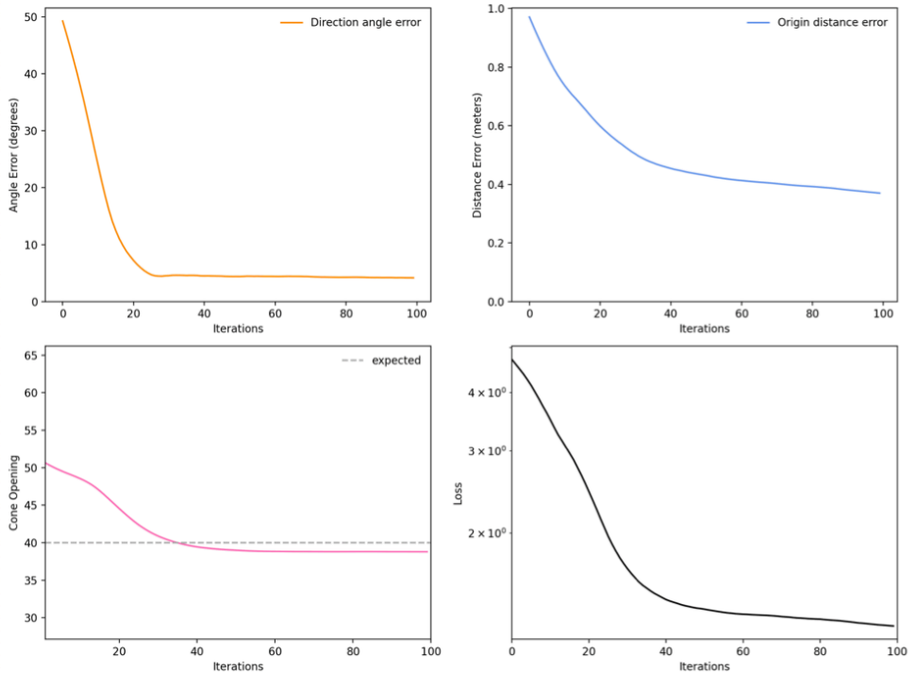
Next step is to add up stochastic processes and optimize related physics parameters (scattering, reflections).

Are we ready to build end-to-end analytical models? No.
But do we want to keep working in the same way in the next 20 years?

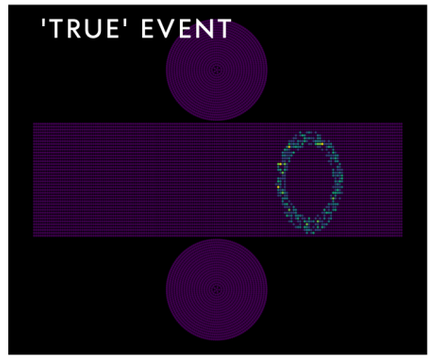
Walking steps in the direction of alternative solutions can help us to understand their limitations & advantages. Differentiable detector simulations have the potential to redefine old-existing paradigms in HEP-ex.

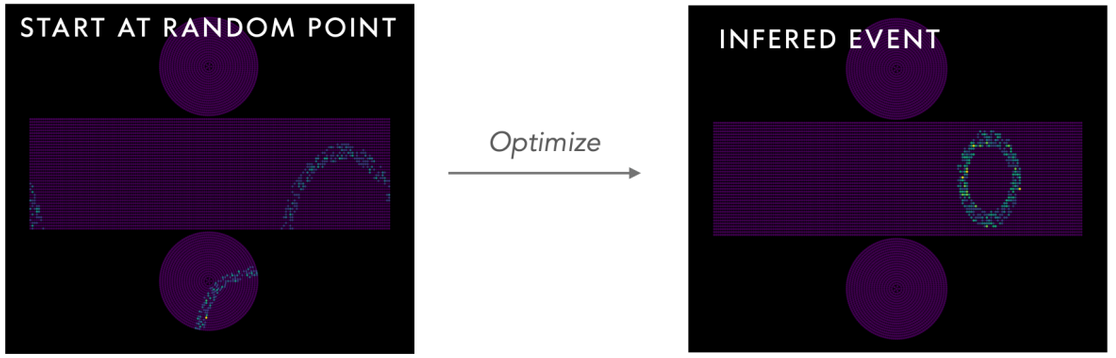
[NPML 2024 Talk]

Proof-of-concept in 2024



*Select **true** parameters: $(\vec{x}, \vec{v}, \theta_{CH})$*



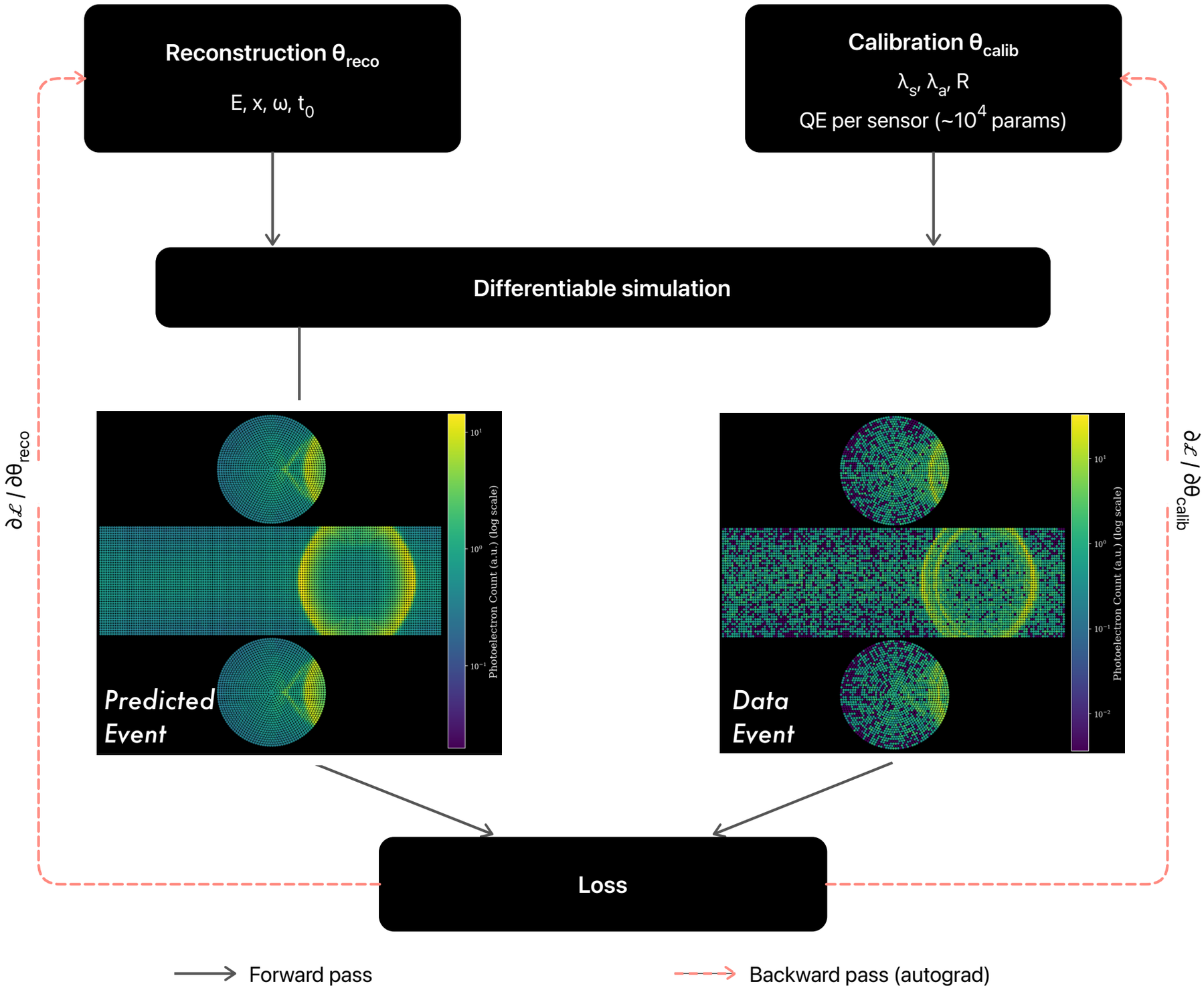


2024 Conclusions

Next step is to add up stochastic processes and optimize related physics parameters (scattering, reflections).

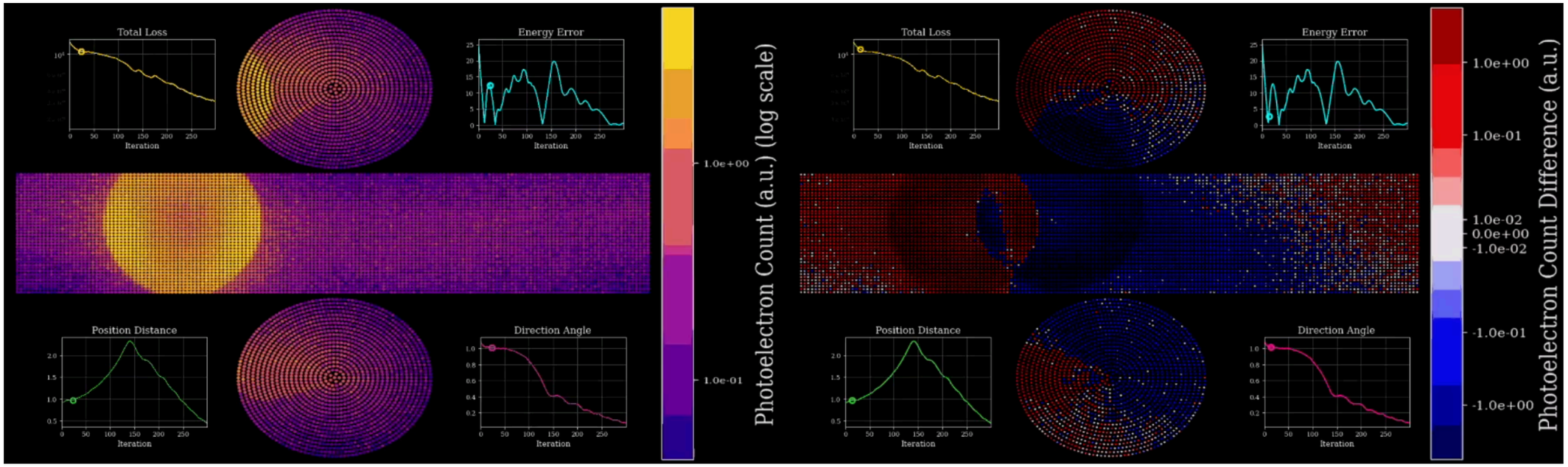
Are we ready to build end-to-end analytical models? No. **Well will see about that...**
But do we want to keep working in the same way in the next 20 years?

Walking steps in the direction of alternative solutions can help us to understand their limitations & advantages. Differentiable detector simulations have the potential to redefine old-existing paradigms in HEP-ex.


Reconstruction θ_{reco}
E, x, ω, t_0
Calibration θ_{calib}
λ_s, λ_a, R
QE per sensor (~10^4 params)
Differentiable simulation
Predicted Event
Data Event
Photoelectron Count (a.u.) (log scale)
Loss
$\partial\mathcal{L} / \partial\theta_{reco}$
$\partial\mathcal{L} / \partial\theta_{calib}$
Forward pass
Backward pass (autograd)

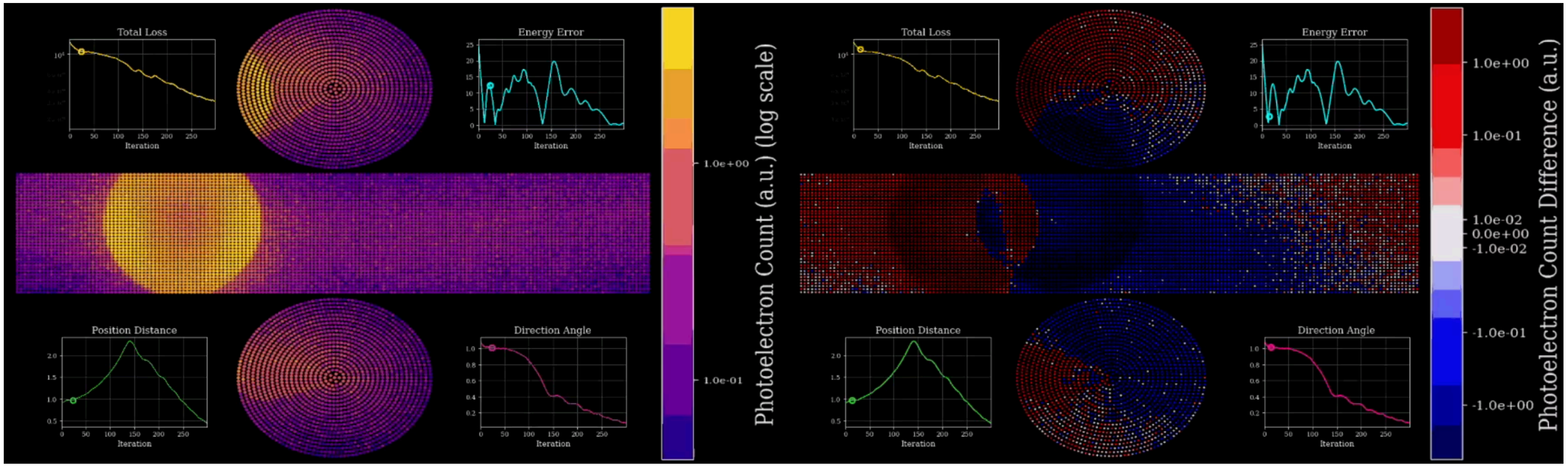
NPML 2025 Talk from Omar

Added full-stochasticity



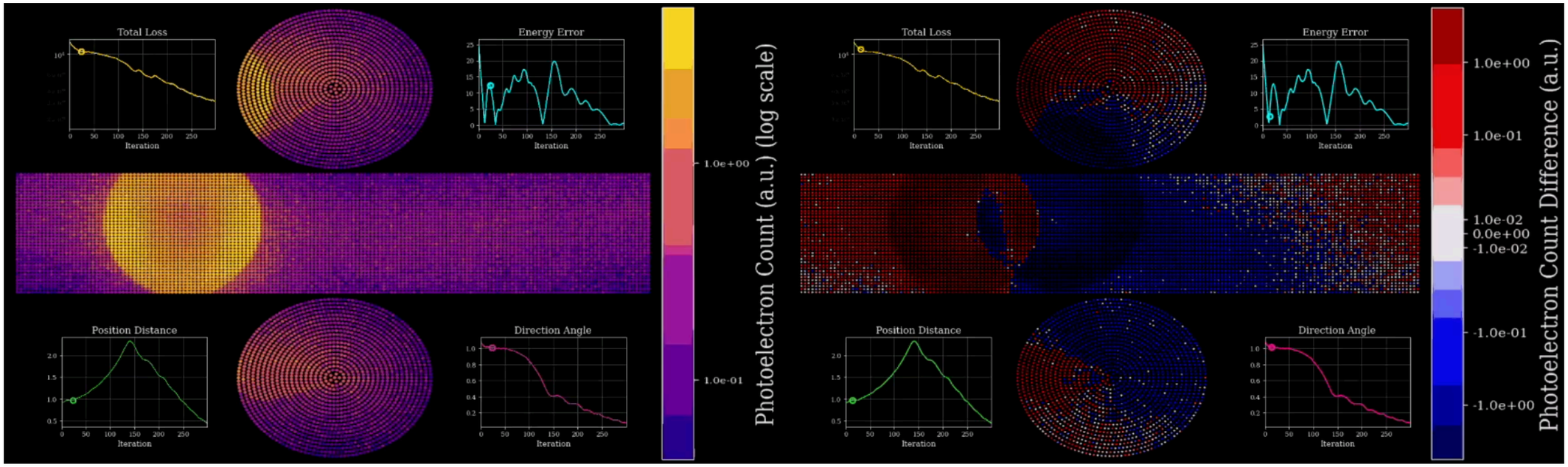
NPML 2025 Talk from Omar

Added full-stochasticity

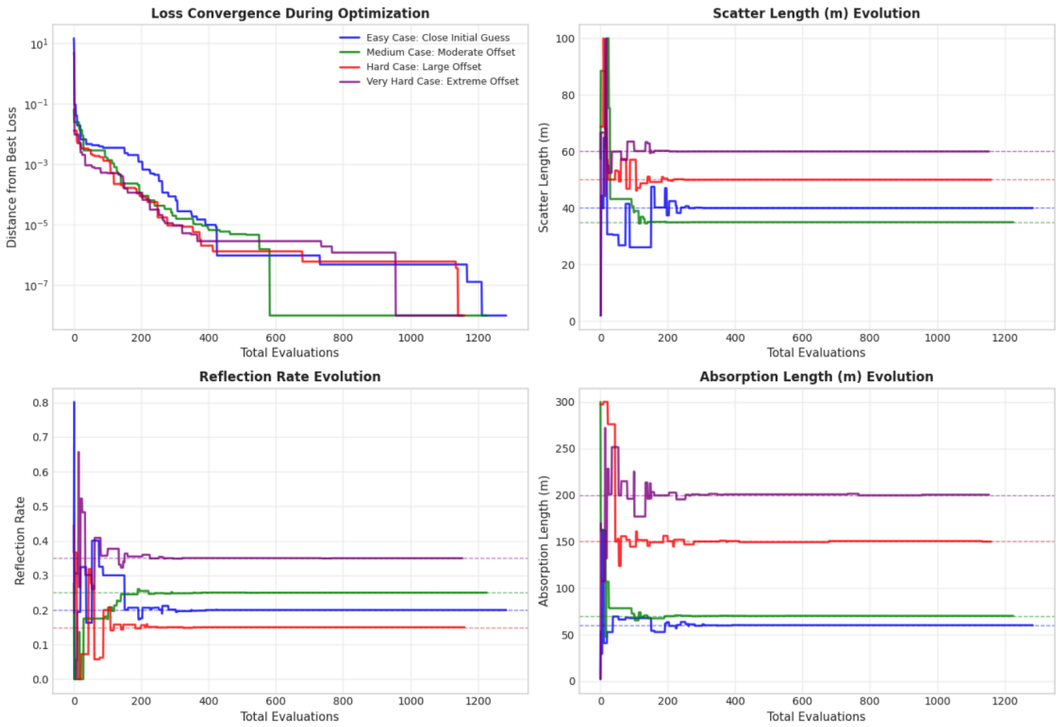


NPML 2025 Talk from Omar

Added full-stochasticity

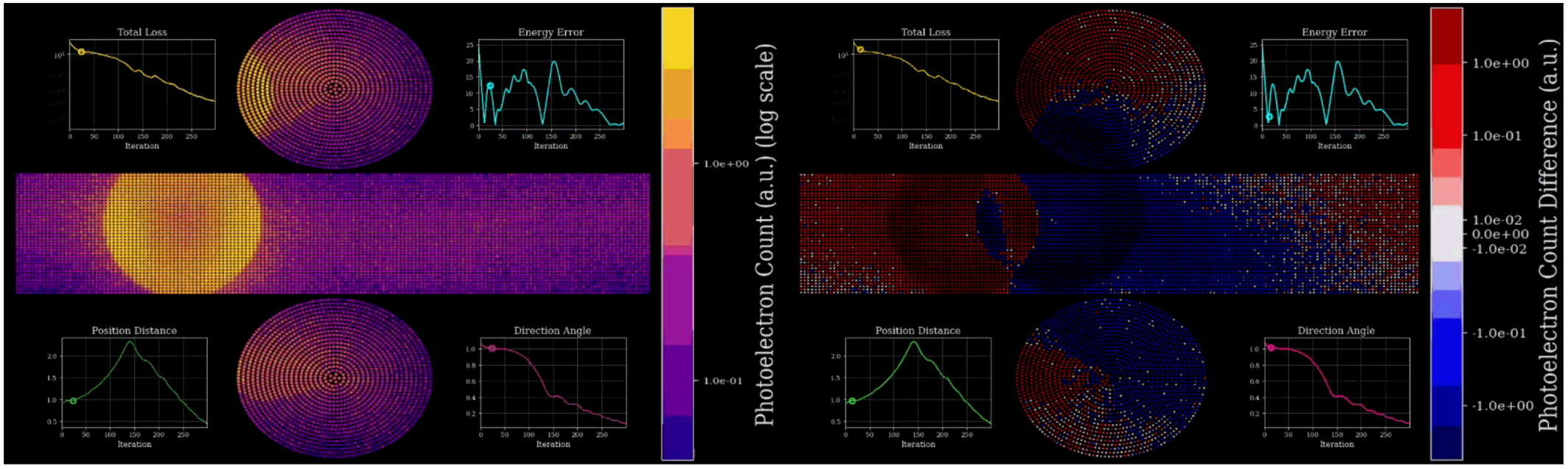


Demonstrated calibration capabilities

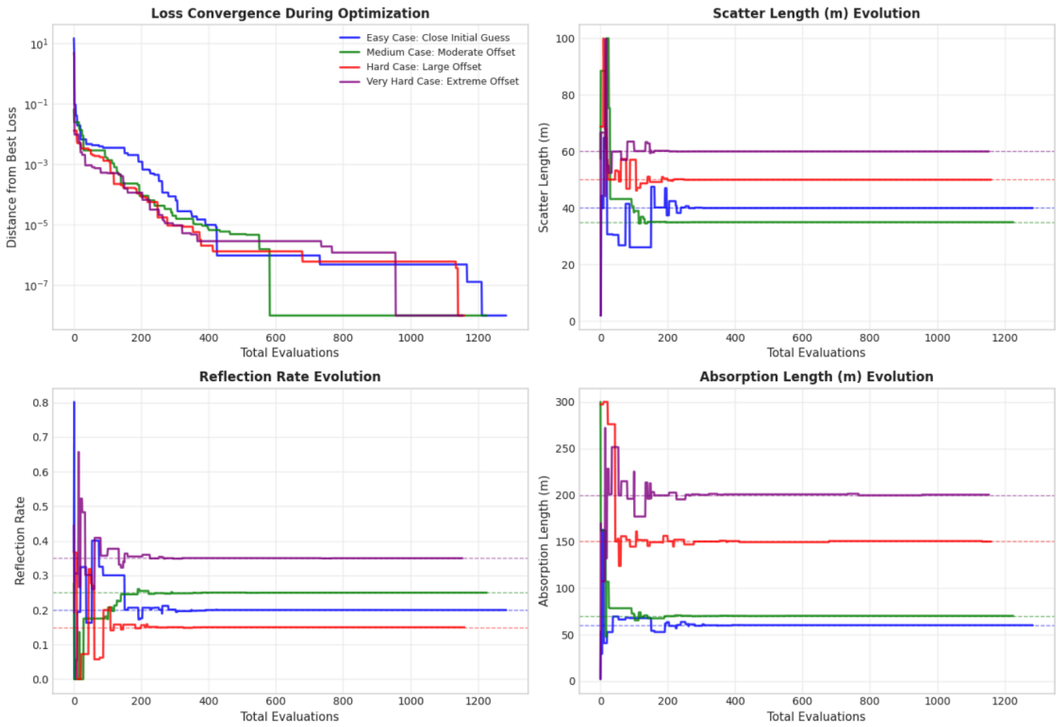


NPML 2025 Talk from Omar

Added full-stochasticity

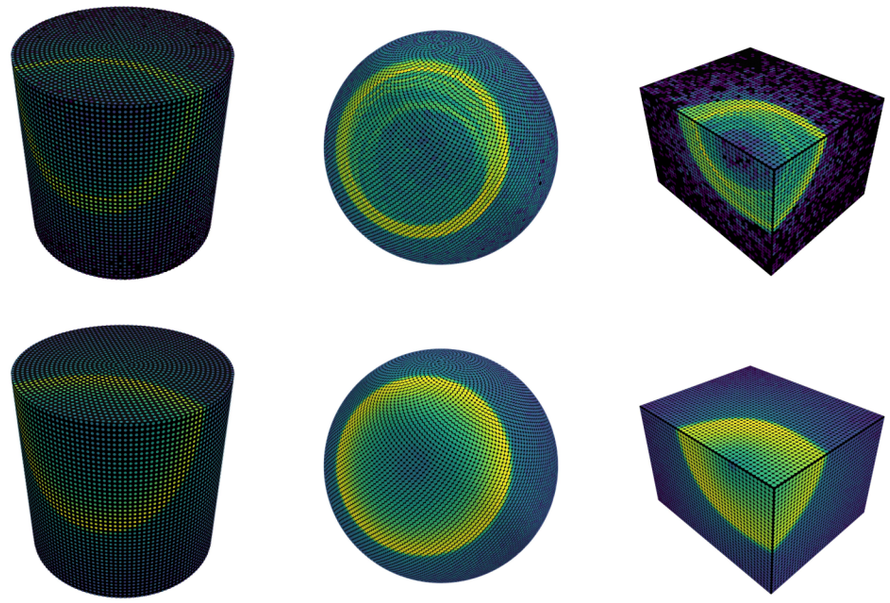


Demonstrated calibration capabilities



Abstracted geometry away

EXAMPLE OF DATA-LIKE EVENTS (TOP) AND ASSOCIATED PREDICTIONS (BOTTOM)



Project consolidation

LUCiD: a Light-based Unified Calibration and trackIng Differentiable simulation

https://arxiv.org/abs/2602.24129

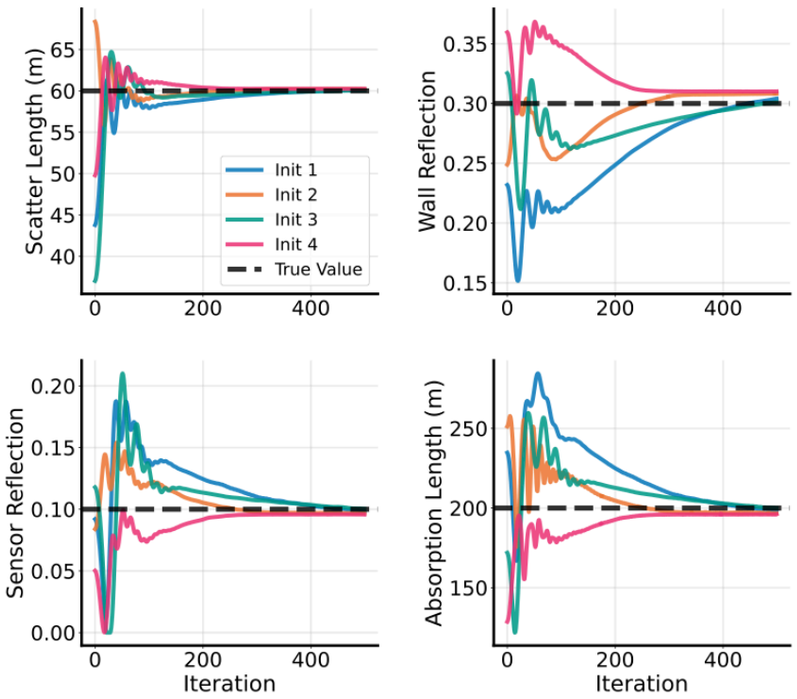
https://github.com/CIDeR-ML/LUCiD/

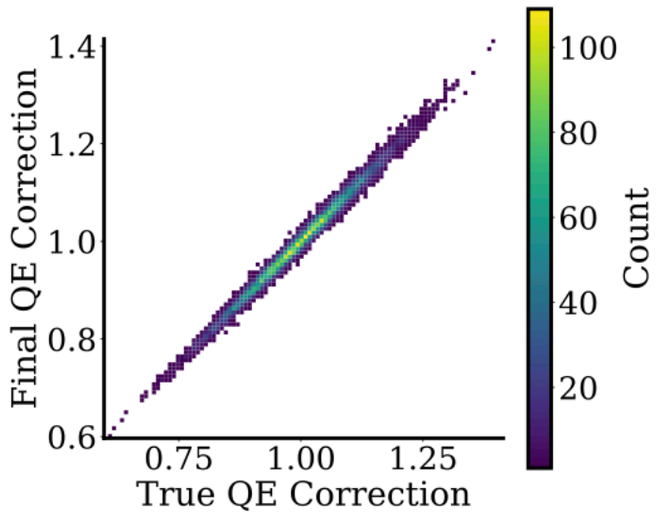
O. Alterkait (U. Tufts/IAIFI)

C. Jesús-Valls (CERN)

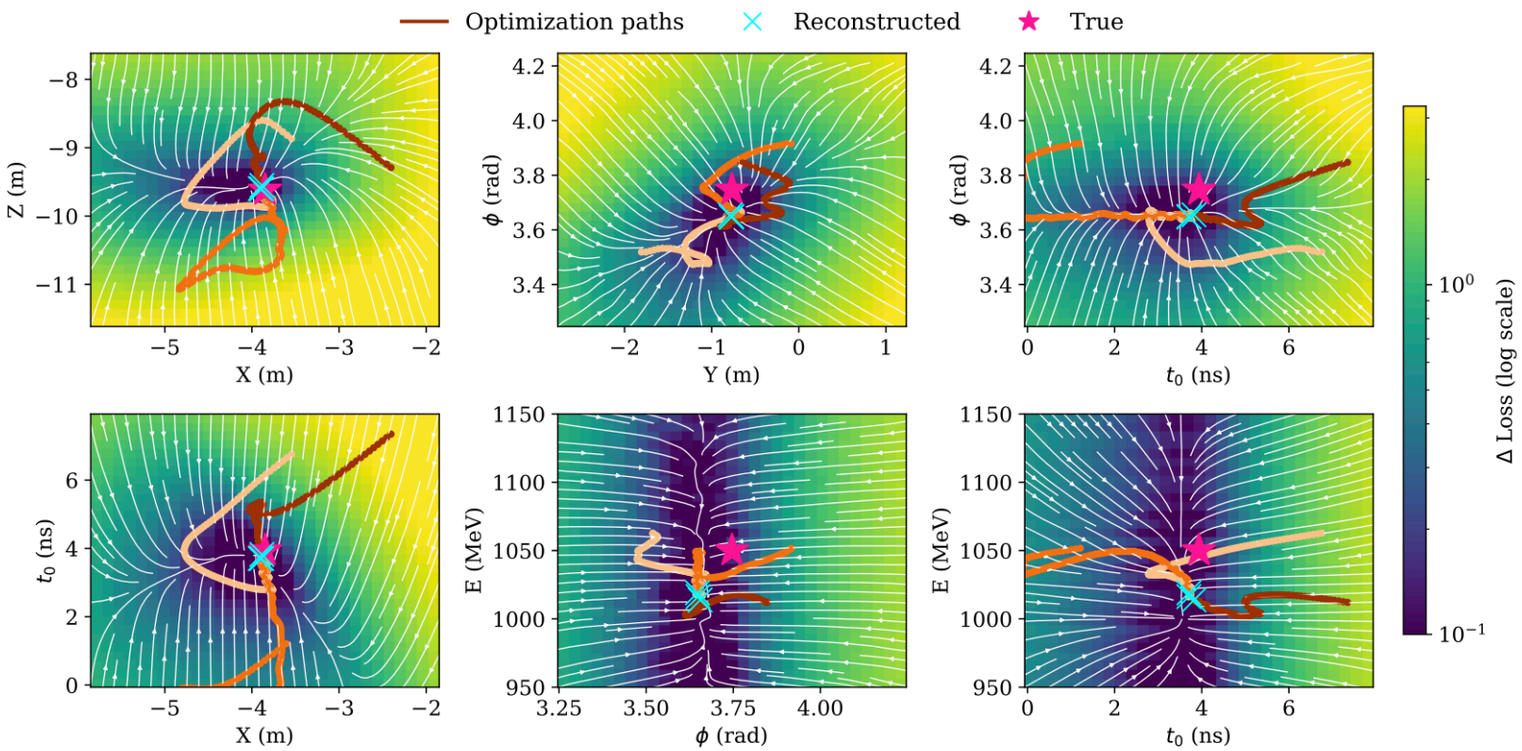
Still under heavy development. We keep refining and adding tools.

Calibration Tests





Track reconstruction tests

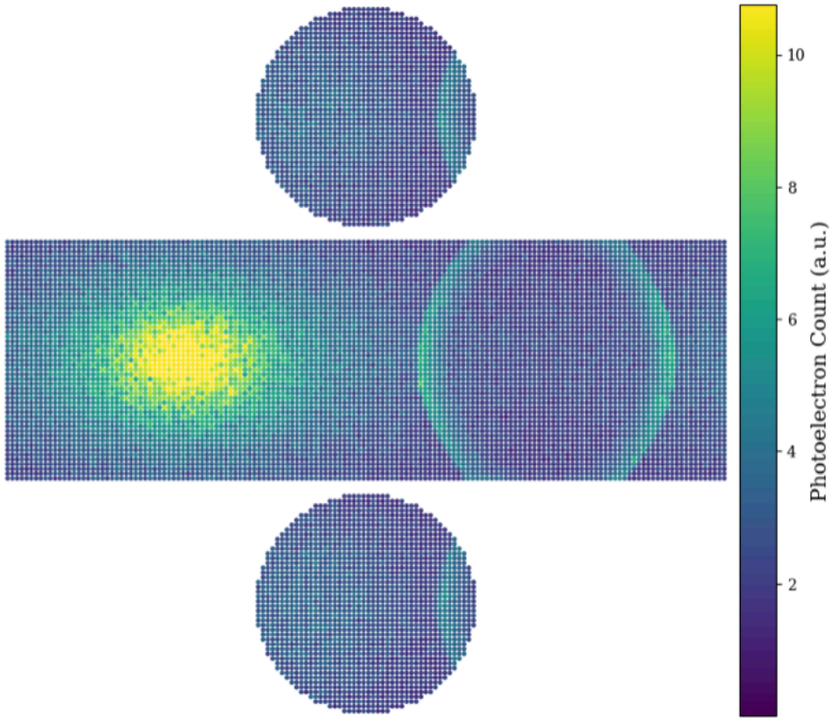


How does this compare to SK reconstruction software?

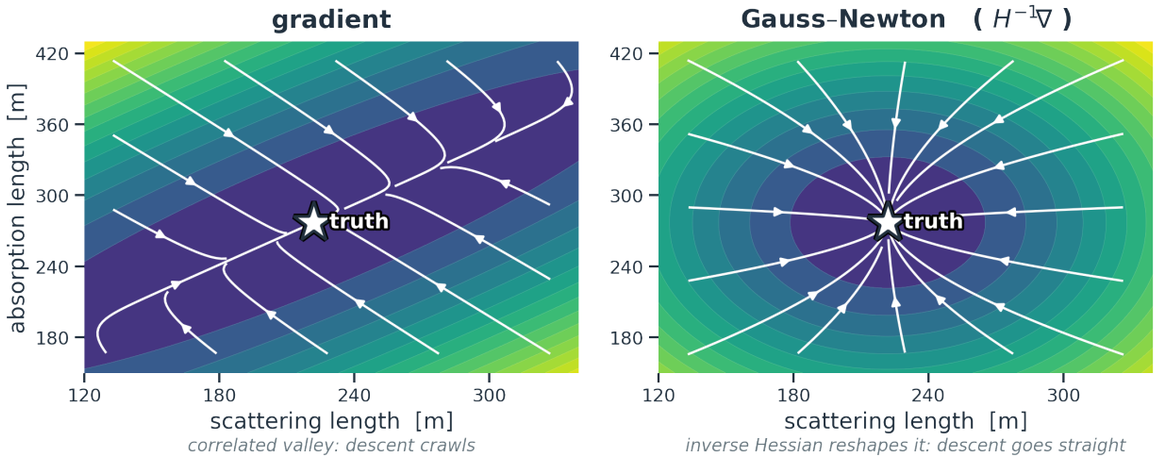
Performance (1 GeV muon, SK-like)				
	Vertex	Direction	Momentum	Time / event
LUCiD	15.6 cm	1.1°	1.5%	~10 s
fiTQun	15.8 cm	1.0°	2.3%	~30 s

Approximate comparison using **very similar dataset*

Differentiable prediction for WbLS



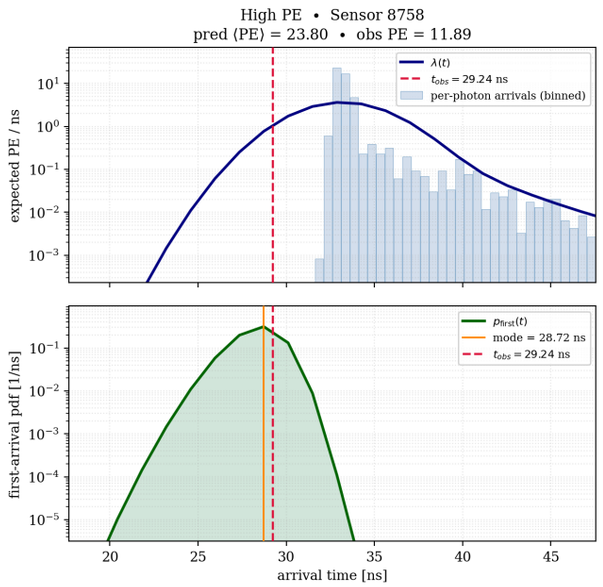
Improving our optimisation landscape



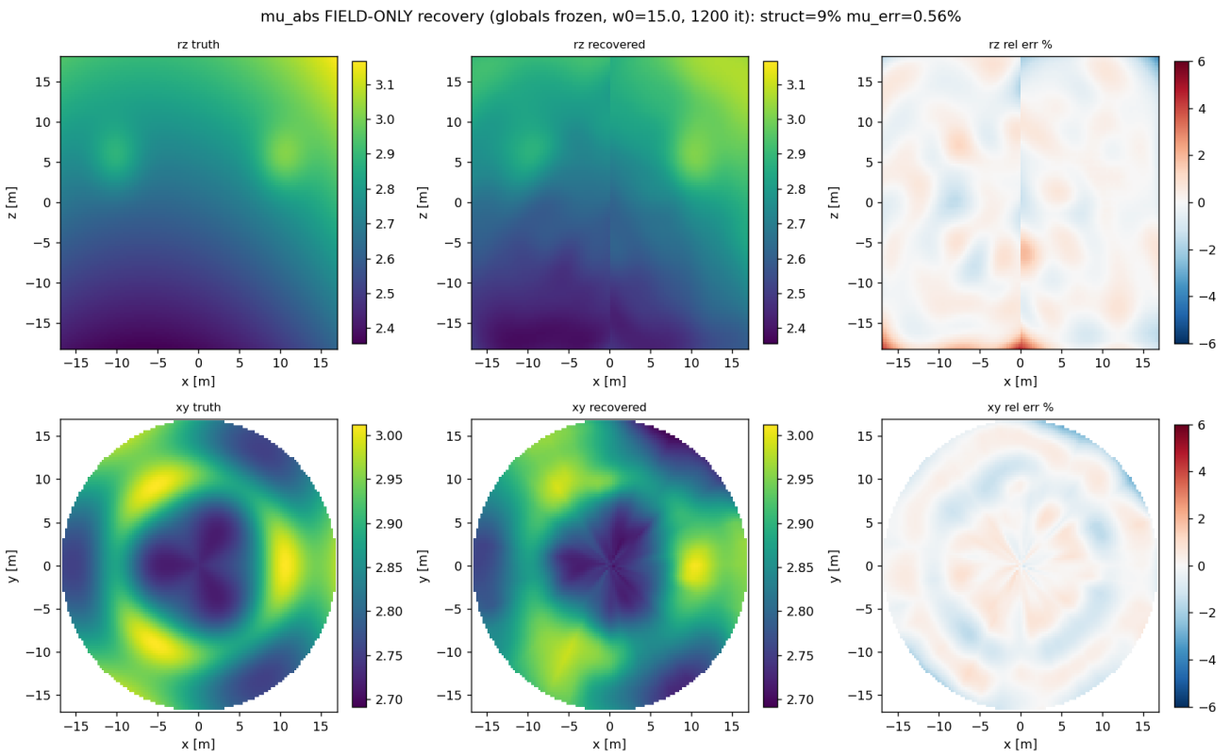
correlated valley: descent crawls

inverse Hessian reshapes it: descent goes straight

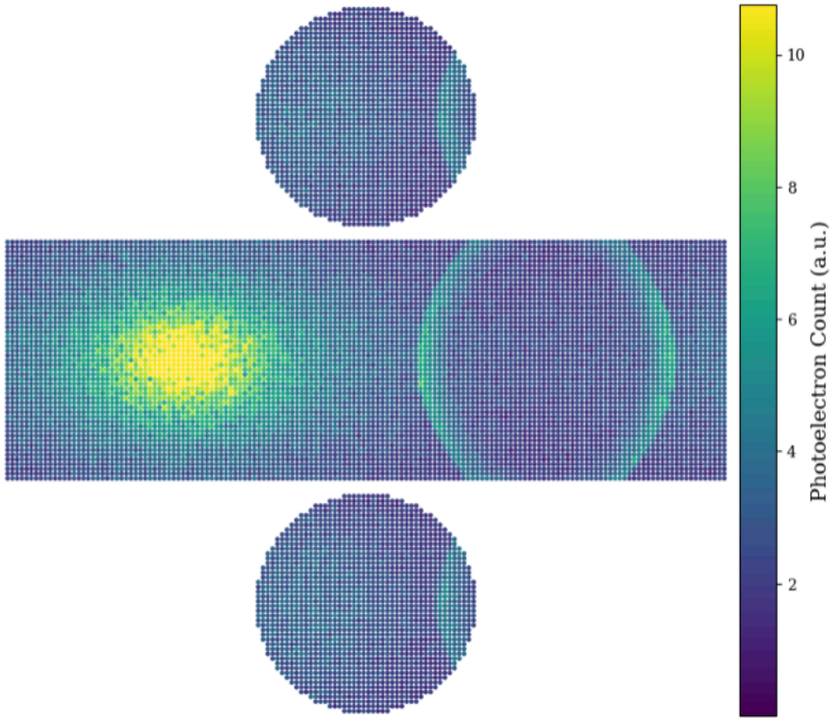
Optimisation moved to both Q and T likelihood loss.



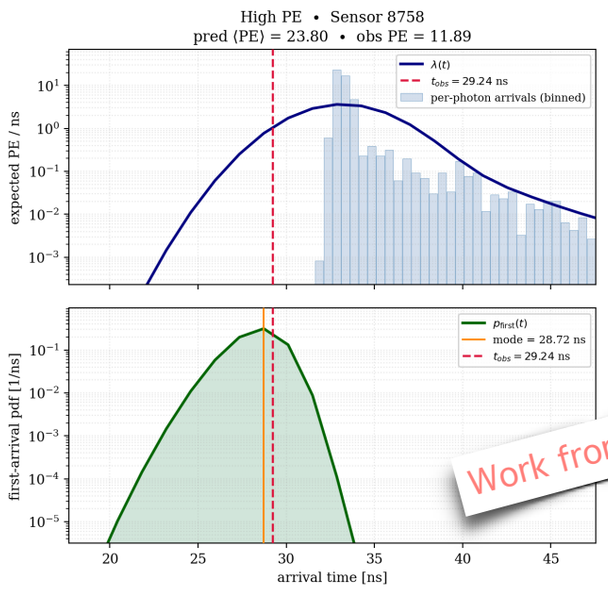
Learning heterogenous medium properties



Differentiable prediction for WbLS

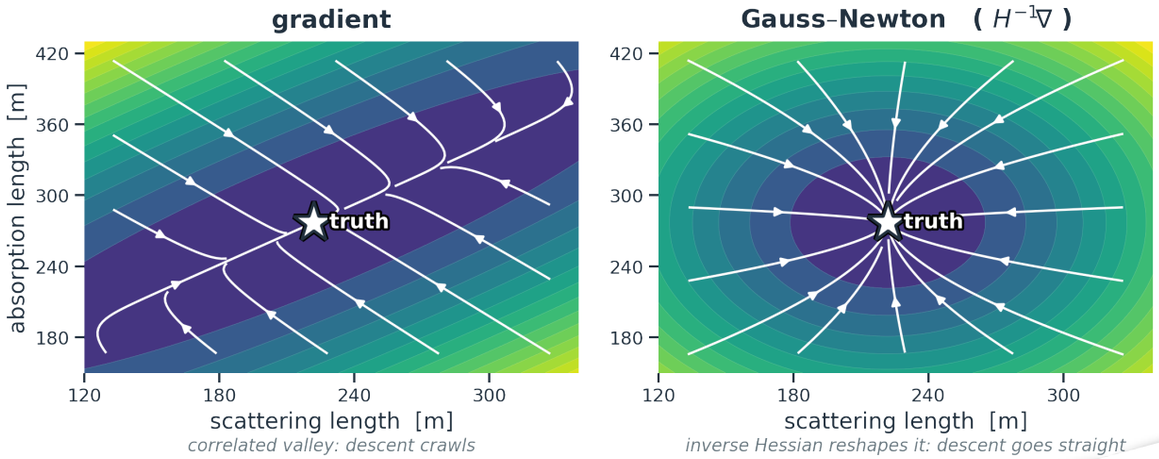


Optimisation moved to both Q and T likelihood loss.



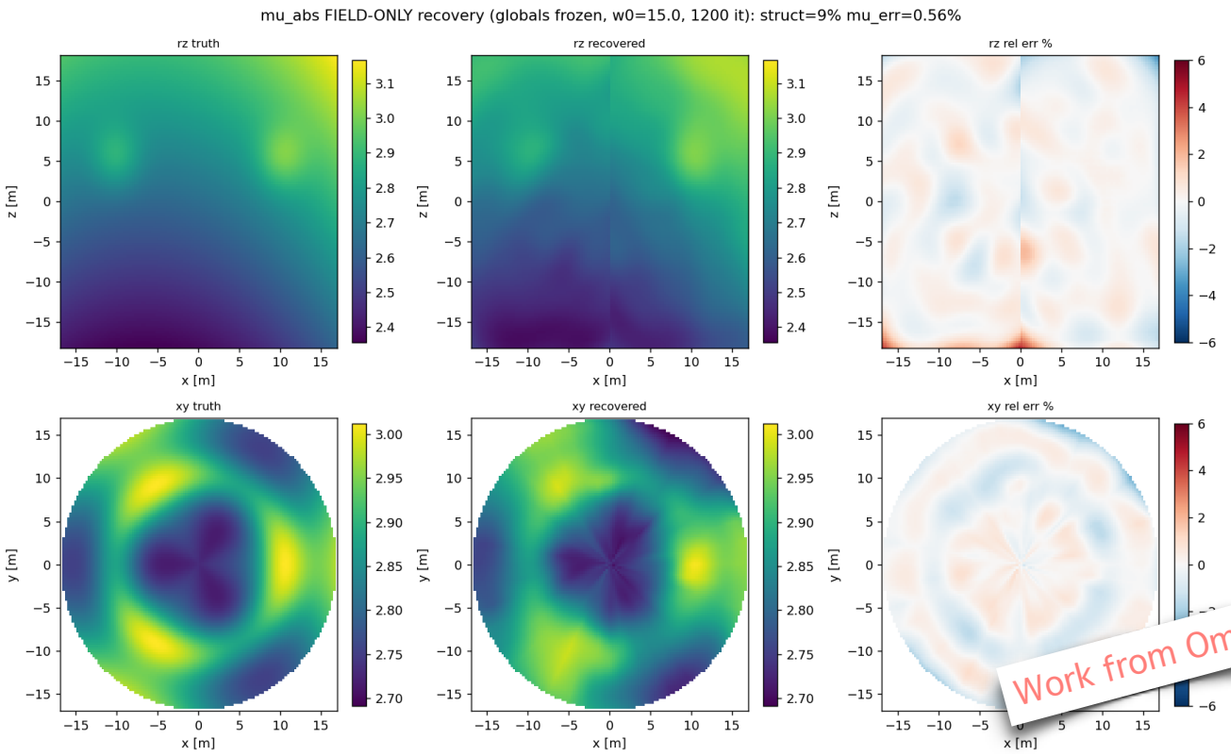
Work from Omar

Improving our optimisation landscape



Work from Omar

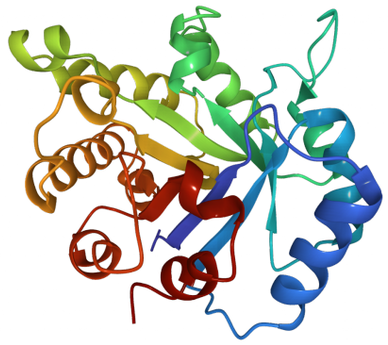
Learning heterogenous medium properties



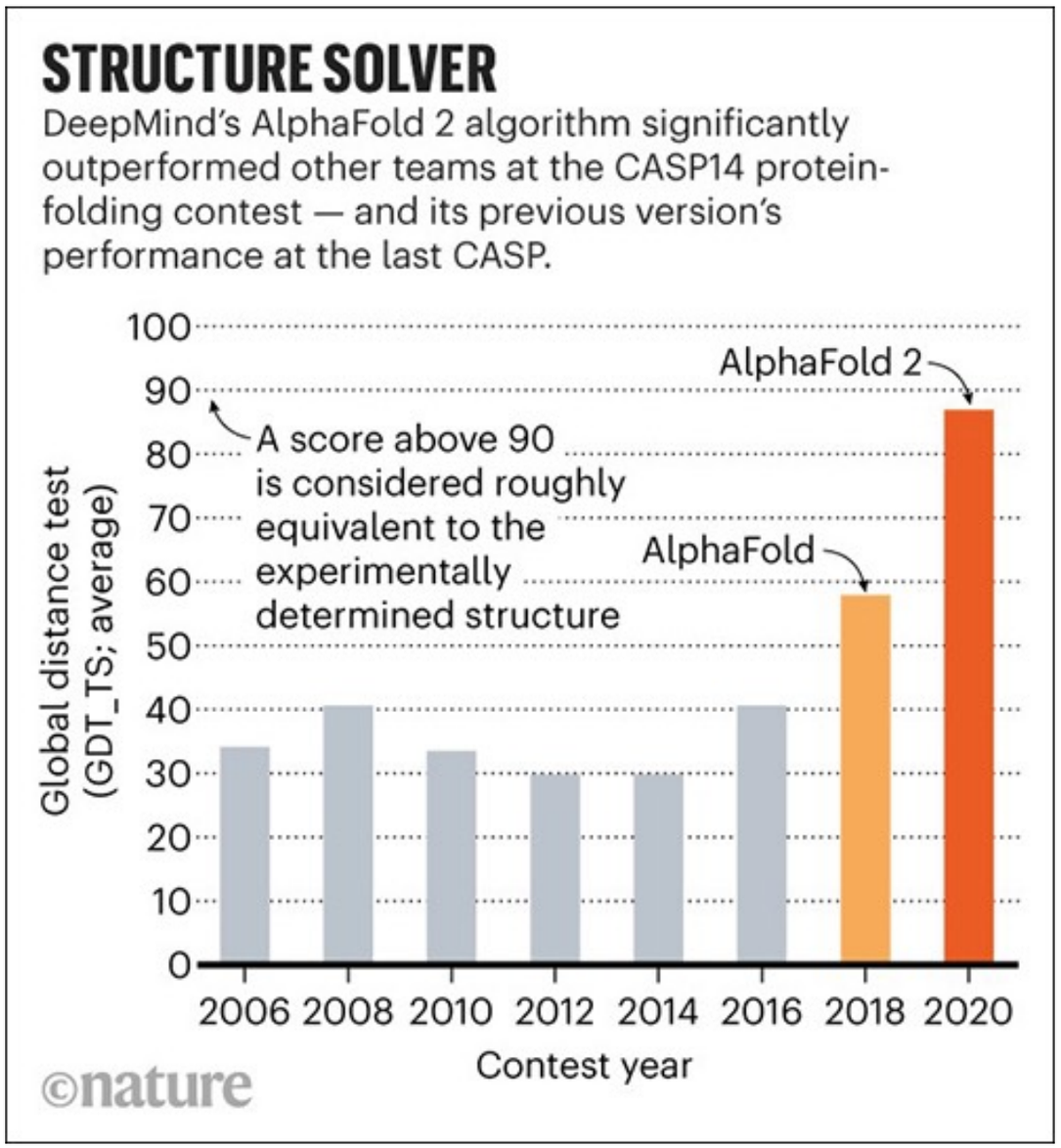
Work from Omar

The power of scientific datasets

Learning from a different field



A CHANGE OF PARADIGM: DATA > DOMAIN KNOWLEDGE



- ***In other domains, AI-tools have shown that, if good datasets exist, one can enable transformative performance boosts.***

- ***HEP is uniquely poised to generate massive datasets of exceptional quality.***

- ***HEP communities are often closed: Software and data are private → Surprising absence of good datasets/ benchmarks***

What can we do about it?

DATA IS OWNED BY COLLABORATIONS

"*Understandable*" but significant roadblock in the era of AI/ML

Common benchmarks and community challenges are CRITICAL to solve existing problems

How do we change this?!

Until we "fix" the data problem we should, at least, have common simulated benchmarks.

Who has time for that?!

Experiment simulations are also often "private". How do we generate realistic simulations?

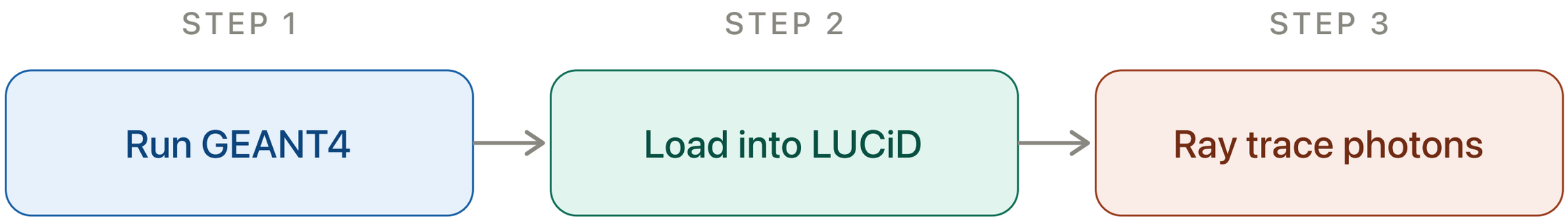
A success example

PILArNet released 6 years ago *https://arxiv.org/abs/2006.01993*

What about the rest of detector technologies?

Well we can't do everything but we could do "all" optical detectors with LUCiD...

HOW DOES THE PIPELINE WORK?

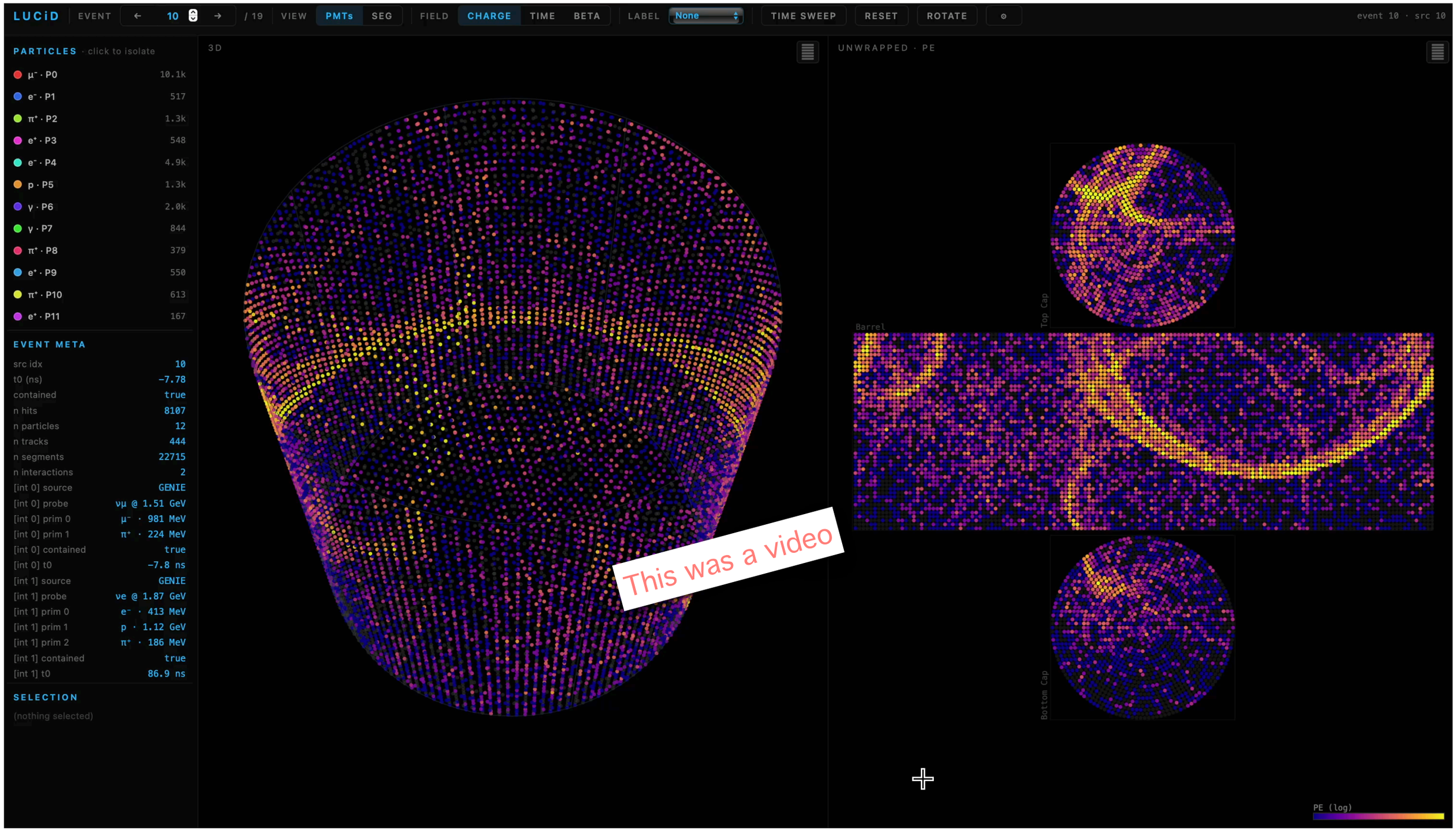


Cherenkov inject photon-by-photon
Scintillation create photons from G4 segments Edep.

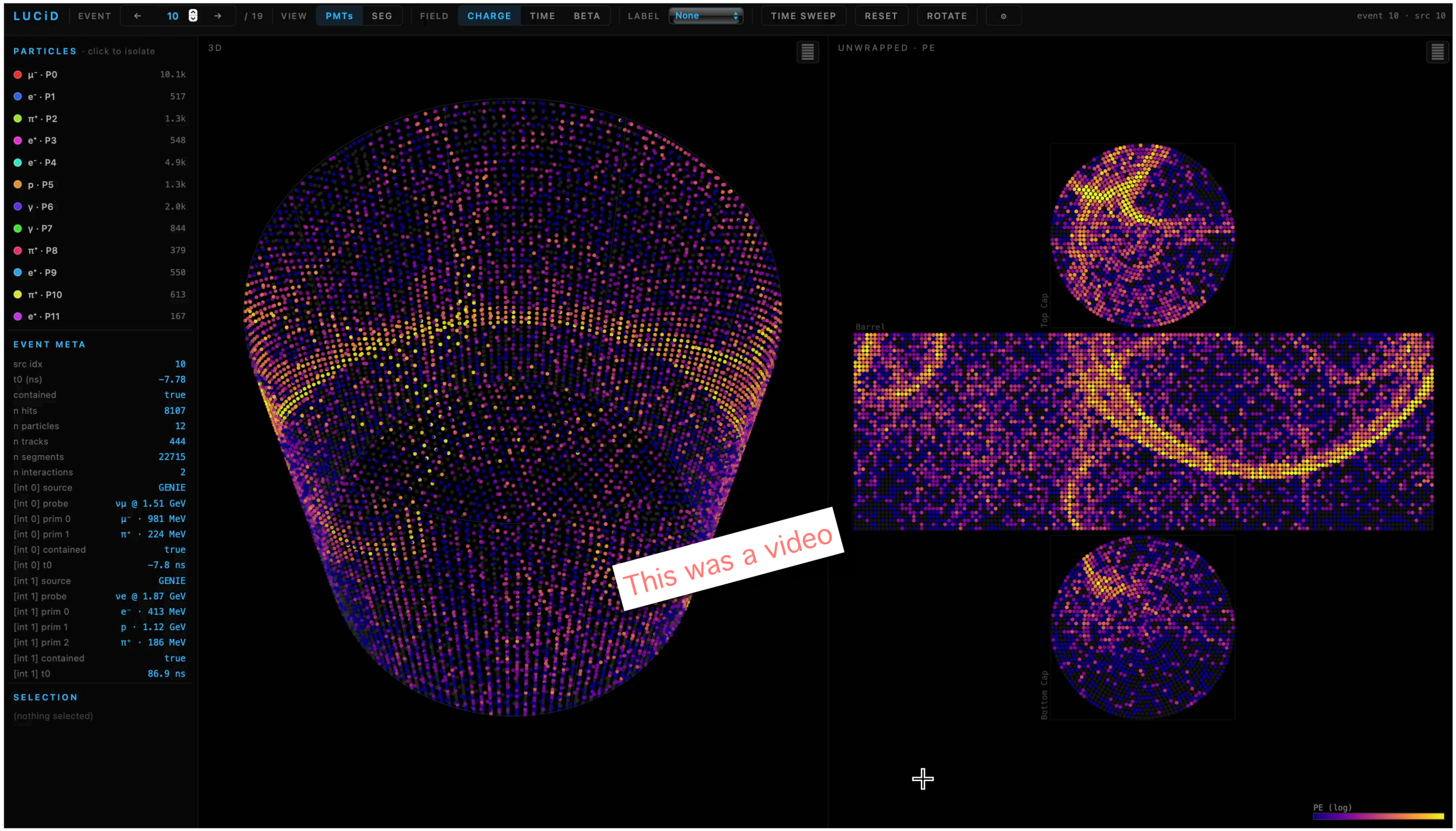
Propagate G4-level information into hits

```
dataset_root/
├── sensor/  sensor_NNNN.h5   →  the detector "image"  (model INPUT)
├── hits/    hits_NNNN.h5     →  input, but split by which particle made each photoelectron
├── step/    step_NNNN.h5     →  the Geant4 truth trajectory + exact per-step→sensor contribution
└── labl/    labl_NNNN.h5     →  the labels: particles, energies, vertex, PID, neutrino info
```

A SNEAK PEEK

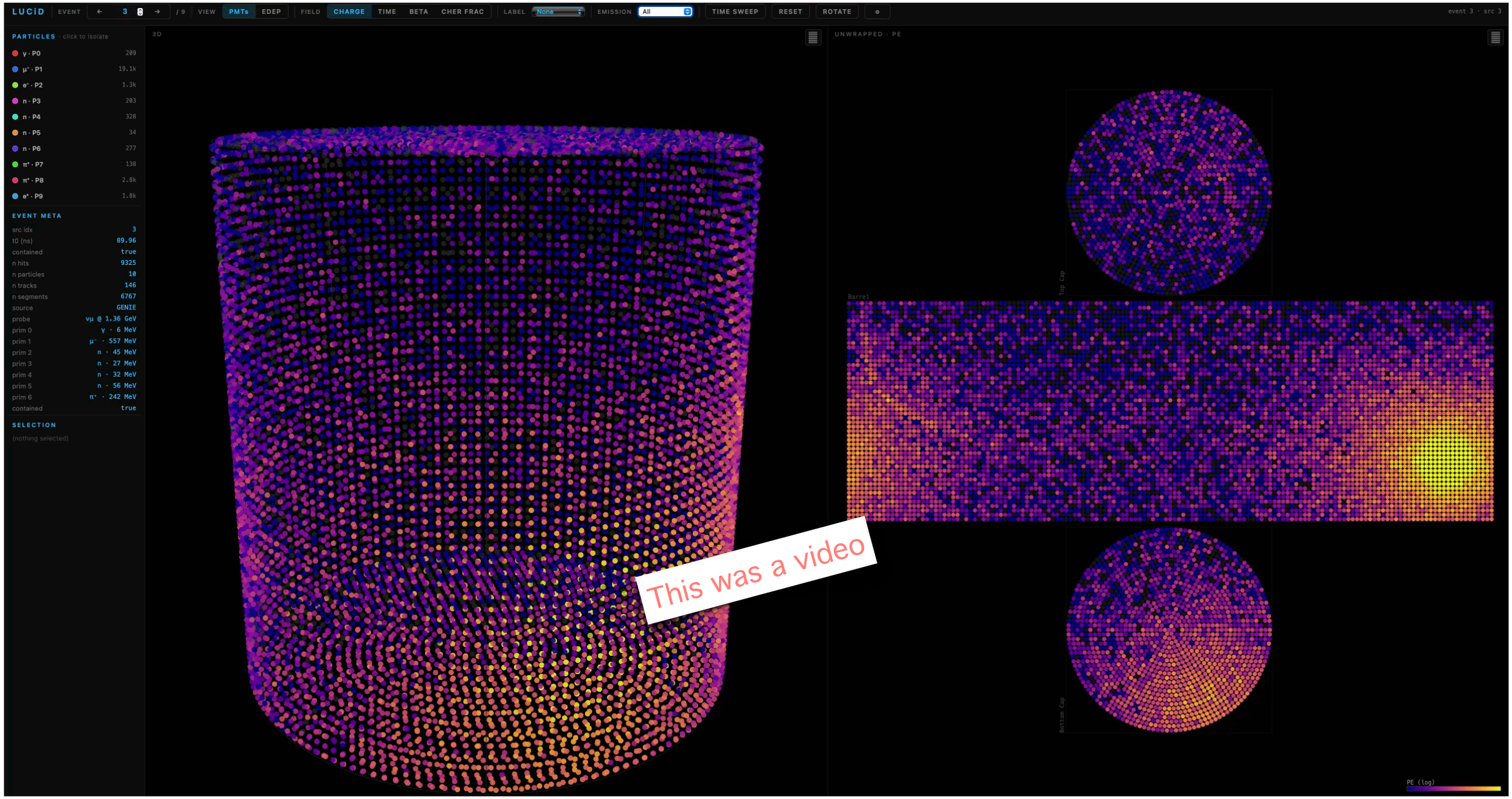

LUCiD
EVENT
10
/ 19
VIEW
PMTs
SEG
FIELD
CHARGE
TIME
BETA
LABEL
None
TIME SWEEP
RESET
ROTATE
event 10 · src 10
PARTICLES · click to isolate
μ⁻ · P0 10.1k
e⁻ · P1 517
π⁺ · P2 1.3k
e⁺ · P3 548
e⁻ · P4 4.9k
p · P5 1.3k
γ · P6 2.0k
γ · P7 844
π⁺ · P8 379
e⁺ · P9 550
π⁺ · P10 613
e⁺ · P11 167
EVENT META
src idx 10
t0 (ns) −7.78
contained true
n hits 8107
n particles 12
n tracks 444
n segments 22715
n interactions 2
[int 0] source GENIE
[int 0] probe νμ @ 1.51 GeV
[int 0] prim 0 μ⁻ · 981 MeV
[int 0] prim 1 π⁺ · 224 MeV
[int 0] contained true
[int 0] t0 −7.8 ns
[int 1] source GENIE
[int 1] probe νe @ 1.87 GeV
[int 1] prim 0 e⁻ · 413 MeV
[int 1] prim 1 p · 1.12 GeV
[int 1] prim 2 π⁺ · 186 MeV
[int 1] contained true
[int 1] t0 86.9 ns
SELECTION
(nothing selected)
3D
UNWRAPPED · PE
Top Cap
Barrel
Bottom Cap
PE (log)
This was a video

A SNEAK PEEK


LUCiD
EVENT
10
/ 19
VIEW
PMTs
SEG
FIELD
CHARGE
TIME
BETA
LABEL
None
TIME SWEEP
RESET
ROTATE
event 10 · src 10
PARTICLES · click to isolate
μ⁻ · P0 10.1k
e⁻ · P1 517
π⁺ · P2 1.3k
e⁺ · P3 548
e⁻ · P4 4.9k
p · P5 1.3k
γ · P6 2.0k
γ · P7 844
π⁺ · P8 379
e⁺ · P9 550
π⁺ · P10 613
e⁺ · P11 167
EVENT META
src idx 10
t0 (ns) −7.78
contained true
n hits 8107
n particles 12
n tracks 444
n segments 22715
n interactions 2
[int 0] source GENIE
[int 0] probe νμ @ 1.51 GeV
[int 0] prim 0 μ⁻ · 981 MeV
[int 0] prim 1 π⁺ · 224 MeV
[int 0] contained true
[int 0] t0 −7.8 ns
[int 1] source GENIE
[int 1] probe νe @ 1.87 GeV
[int 1] prim 0 e⁻ · 413 MeV
[int 1] prim 1 p · 1.12 GeV
[int 1] prim 2 π⁺ · 186 MeV
[int 1] contained true
[int 1] t0 86.9 ns
SELECTION
(nothing selected)
3D
UNWRAPPED · PE
Top Cap
Barrel
Bottom Cap
PE (log)
This was a video

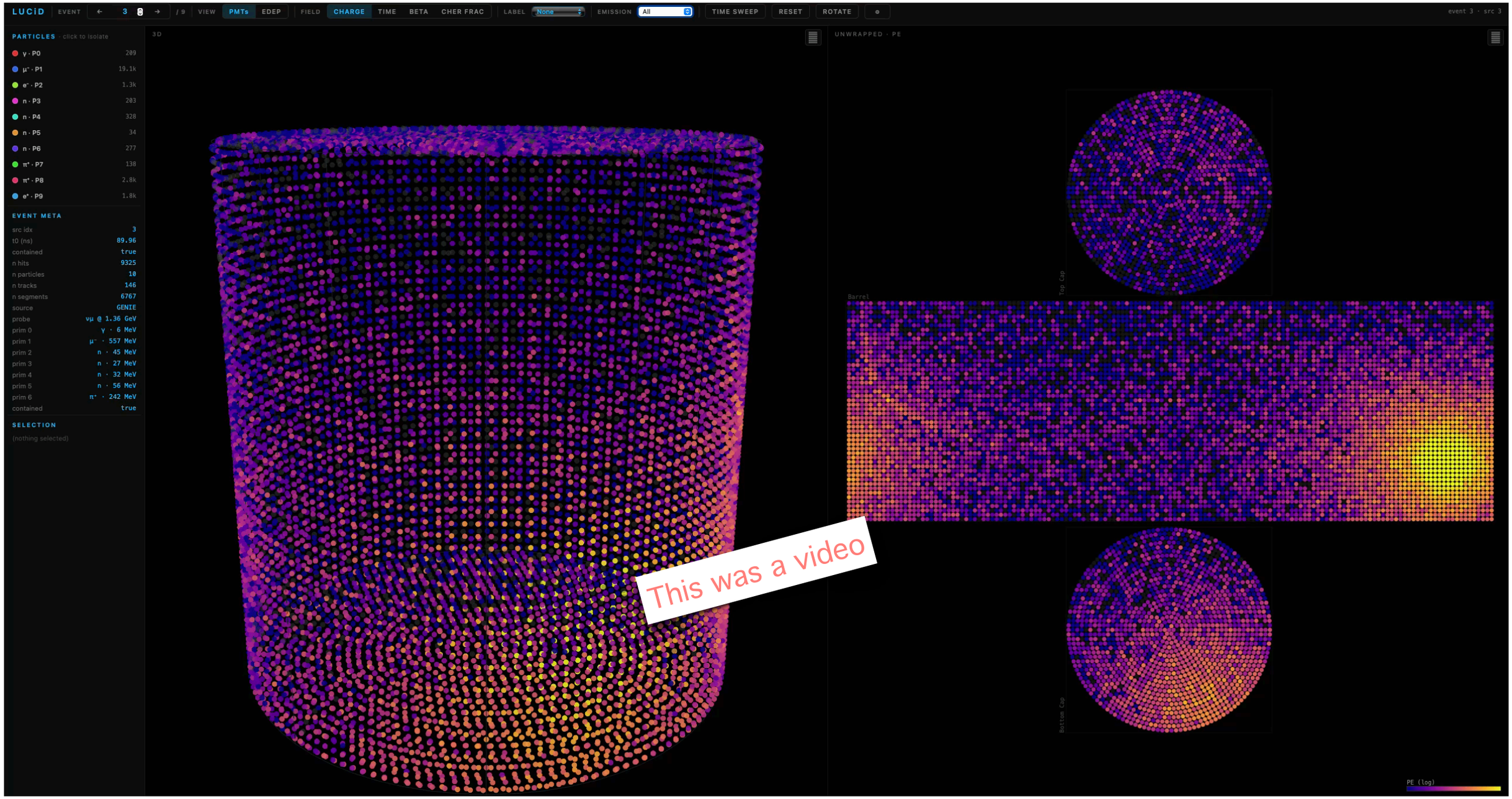
A SNEAK PEEK

CERN



A SNEAK PEEK

CERN



LUCiD | EVENT ← 7 → / 49 | VIEW PMTs EDEP | FIELD CHARGE TIME BETA CHER FRAC | LABEL Particle | EMISSION All | TIME SWEEP | RESET | ROTATE | ⚙ | event 7 · sr…

PARTICLES · click to isolate

γ · P0	208
μ⁻ · P1	41.5k
n · P2	630
π⁺ · P3	6.4k
π⁺ · P4	875
π⁺ · P5	3.2k
π⁰ · P6	23.5k
π⁰ · P7	83.7k

EVENT META

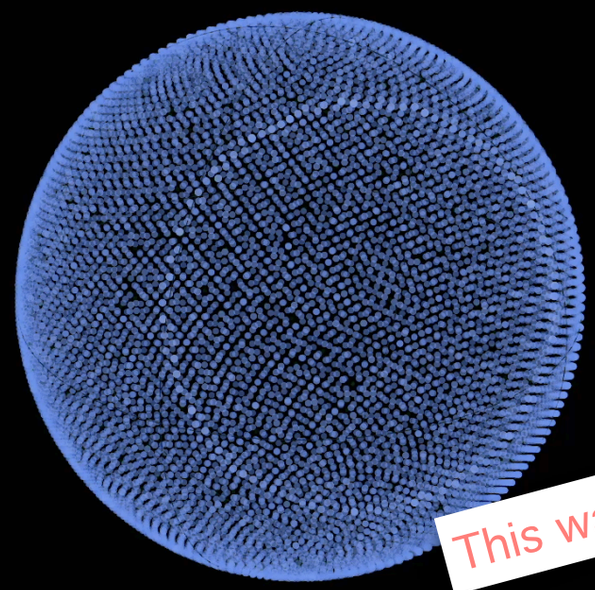
src idx	7
t0 (ns)	77.58
contained	true
n hits	10000
n particles	8
n tracks	756
n segments	33815
source	GENIE
probe	νμ @ 4.49 GeV
prim 0	γ · 6 MeV
prim 1	μ⁻ · 814 MeV
prim 2	n · 463 MeV
prim 3	π⁺ · 813 MeV
prim 4	π⁰ · 2.00 GeV
contained	true

SELECTION

μ⁻ · P1

pdg bucket	μ⁻
contained	true
Σ PE (hits)	41.5k
sensors hit	9794
n tracks	108
n Cherenkov	158751
max init E	813.5 MeV
PDG	μ⁻, e⁻, γ
genealogy	2

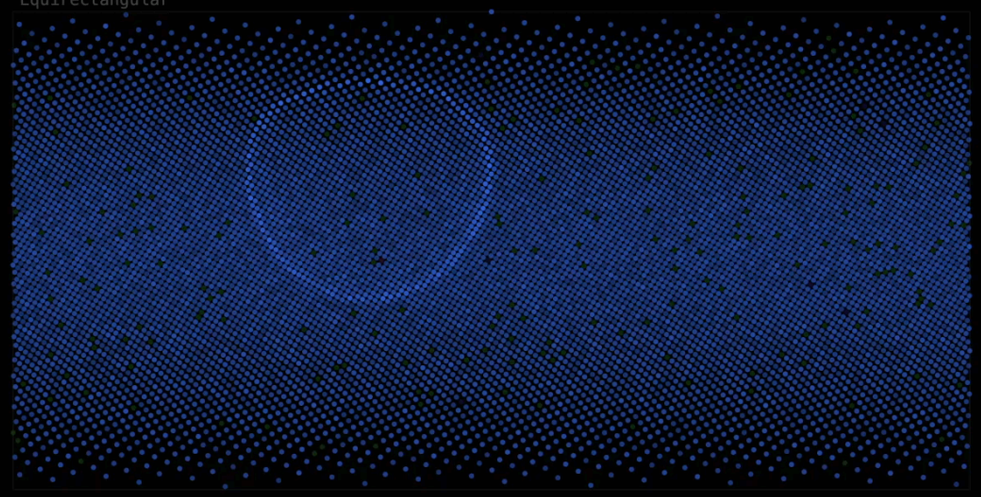
3D



This was a video

UNWRAPPED · PARTICLE

Equirectangular



LUCiD | EVENT ← 7 → / 49 | VIEW PMTs EDEP | FIELD CHARGE TIME BETA CHER FRAC | LABEL Particle | EMISSION All | TIME SWEEP | RESET | ROTATE | ⚙ | event 7 · sr…

PARTICLES · click to isolate

γ · P0	208
μ⁻ · P1	41.5k
n · P2	630
π⁺ · P3	6.4k
π⁺ · P4	875
π⁺ · P5	3.2k
π⁰ · P6	23.5k
π⁰ · P7	83.7k

EVENT META

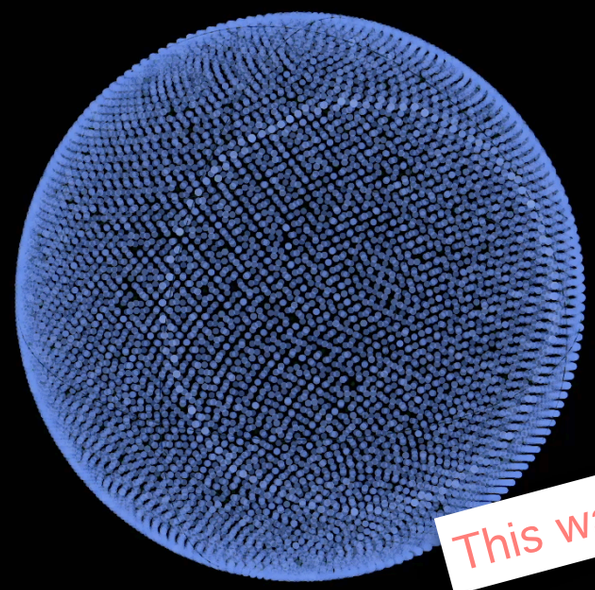
src idx	7
t0 (ns)	77.58
contained	true
n hits	10000
n particles	8
n tracks	756
n segments	33815
source	GENIE
probe	νμ @ 4.49 GeV
prim 0	γ · 6 MeV
prim 1	μ⁻ · 814 MeV
prim 2	n · 463 MeV
prim 3	π⁺ · 813 MeV
prim 4	π⁰ · 2.00 GeV
contained	true

SELECTION

μ⁻ · P1

pdg bucket	μ⁻
contained	true
Σ PE (hits)	41.5k
sensors hit	9794
n tracks	108
n Cherenkov	158751
max init E	813.5 MeV
PDG	μ⁻, e⁻, γ
genealogy	2

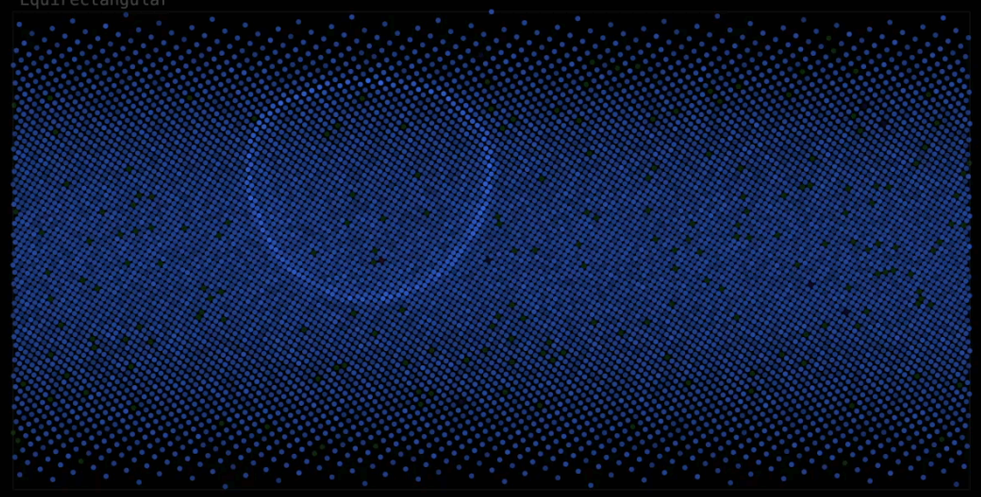
3D

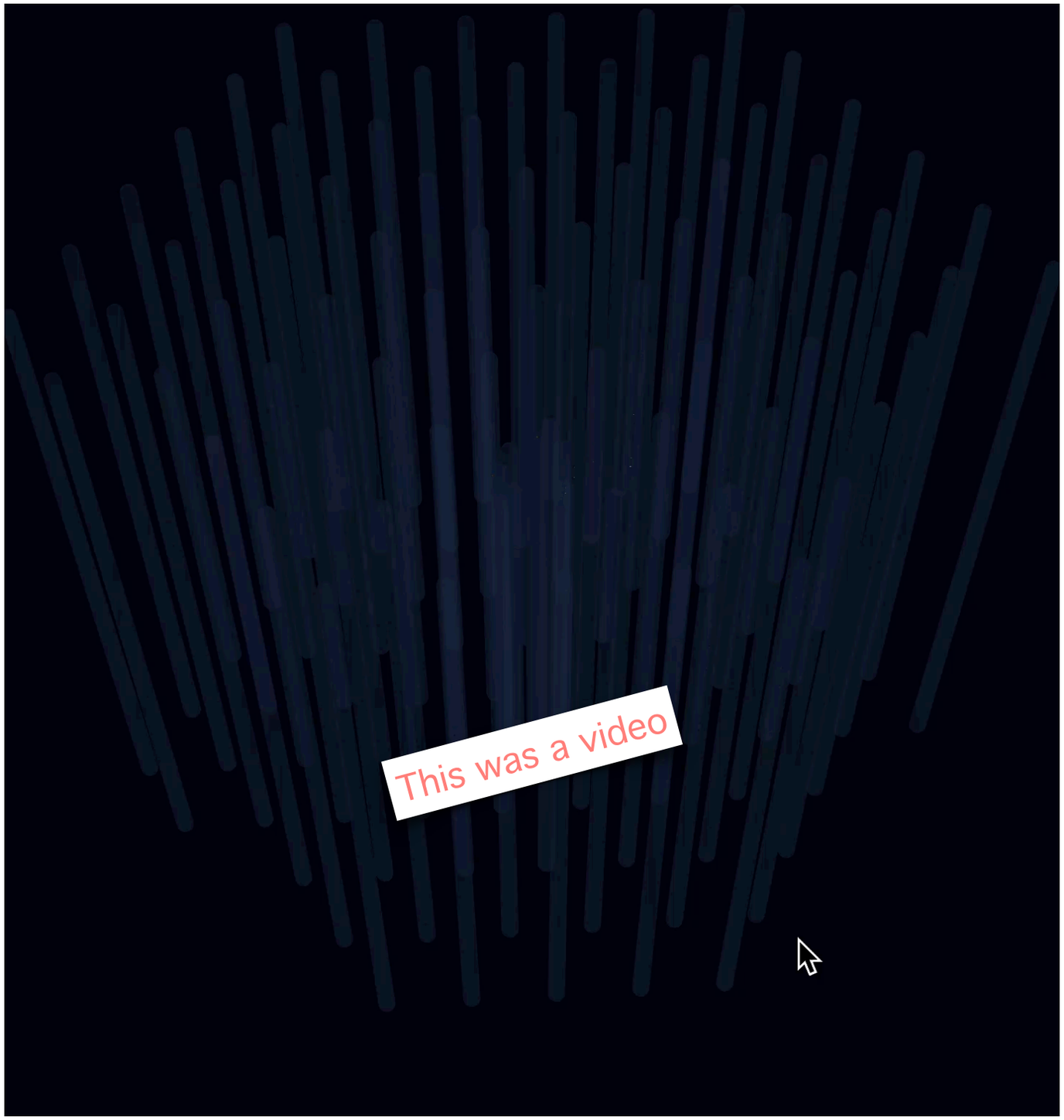


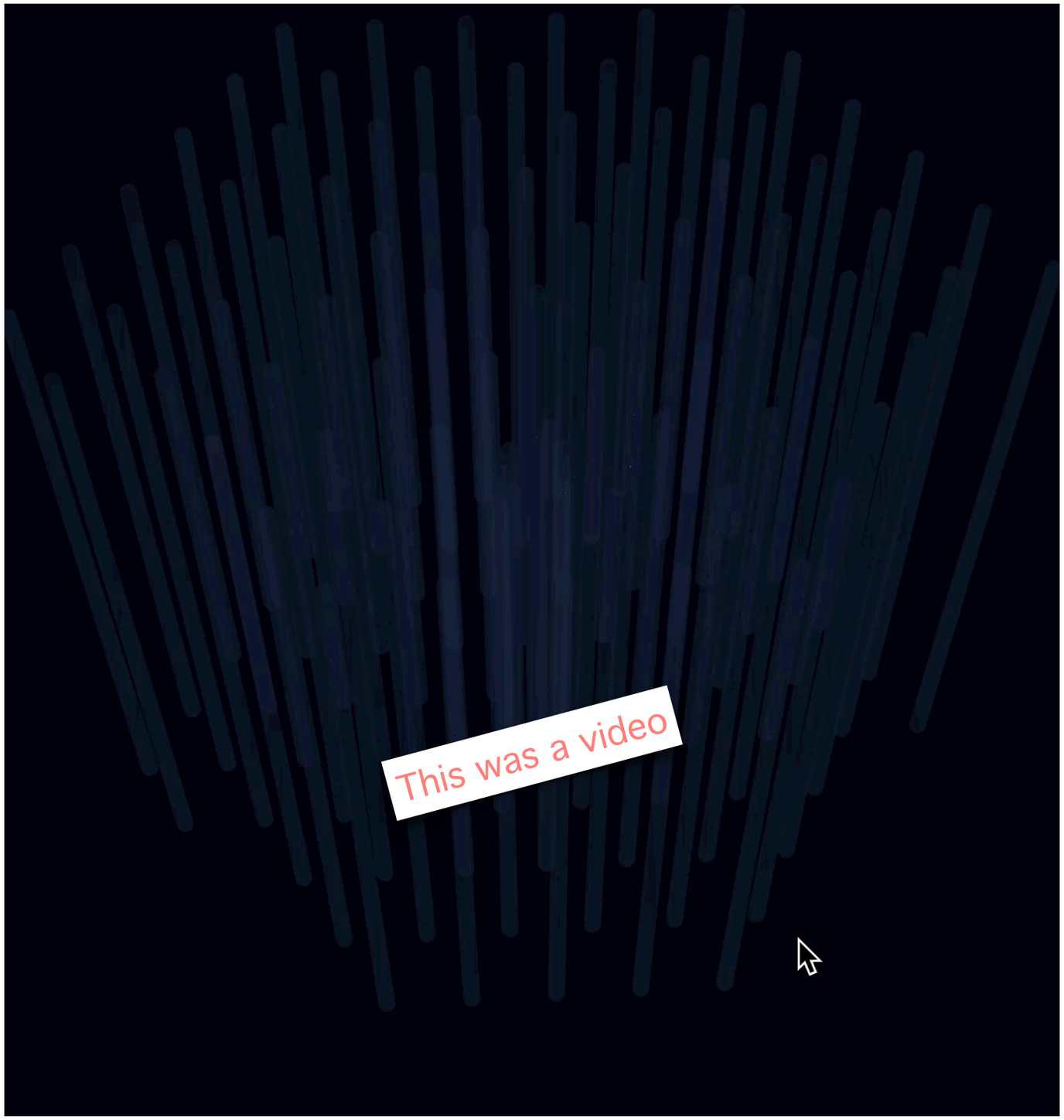
This was a video

UNWRAPPED · PARTICLE

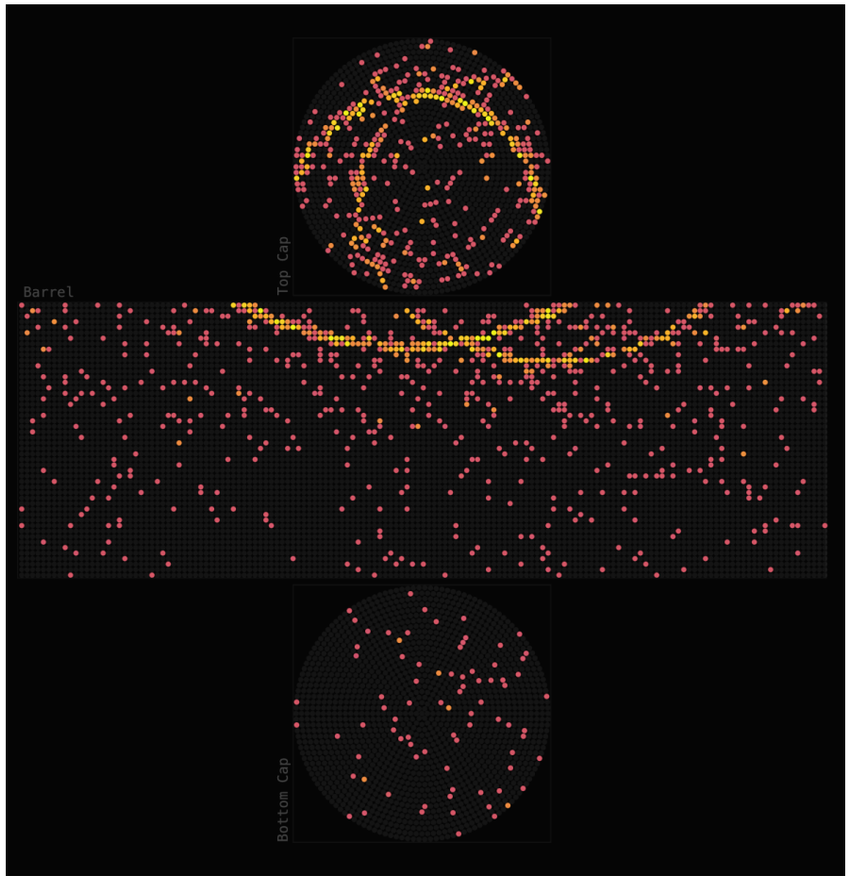
Equirectangular




This was a video


This was a video

What is the right way to define 'particle labels'?

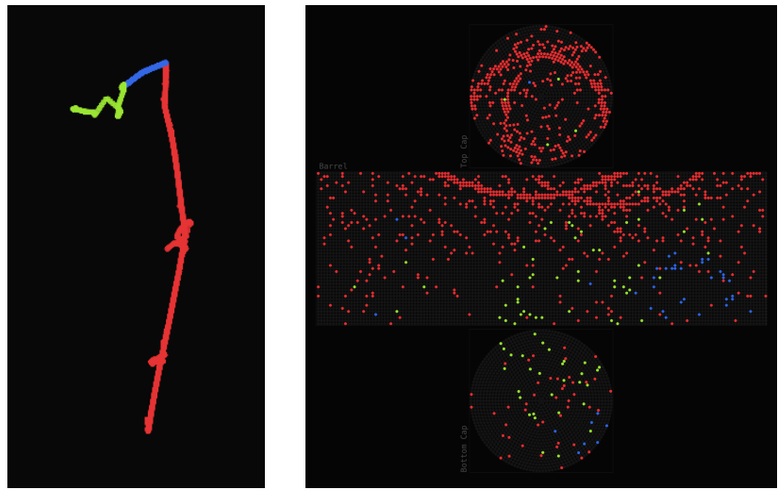


There is no obvious answer!

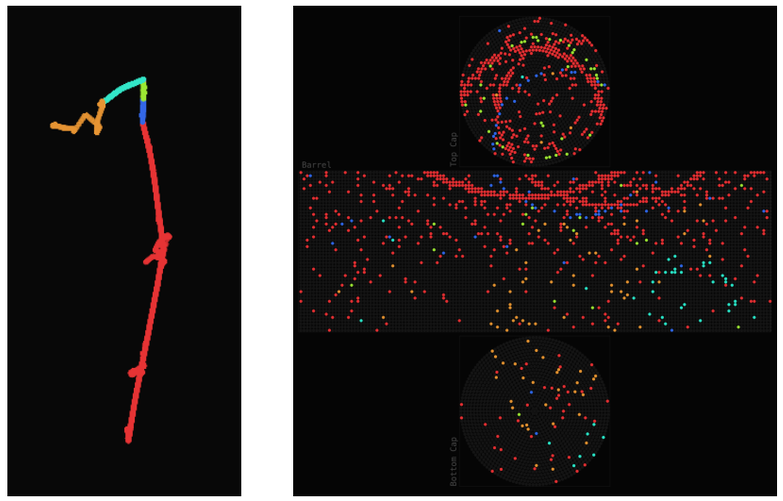
Our dataset allow to define labels as <u>derived views</u>

If in the future we decide we need more/less granularity we can redefine the labels without having to re-generate the events

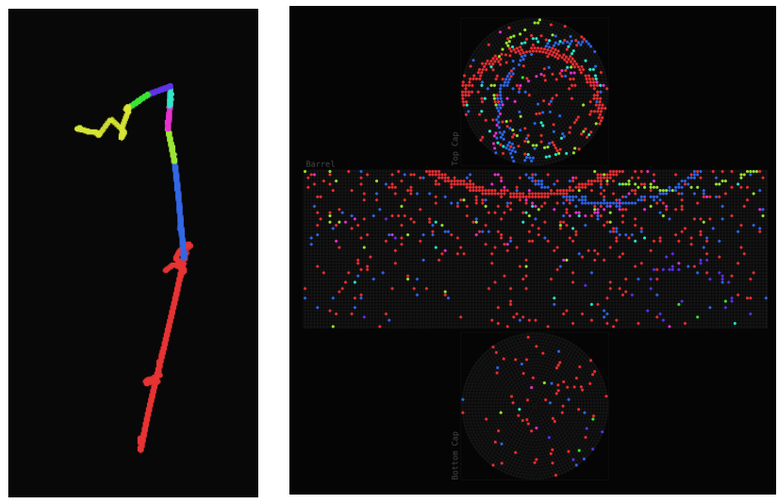
A)



B)



C)



DORAEMON | Open Dataset Challenge

OpenDC Data Hub Documentation Software Updates

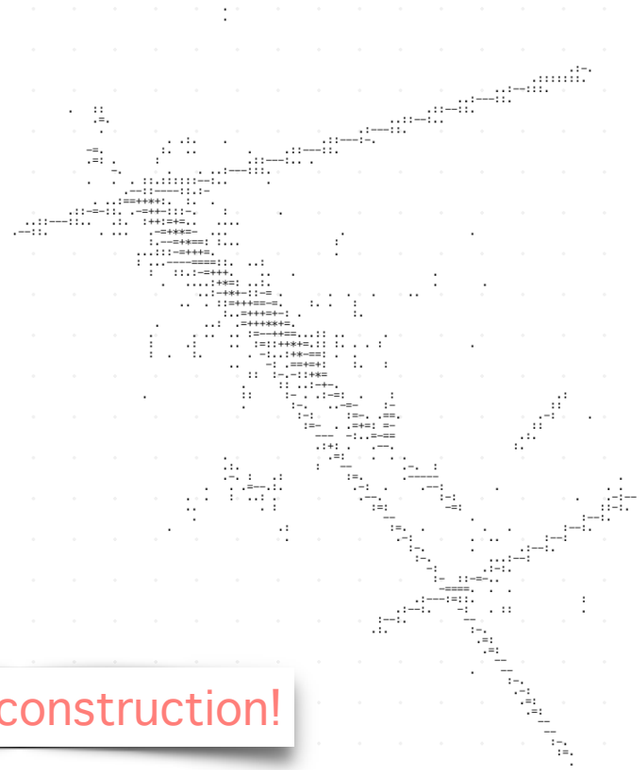
DORAEMON Open Dataset Challenge

Public simulated datasets and shared evaluation tasks for AI methods in neutrino physics.

Explore OpenDC | Read documentation

https://www.deeplearnphysics.org/doraemon_site/

Site under construction!



A NEW APPROACH

- *A site to coordinate distribution of datasets, publication of challenges and evaluation of benchmarks.*

See the new talk from Kazu!

DATASETS ARE USELESS UNLESS THEY TARGET A COMMUNITY CHALLENGE

What things do we care about in water Cherenkov detectors?

- **Flavour ID: μ vs e. MIP PID: μ vs charge $\pi^{\pm}$. EM PID: e vs π^{0}.**

Two environments: 1) How well can we do this in single particle events

2) How well can we do this in multi-particle events

Defines 'ultimate possible performance'

The thing we actually care about

DATASETS ARE USELESS UNLESS THEY TARGET A COMMUNITY CHALLENGE

What things do we care about in water Cherenkov detectors?

- **Flavour ID: μ vs e. MIP PID: μ vs charge $\pi^{\pm}$. EM PID: e vs π^{0}.**

Two environments: 1) How well can we do this in single particle events

2) How well can we do this in multi-particle events

Defines 'ultimate possible performance'

The thing we actually care about

- **Regression Tasks (Particle energy and direction) for leptons and hadrons.**

Same datasets as above. Same data, different goal.

DATASETS ARE USELESS UNLESS THEY TARGET A COMMUNITY CHALLENGE

What things do we care about in water Cherenkov detectors?

- **Flavour ID: μ vs e. MIP PID: μ vs charge $\pi^{\pm}$. EM PID: e vs π^{0}.**

Two environments: 1) How well can we do this in single particle events

2) How well can we do this in multi-particle events

Defines 'ultimate possible performance'

The thing we actually care about

- **Regression Tasks (Particle energy and direction) for leptons and hadrons.**

Same datasets as above. Same data, different goal.

- **Pile-up: Relevant for near detectors (e.g. IWCD @HK).**

How well can we separate pile-up?

DATASETS ARE USELESS UNLESS THEY TARGET A COMMUNITY CHALLENGE

What things do we care about in water Cherenkov detectors?

- **Flavour ID: μ vs e. MIP PID: μ vs charge $\pi^{\pm}$. EM PID: e vs π^{0}.**

Two environments: 1) How well can we do this in single particle events

2) How well can we do this in multi-particle events

Defines 'ultimate possible performance'

The thing we actually care about

- **Regression Tasks (Particle energy and direction) for leptons and hadrons.**

Same datasets as above. Same data, different goal.

- **Pile-up: Relevant for near detectors (e.g. IWCD @HK).**

How well can we separate pile-up?

- **$\nu \leftrightarrow$ particles & flavour ID in high-energy neutrino interactions**

Can we learn how to identify ν_τ? Single shot to flavor? Find exclusive final-states?

Can we learn robust observables that inform about the incident neutrino kinematics?

Ongoing discussion! We must be EXTREMELY careful to be robust to modelling biases

DATASETS ARE USELESS UNLESS THEY TARGET A COMMUNITY CHALLENGE

What things do we care about in water Cherenkov detectors?

Currently not thought to be part of WAND, but discussions are also ongoing...

- **Calibration?**

  We can generate calibration samples (using various calibration sources) and make a challenge to determine various calibration parameters (e.g. properties of the optical medium, sensor efficiency... etc)

- **Data-Sim shifts?**

  We can introduce intentional "unknown" shifts in datasets A, B

  The challenge is to think of what shifts present meaningful challenges and what are useful performance metrics.

Currently using SK-like geometry for all, it might change (HK-like, IWCD)

#	Name	Particles	Energy	Type
1	mu	μ^-	0.2 – 2 GeV	single particle
2	pi_plus	π^+	0.2 – 2 GeV	single particle
3	e	e^-	1 MeV – 2 GeV	single particle
4	pi_minus	π^-	0.2 – 2 GeV	single particle
5	pi0	π^0	0.2 – 2 GeV	single particle
6	e_low_energy	e^-	1 – 20 MeV	single particle
7	mu_pi_plus	$\mu^- + \pi^+$	0.2 – 2 GeV	multi-particle
8	e_pi_plus	$e^- + \pi^+$	e: 1 MeV–2 GeV, π: 0.2–2 GeV	multi-particle
9	e_pi0	$e^- + \pi^0$	e: 1 MeV–2 GeV, π: 0.2–2 GeV	multi-particle
10	mu_pi_plus_pi0	$\mu^- + \pi^+ + \pi^0$	0.2 – 2 GeV	multi-particle
11	mu_pi_plus_pi_minus	$\mu^- + \pi^+ + \pi^-$	0.2 – 2 GeV	multi-particle
12	e_pi_plus_pi0	$e^- + \pi^+ + \pi^0$	e: 1 MeV–2 GeV, π: 0.2–2 GeV	multi-particle
13	numu	ν_μ on water (CC+NC)	0.1 – 2 GeV	neutrino (GENIE)
14	nue	ν_e on water (CC+NC)	0.1 – 2 GeV	neutrino (GENIE)
15	pile_up	gun + neutrino, overlapping	mixed	pile-up (2 vertices)
16	pile_up_particles	gun + gun, overlapping	mixed	pile-up (2 vertices)
17	pile_up_genie	neutrino + neutrino	mixed	pile-up (2 vertices)
18	pile_up_bombs	random multi-particle ×2	mixed	pile-up (2 vertices)

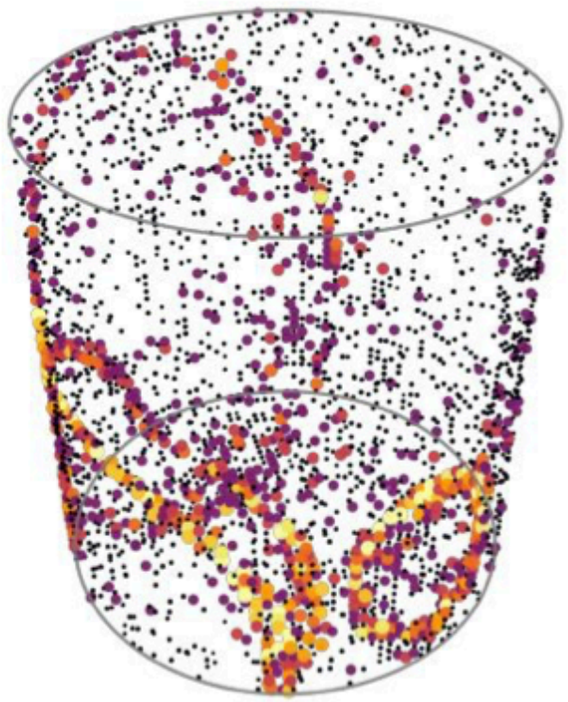
will probably stay as they are

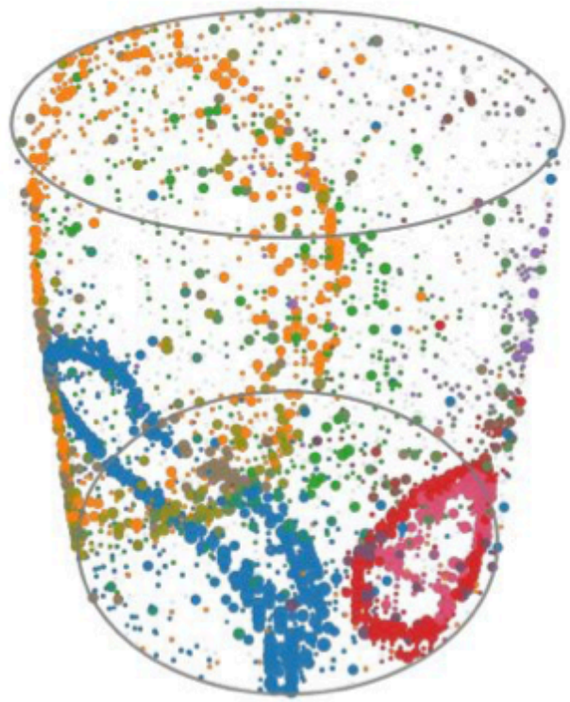
might only be used as evaluation datasets

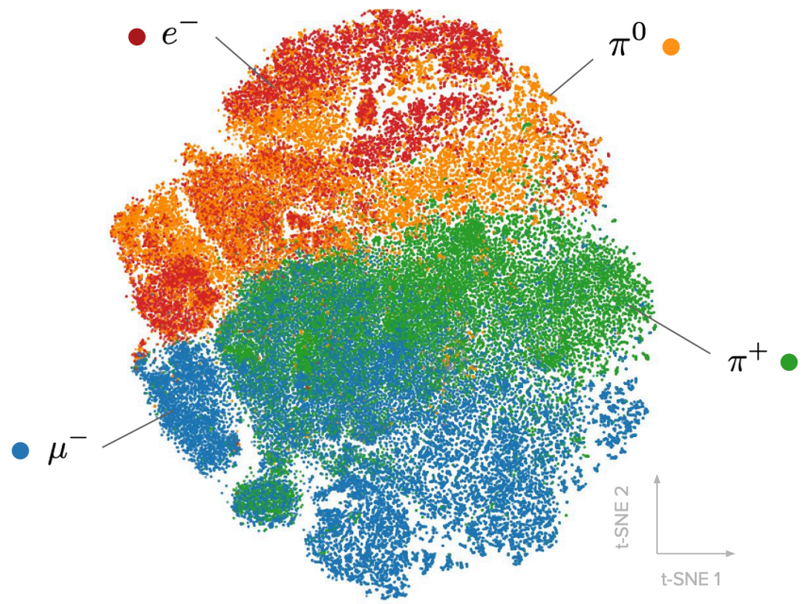
will likely extend to other generators / models

likely only 16 & 18 will be kept

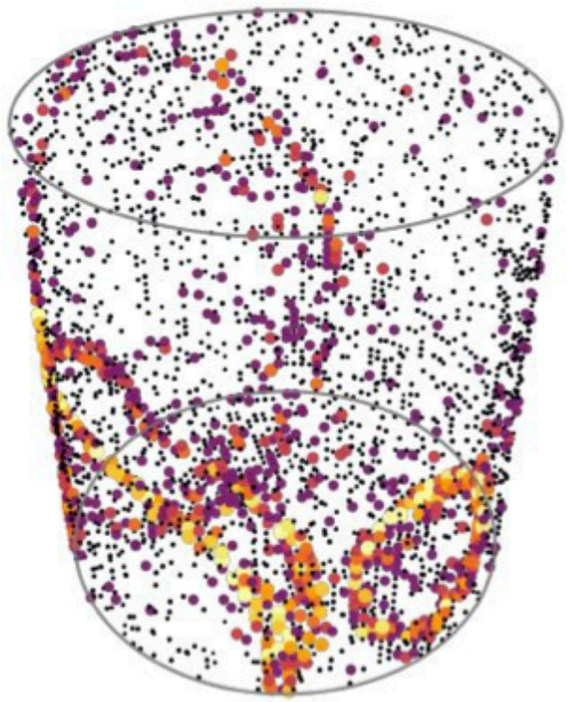
Our intention is to release baseline benchmarks for all challenges

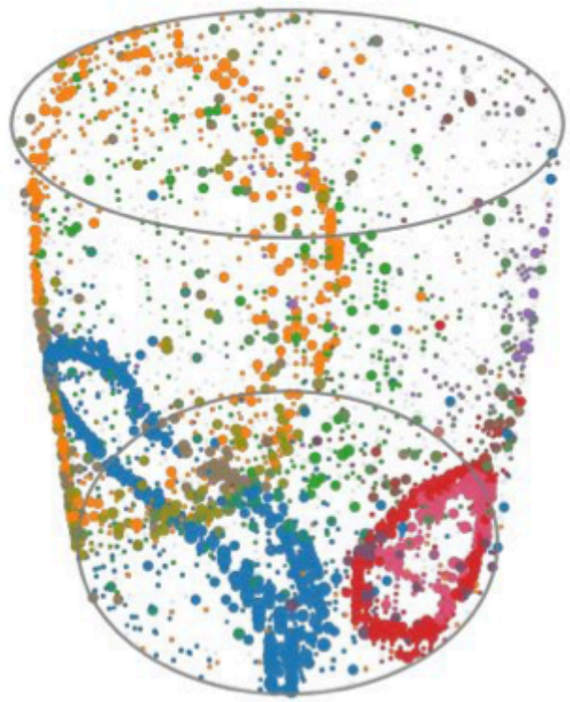


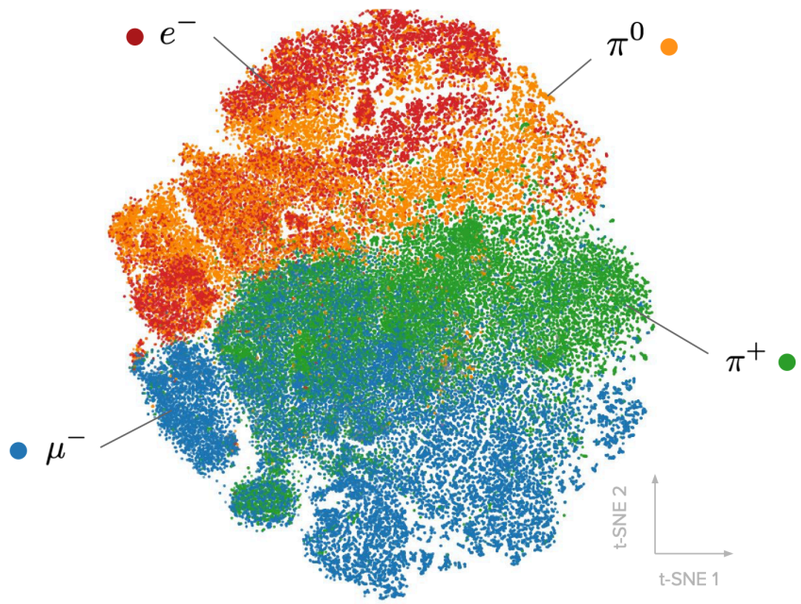




- *Sam Young (see his Talk!) is already using config 18 and showed very promising results yesterday.*







- *Sam Young (see his Talk!) is already using config 18 and showed very promising results yesterday.*

Important!

- *If you have in mind a challenge you want to choose and think we could generate it with LUCiD get in touch with us, we are very interested in supporting you!*

cesar.jesus@cern.ch omar.alterkait@tufts.edu

Conclusions

Were are we now?

- **LUCiD (differentiability):** Consolidated as a flexible & robust end-to-end differentiable tool capable of optimising calibration parameters and model Cherenkov and scintillation light across most common detector geometries in the field.

ToDo (ongoing): Demonstrate differentiability for scintillation parameters (calibration task). Demonstrate how to learn heterogeneous medium properties. Move away from FitQun-like parametrization (straight-track assumption). Extend to full physics differentiable chain.

Challenge: Add differentiable GEANT4 & neutrino interaction generator :)

Were are we now?

- **LUCiD (differentiability):** Consolidated as a flexible & robust end-to-end differentiable tool capable of optimising calibration parameters and model Cherenkov and scintillation light across most common detector geometries in the field.

ToDo (ongoing): Demonstrate differentiability for scintillation parameters (calibration task). Demonstrate how to learn heterogeneous medium properties. Move away from FitQun-like parametrization (straight-track assumption). Extend to full physics differentiable chain.

Challenge: Add differentiable GEANT4 & neutrino interaction generator :) **A big goal for NPML 2027!**

Were are we now?

- **LUCiD (differentiability):** Consolidated as a flexible & robust end-to-end differentiable tool capable of optimising calibration parameters and model Cherenkov and scintillation light across most common detector geometries in the field.

ToDo (ongoing): Demonstrate differentiability for scintillation parameters (calibration task). Demonstrate how to learn heterogeneous medium properties. Move away from FitQun-like parametrization (straight-track assumption). Extend to full physics differentiable chain.

Challenge: Add differentiable GEANT4 & neutrino interaction generator :) **A big goal for NPML 2027!**

Were are we now?

- **LUCiD (differentiability):** Consolidated as a flexible & robust end-to-end differentiable tool capable of optimising calibration parameters and model Cherenkov and scintillation light across most common detector geometries in the field.

  ToDo (ongoing): Demonstrate differentiability for scintillation parameters (calibration task). Demonstrate how to learn heterogeneous medium properties. Move away from FitQun-like parametrization (straight-track assumption). Extend to full physics differentiable chain.

  Challenge: Add differentiable GEANT4 & neutrino interaction generator :) **A big goal for NPML 2027!**

- **LUCiD (pseudo-data):** Now fully supporting generation of pseudo-data (WAND-like) but also of WbLS and Scintillation detectors across arbitrary geometries.

  ToDo: Decide on final size and configurations of WAND. Upload them to DORAEMON. Improve scalability for light-propagation for telescopes.

  Challenge: Provide baseline benchmarks for WAND. Release other data challenges.

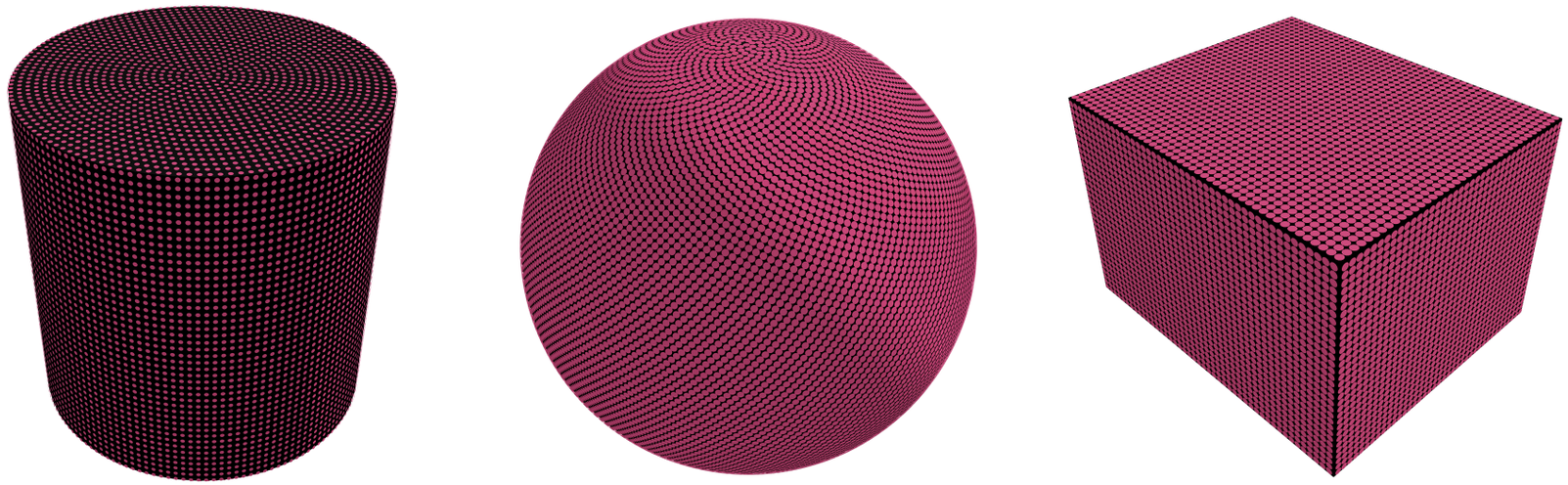
  Another big goal for NPML 2027!

Extra Slides

File	Think of it as	Contents
sensor	a **sparse detector image**	lit PMTs: `sensor_idx`, charge `PE`, time `T`. PMT xyz in `config/sensor_positions`.
hits	the signal, **truth-attributed**	each photoelectron tagged with its particle (`particle_idx`) and **Cherenkov vs scintillation** (`emission_process`). Pre-smearing.
step	the **ground-truth "why"**	Geant4 trajectory as segments (pos, dir, energy deposit, β…) **plus** the exact charge each segment put on each PMT (`sensor_hits`). Finest truth.
labl	your **targets / metadata**	per-event, per-interaction (vertex, neutrino PID & energy), per-particle (PID), per-track tables.

CURRENTLY SUPPORTS MOST COMMON DETECTOR GEOMETRIES: CYLINDERS, SPHERES AND BOXES

LUCiD



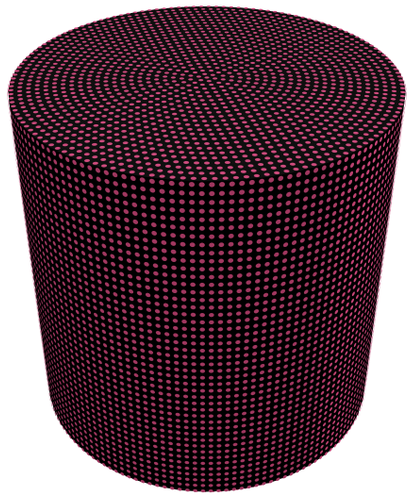
KEY ASPECTS

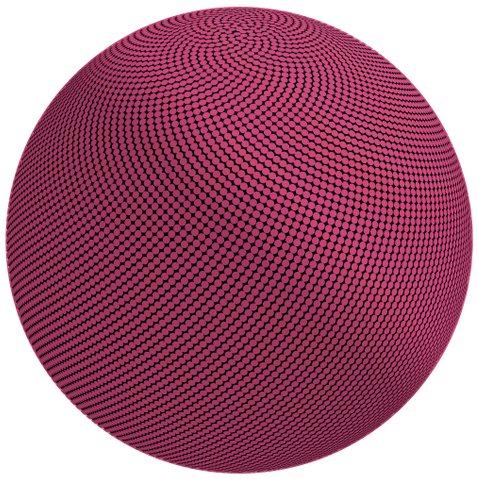
A base Detector class defines abstract methods one needs to fill to create a new detector.

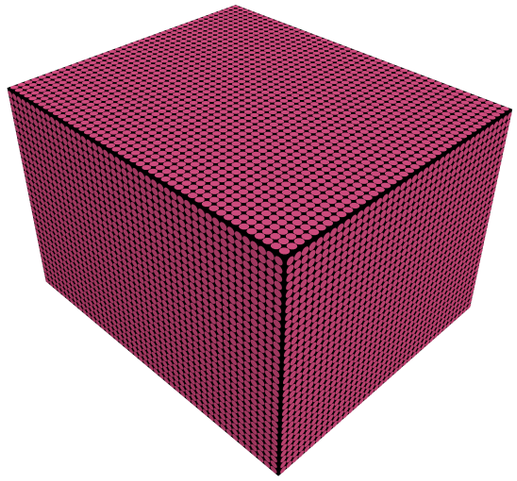
One propagator, shared across geometries, orchestrates the ray-tracing calculation. A detector geometry consists of an array of sensors in 3D space + boundary description + custom propagator primitives.

CURRENTLY SUPPORTS MOST COMMON DETECTOR GEOMETRIES: CYLINDERS, SPHERES AND BOXES

LUCiD







KEY ASPECTS

A base Detector class defines abstract methods one needs to fill to create a new detector.

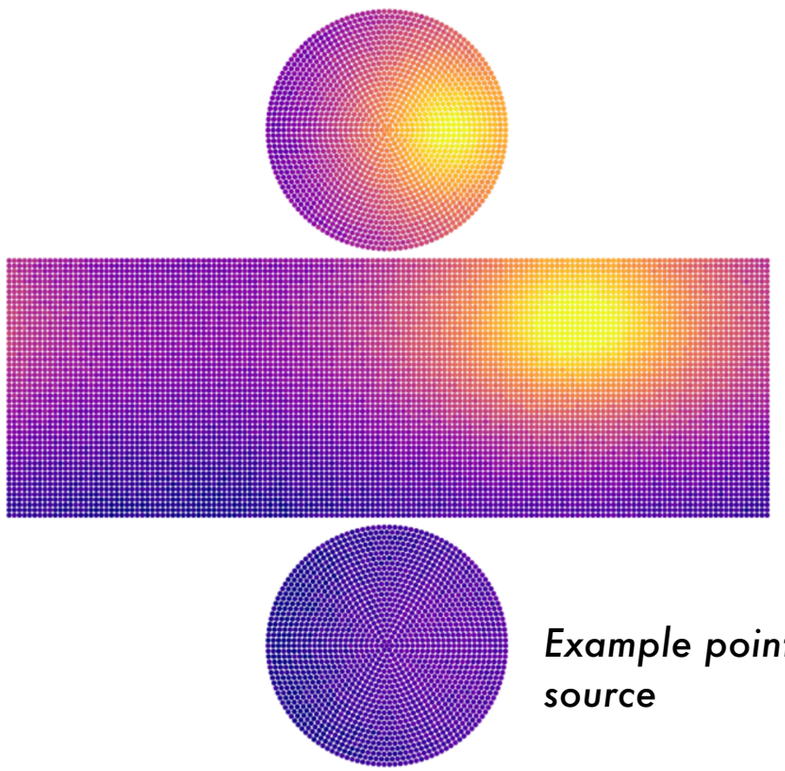
One propagator, shared across geometries, orchestrates the ray-tracing calculation. A detector geometry consists of an array of sensors in 3D space + boundary description + custom propagator primitives.

We have templates and examples, it is modular and easy to adapt to individual needs.

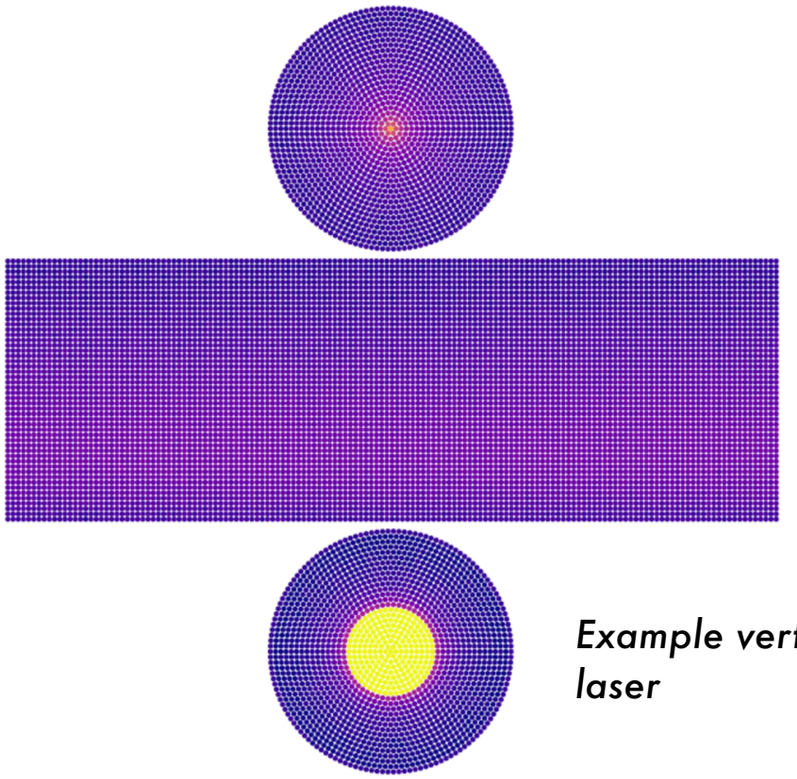
Talking to community is critical at this stage, we look for interested partners across ALL experiments.

The ray generation step consists in describing the list of the initial ray properties (photon initial positions, times, directions, etc).

Calibration sources are generally straightforward, e.g. a a point source has all rays with common position, and random isotropic direction. A laser has also common position and the direction distribution depends on assumptions.



Example point source

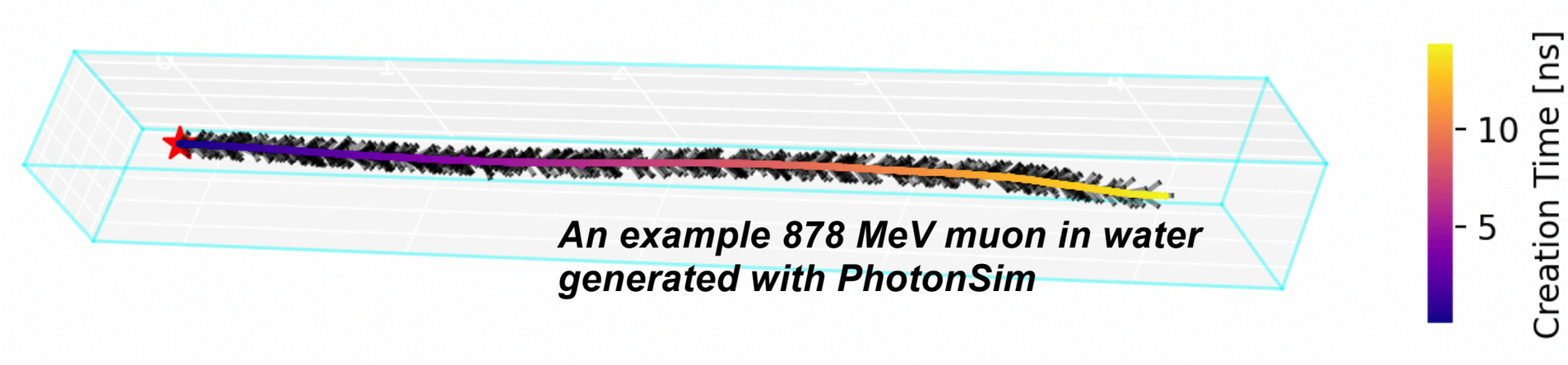


Example vertical laser

Modelling ray generation in tracks is more challenging: We need to model a distribution that incorporates all the microphysics, stochasticity, etc. → Goal was to mimic FitQun, SK reconstruction algorithm, so we parametrised the 'average' ray emission for a given track.

LET'S USE GEANT4

GEANT4 is not differentiable, but can be used to generate inputs to create a differentiable surrogate.
For this purpose, in the context of this project, I created PhotonSim.

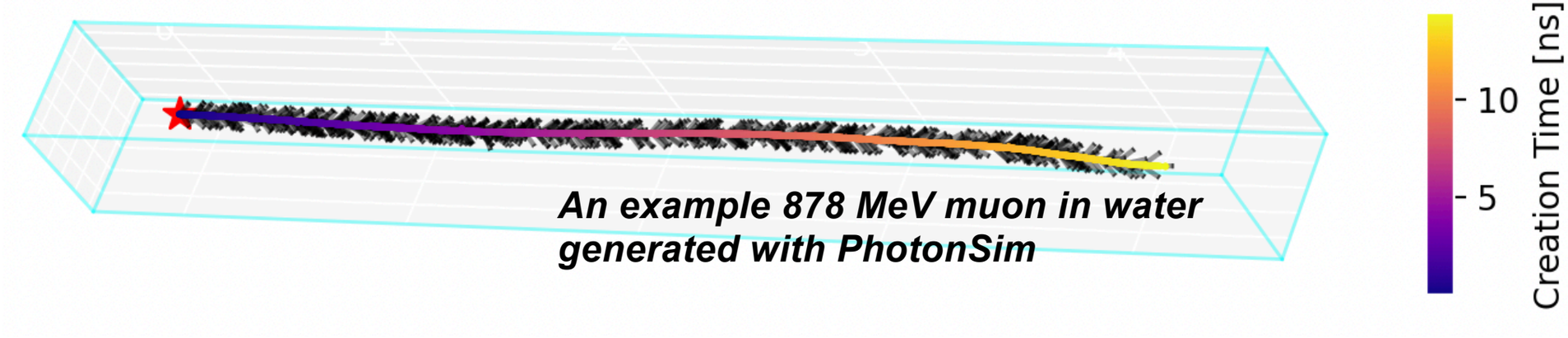


RAY GENERATION (TRACKS)

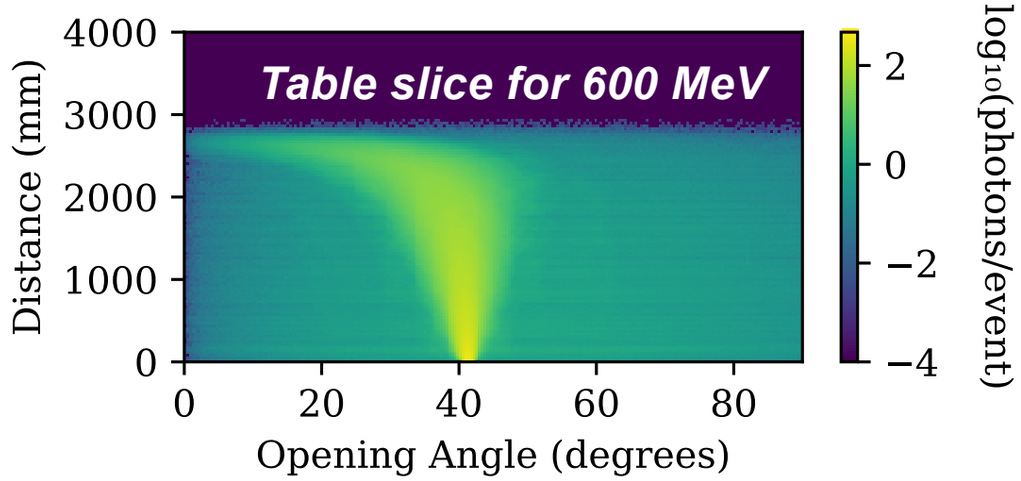
Modelling ray generation in tracks is more challenging: We need to model a distribution that incorporates all the microphysics, stochasticity, etc. → Goal was to mimic FitQun, SK reconstruction algorithm, so we parametrised the 'average' ray emission for a given track.

LET'S USE GEANT4

GEANT4 is not differentiable, but can be used to generate inputs to create a differentiable surrogate.
For this purpose, in the context of this project, I created PhotonSim.



We model the distribution of opening angles vs the distance to the track origin as a function of the track energy averaging 10k mono-energetic initial particles. We store the outcome in a 3D table:



We model the 3D look-up table using a neural network.
In particular we use SIREN (SInusoidal REpresentation Network). https://arxiv.org/pdf/2006.09661

HOW DOES IT WORK?

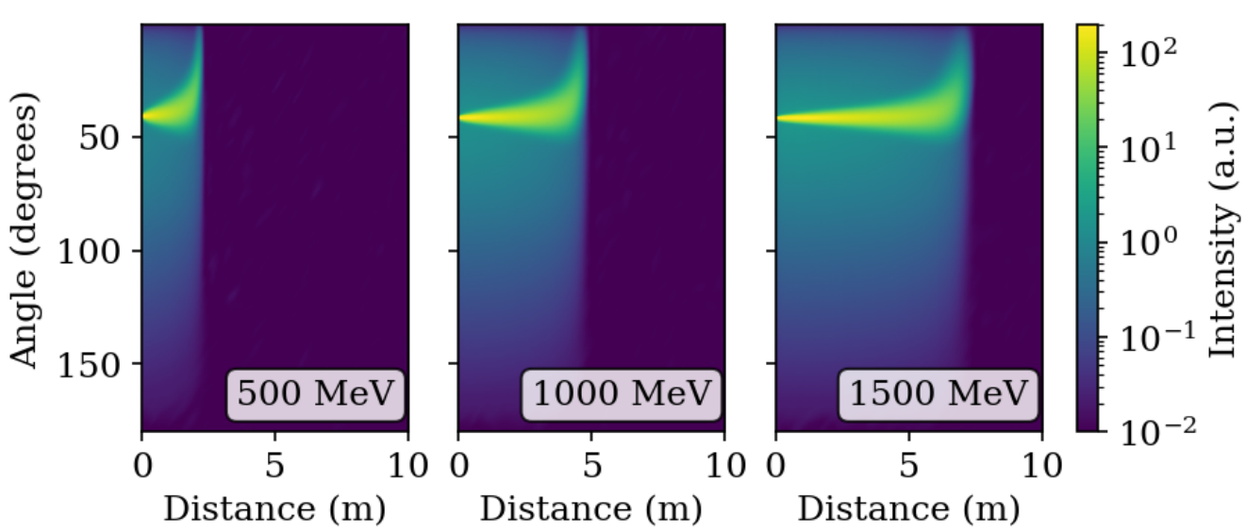
SIREN is essentially an MLP that uses periodic (sinusoidal) activation functions. SIRENs are great for representing continuous signals with fine detail and smooth derivatives, and are therefore well suited as implicit neural representations of images, audio and similar high-frequency targets. It is also a clean fit for tasks where the gradient of the network is also part of the pipeline.

We model the 3D look-up table using a neural network.
In particular we use SIREN (SInusoidal REpresentation Network). https://arxiv.org/pdf/2006.09661

HOW DOES IT WORK?

SIREN is essentially an MLP that uses periodic (sinusoidal) activation functions. SIRENs are great for representing continuous signals with fine detail and smooth derivatives, and are therefore well suited as implicit neural representations of images, audio and similar high-frequency targets. It is also a clean fit for tasks where the gradient of the network is also part of the pipeline.

Learned Intensity distributions with SIREN

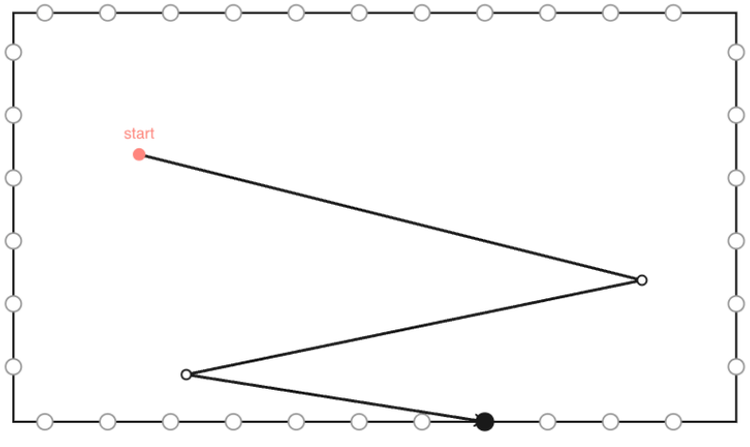


RAY PROPAGATION

TWO PROPAGATION MODES

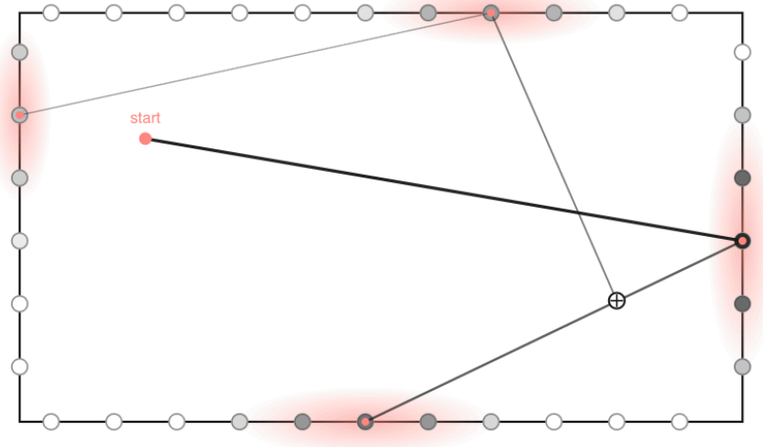
MC MODE

Discrete sampling



EXPECTED-VALUE MODE

Continuous & differentiable



Each iteration, a ray traverses some distance to its next interaction: either a scatter point in the medium or a wall/sensor surface, and then either scatters, reflects, or contributes to detection at a sensor. Absorption acts over the path length traversed in that iteration.

repeat K times

① Medium interactions

$P_{reach} = \exp(-d_{surf} / \lambda_s)$ $P_{det} = P_{reach} \cdot (1 - R)$ $A(d) = \exp(-d_{surf} / \lambda_a)$

implicit capture

② Deposit at sensors (always — ray is never killed)

$n_{dep} = I_i \cdot \alpha_i \cdot P_{det} \cdot A(d) \cdot w_{geo}$ *gaussian relaxation* $t_{hit} = t + d_{surf} / v_g$

straight through estimator

③ Path selection (scatter vs. reflect — detection excluded)

$P_{ref} = P_{reach} \cdot R$
$P_{scat} = 1 - P_{reach}$
$P = [\, P_{scat}, P_{ref} \,] / (P_{scat} + P_{ref})$

$h \sim \text{Categorical}(P)$
$w = h - sg(P) + P$
$x_{new} = w_0\, x_{scat} + w_1\, x_{surf}$

④ State update

$\alpha_i \leftarrow \alpha_i \cdot \exp(-d_{step} / \lambda_a)$ $t \leftarrow t + d_{step} / v_g$

Legend

λ_s scattering length
d_{surf} ray–surface distance
w_{geo} geometric overlap wei
h one-hot path sample

λ_a absorption length
I_i generation intensity
v_g group velocity in medium
P path probability vector

R surface reflectivity
α_i survival factor
d_{step} path length this step
$sg(\cdot)$ stop-gradient operator

HOW DO WE CREATE HITS FROM RAYS?

When working in 'data mode':

CANDIDATE RAYS

From the propagator, each candidate carries:

- a weight
- which sensor it hits
- arrival time

M = K × max_sensors × n_rays entries in total

QE DETECTION

Each candidate is kept with a per-sensor probability:

```
p = QE × QE_corr[sensor]
keep ~ Bernoulli(p)
```

Hard 0/1 decision — what a real sensor does.

AGGREGATE PER SENSOR

Group surviving rays by target sensor and reduce:

```
charge[sensor] = Σ weights
time[sensor] = min(times)
```

Total charge and first-arrival time per sensor.

DETECTOR SMEARING

Apply SK-like resolution to mimic real electronics:

- Gaussian jitter on times (TTS)
- Gain smearing on charges

Matches the resolution of the real detector.

HOW DO WE CREATE HITS FROM RAYS?

When working in 'data mode':

CANDIDATE RAYS

From the propagator, each candidate carries:

- a weight
- which sensor it hits
- arrival time

M = K × max_sensors × n_rays entries in total

→

QE DETECTION

Each candidate is kept with a per-sensor probability:

```
p = QE × QE_corr[sensor]
keep ~ Bernoulli(p)
```

Hard 0/1 decision — what a real sensor does.

→

AGGREGATE PER SENSOR

Group surviving rays by target sensor and reduce:

```
charge[sensor] = Σ weights
time[sensor] = min(times)
```

Total charge and first-arrival time per sensor.

→

DETECTOR SMEARING

Apply SK-like resolution to mimic real electronics:

- Gaussian jitter on times (TTS)
- Gain smearing on charges

Matches the resolution of the real detector.

When working in 'prediction mode':

CANDIDATE RAYS

From the propagator, each candidate carries:

- a weight
- which sensor it hits
- arrival time

M = K × max_sensors × n_rays entries in total

→

QE WEIGHTING

Same per-sensor probability as data mode — applied as a continuous weight:

```
p = QE × QE_corr[sensor]
weight ← weight × p
```

Continuous — keeps the pipeline differentiable.

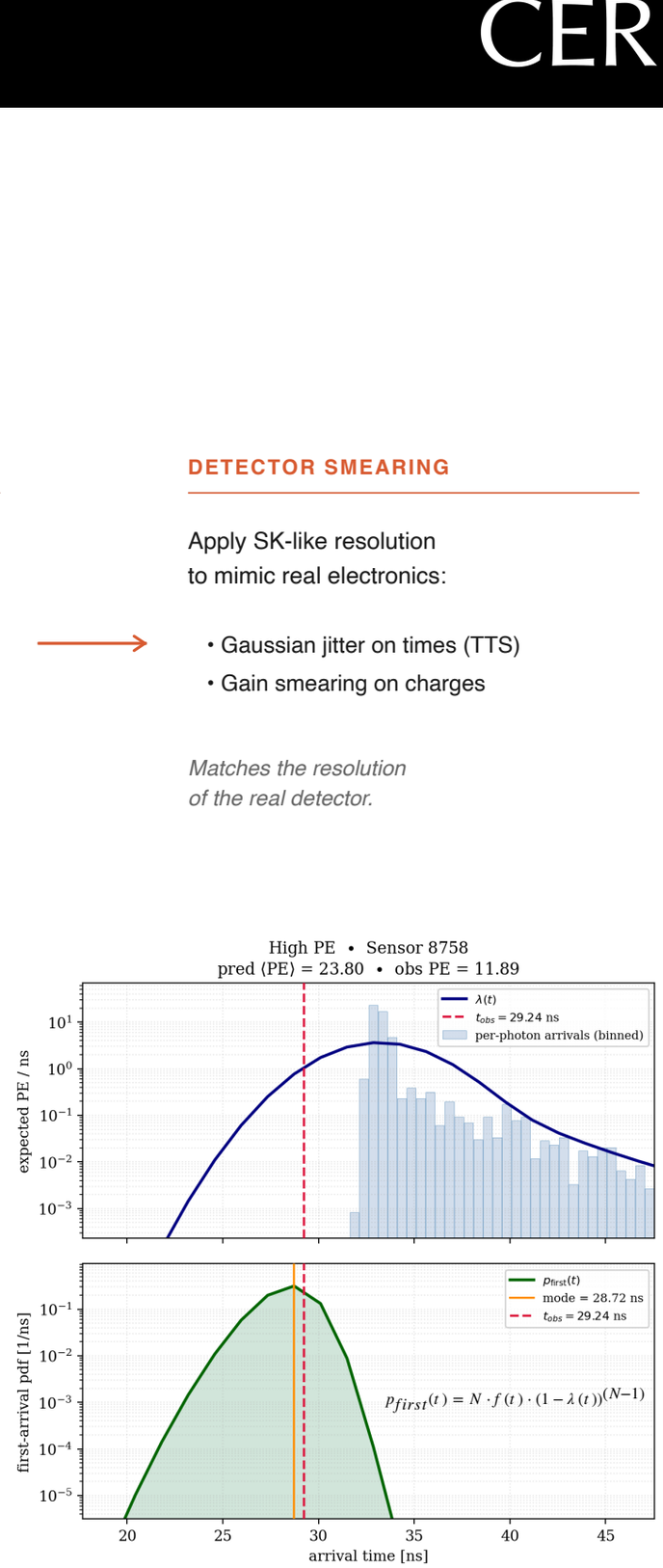
→

AGGREGATE PER SENSOR

Reduce only the charge — per-ray list stays intact:

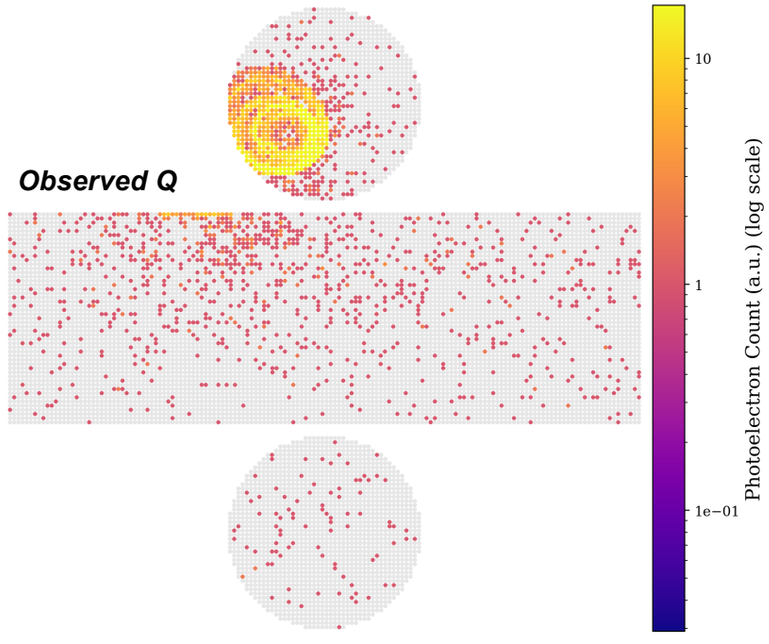
```
charge[sensor] = Σ weights
(times, sensor_id → kept)
```

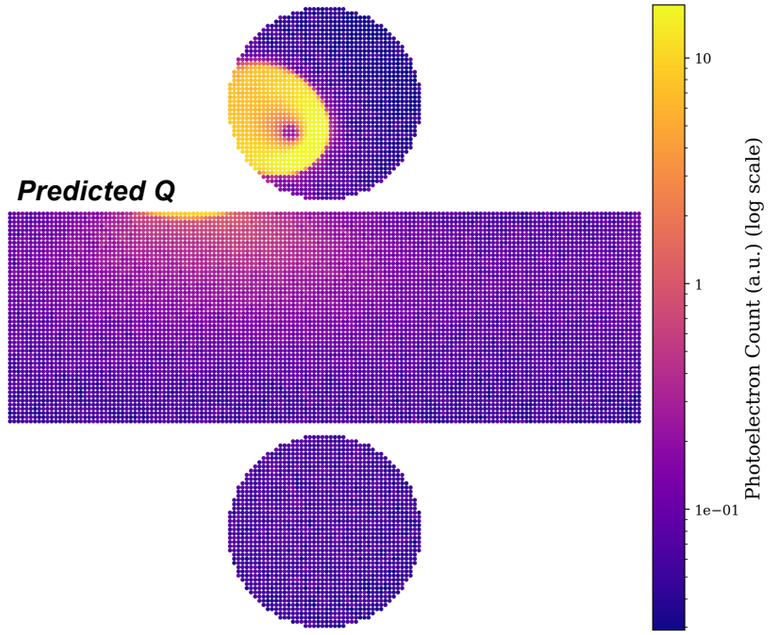
Likelihood consumes the full per-ray list directly.



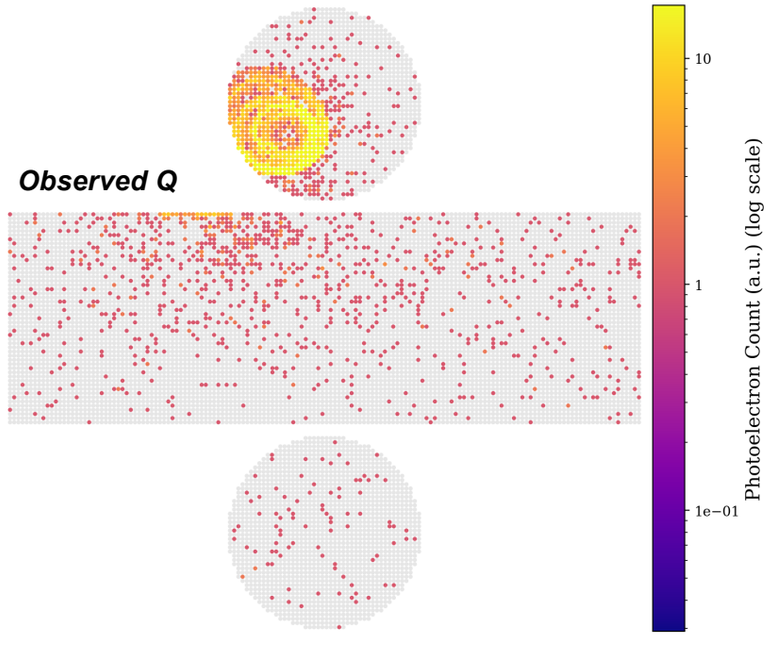
time likelihood example for one sensor

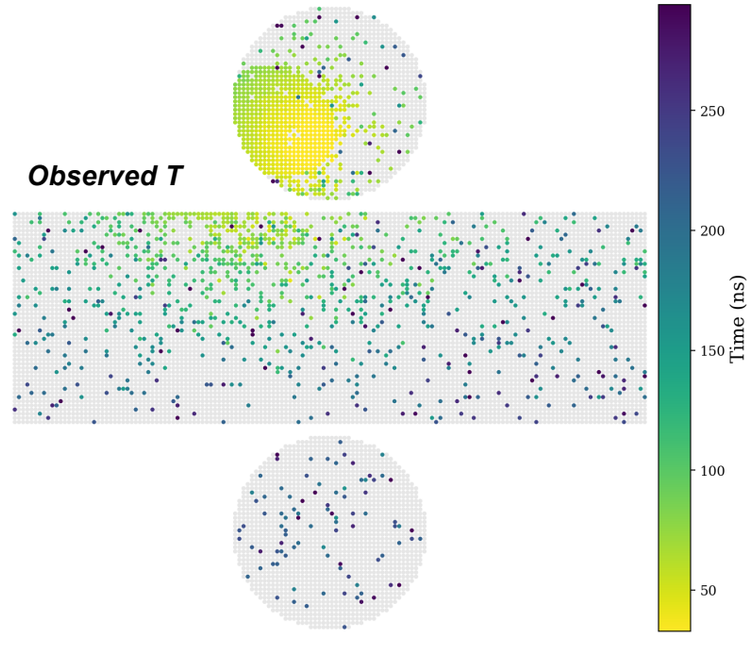
Summed charged weights correspond directly to charge expectation per sensor.
Full list of arrival times is used to calculate first time of arrival likelihood in the loss.

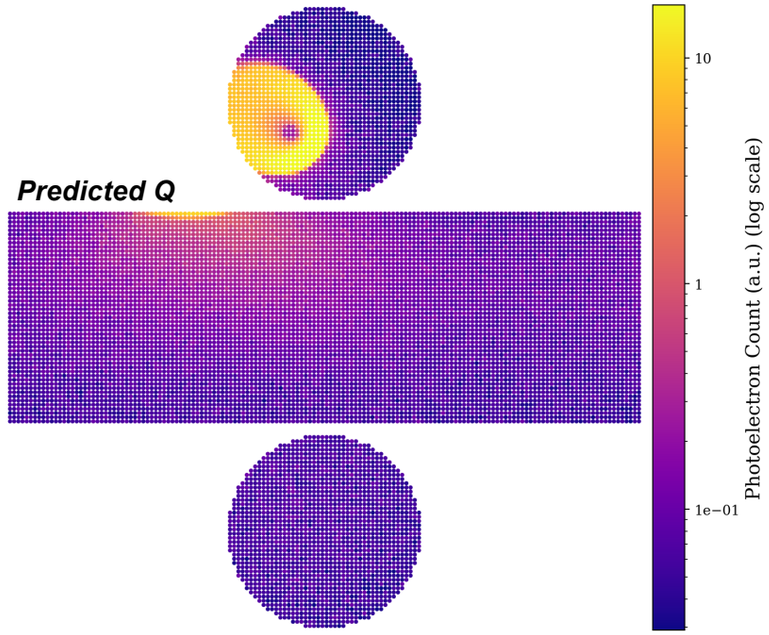


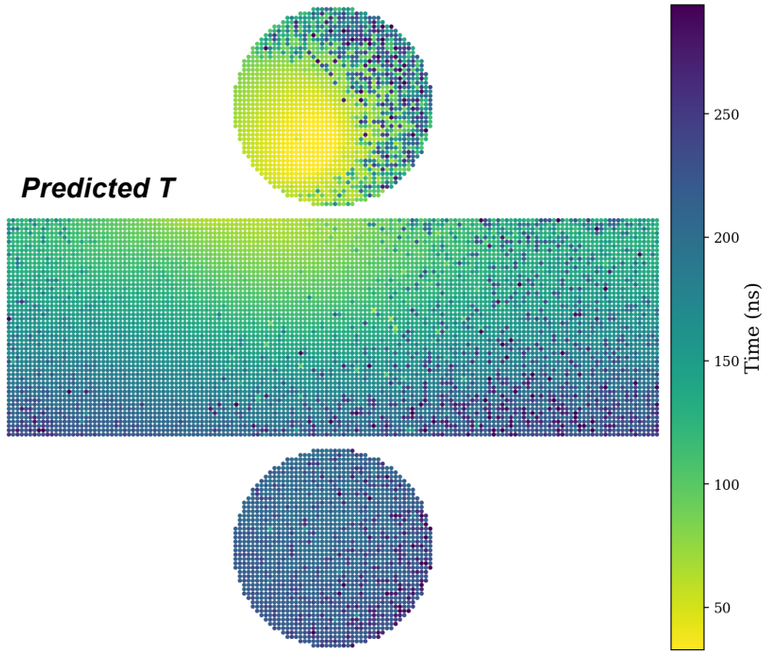


$$\mathscr{L}_{charge} = \sum_i \left[n_i^{pred} - n_i^{obs} \log n_i^{pred} + \log(n_i^{obs}!) \right]$$







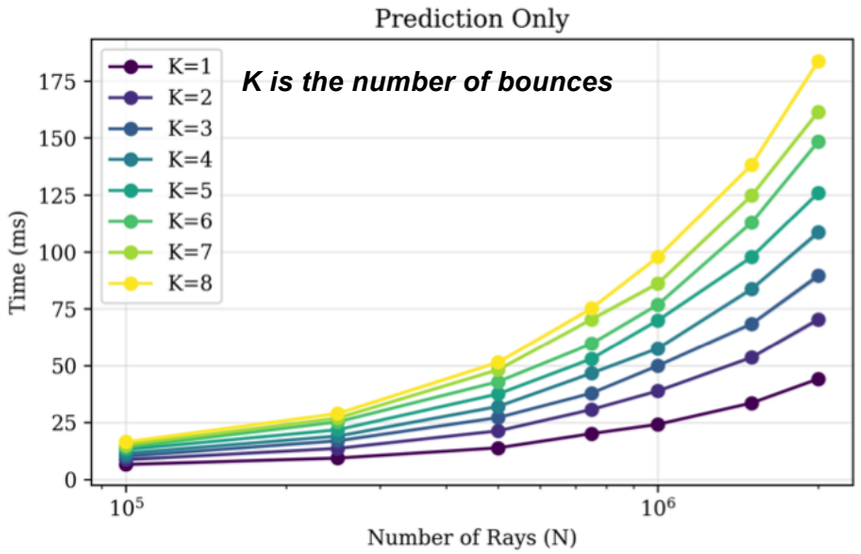


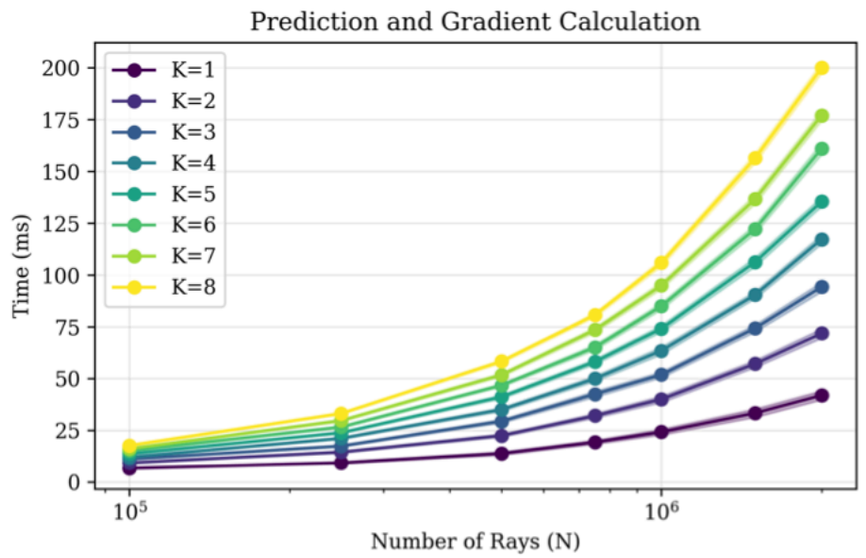
$$\mathscr{L}_{charge} = \sum_i \left[n_i^{pred} - n_i^{obs} \log n_i^{pred} + \log(n_i^{obs}!) \right]$$

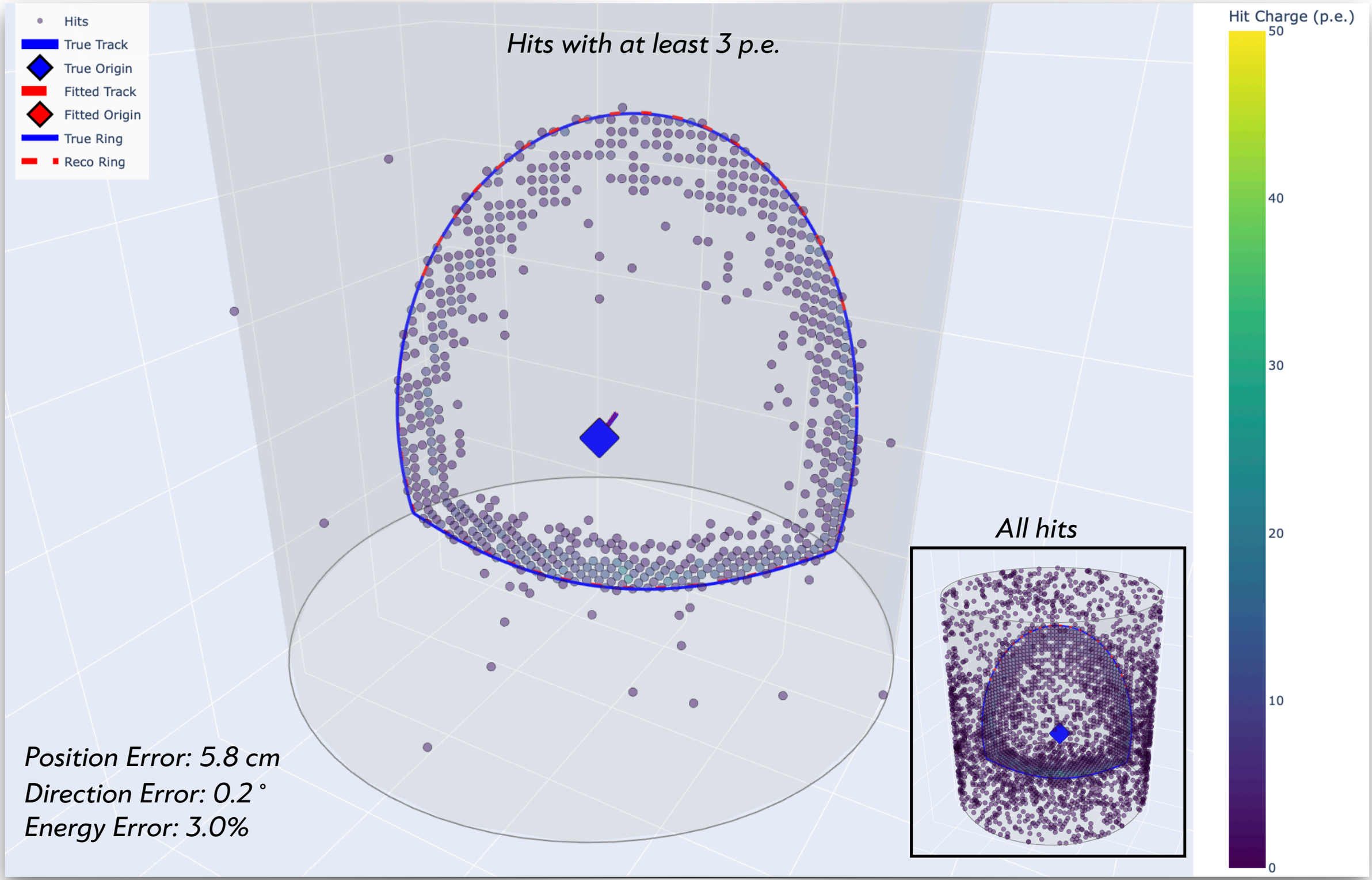
$$\mathscr{L}time = \sum i \in H - \log p_{first}^{(i)}(t_i^{obs}) \qquad H = i : n_i^{obs} > 0$$

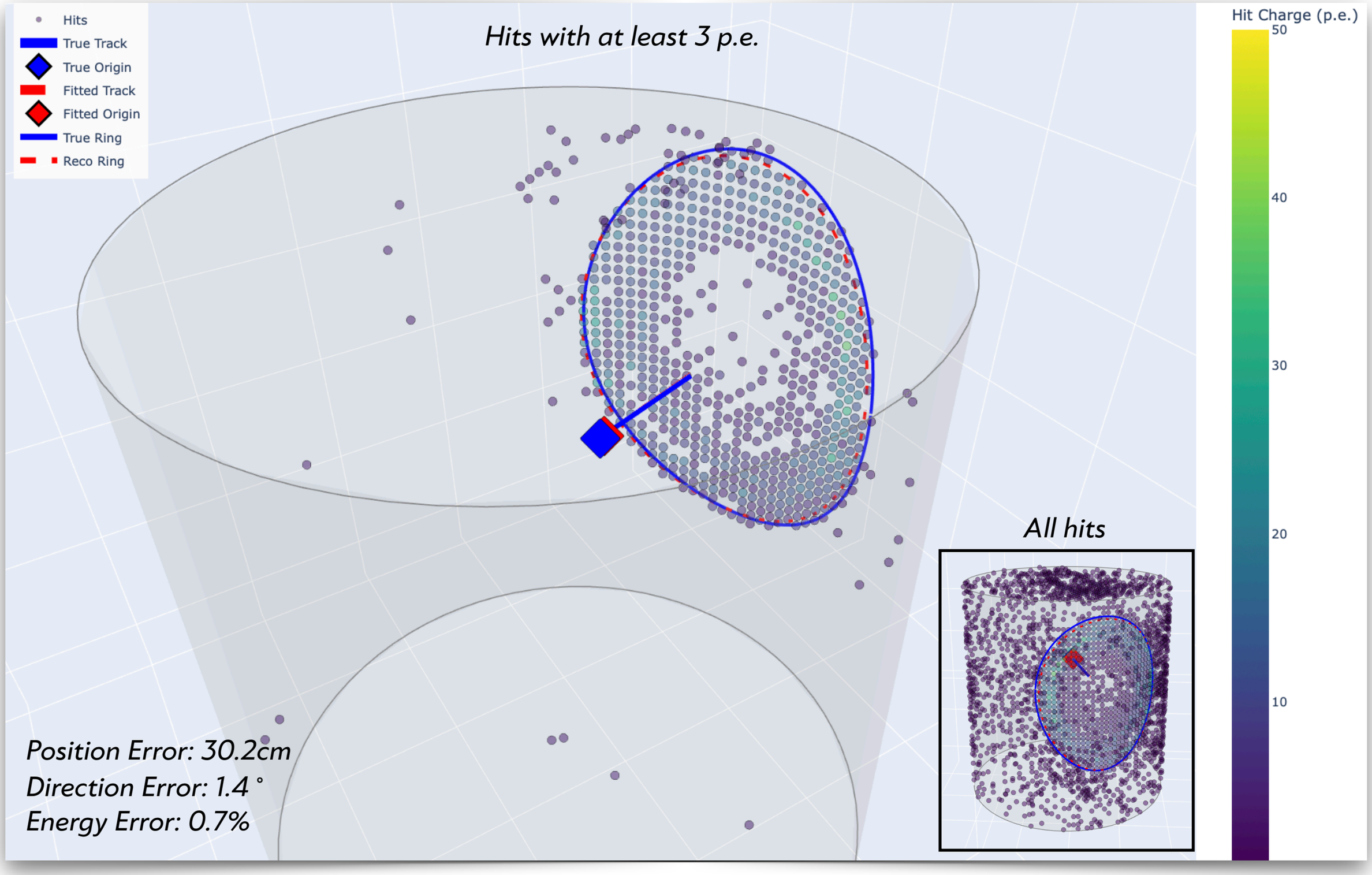
- The combination of JAX, JIT compiled kernels and modern GPUs allows generating accurate predictions in O(ms).
- Gradient calculation adds a small overhead.
- We play some tricks to minimise JIT overhead → Rays are padded in fixed-sized buckets, we ignore padded rays.

CALCULATIONS USING NVIDIA A100






Hits
True Track
True Origin
Fitted Track
Fitted Origin
True Ring
Reco Ring
Hits with at least 3 p.e.
Hit Charge (p.e.)
50
40
30
20
10
0
All hits
Position Error: 5.8 cm
Direction Error: 0.2 °
Energy Error: 3.0%


Hits
True Track
True Origin
Fitted Track
Fitted Origin
True Ring
Reco Ring
Hits with at least 3 p.e.
Hit Charge (p.e.)
50
40
30
20
10
All hits
Position Error: 30.2cm
Direction Error: 1.4 °
Energy Error: 0.7%