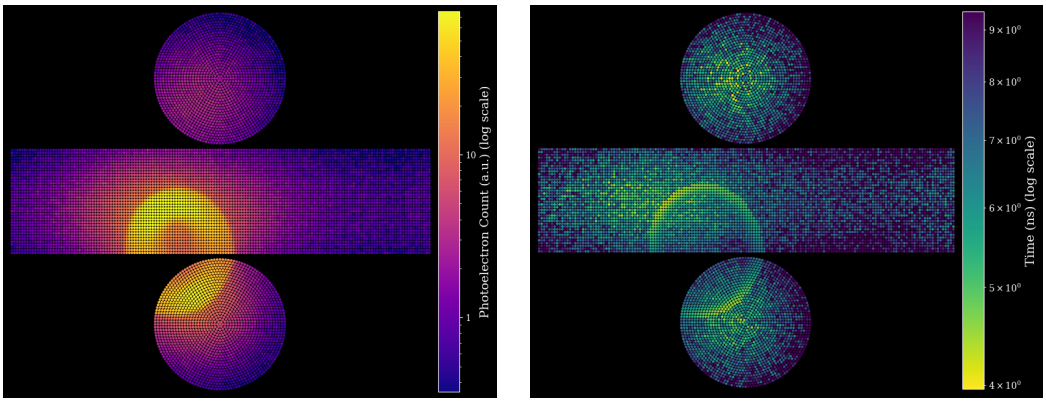
Differentiable Simulation for Calibration and Reconstruction of Optical Particle Detectors

Omar Alterkait
Tufts University, IAIFI
omar.alterkait@tufts.edu

César Jesús-Valls
CERN
Cesar.Jesus@cern.ch

Abstract

Optical particle detectors face complex calibration challenges as experiments scale, with high-dimensional parameter spaces and strong correlations making traditional Monte Carlo sampling computationally prohibitive. We introduce LUCiD[1], a differentiable ray-tracing framework for optical particle detectors that computes expected detector responses by propagating probability weights rather than sampling discrete paths. Straight-through estimation handles stochastic path selection while Gaussian relaxations enable differentiable hit detection, implemented in JAX for GPU acceleration. The resulting gradients enable direct navigation of correlated parameter spaces where sampling methods struggle, unlocking new gradient-based approaches to calibration, reconstruction, and physics-ML integration.



(a) (b)

Figure 1: Simulated detection pattern for a 600 MeV muon in a cylindrical water Cherenkov detector. Cylindrical detector geometry unrolled for visualization. (a) Accumulated charge per sensor. (b) First-photon arrival times.

1 Introduction

As neutrino physics approaches its precision era, next-generation detectors face systematic uncertainties that will dominate their sensitivity. The 260-kiloton Hyper-Kamiokande detector [8], a water tank with thousands of photosensors, demands new calibration and reconstruction approaches. Better systematic control could achieve target sensitivities years earlier. This requires new computational approaches that can handle the scale and complexity of these detectors.

[1] Code: https://github.com/CIDeR-ML/LUCiD

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

Modern GPU-accelerated frameworks with automatic differentiation [2] capabilities offer new approaches to this challenge. We implement our framework using JAX [4], leveraging its automatic differentiation and GPU compilation to enable efficient gradient-based optimization of high-dimensional parameter spaces. Ray-tracing computations naturally parallelize on GPUs, making them ideal for photon transport simulation. However, particle detection involves discrete processes such as photon-sensor intersections and stochastic effects like scattering and absorption, which break standard automatic differentiation and require physics-aware modifications to maintain meaningful gradients. We present LUCiD, a differentiable ray-tracing framework that addresses these challenges. [2]

2 Differentiable Ray-Tracing Framework

The LUCiD simulation pipeline consists of three interconnected stages: photon generation from particle tracks, propagation through the detector medium, and registration at photomultiplier tube (PMT) surfaces, where these photosensors detect individual photons. Figure 1 demonstrates the complete simulation pipeline output, showing both the spatial charge distribution and temporal photon arrival patterns for a single muon event in our differentiable framework.

2.1 Photon Generation via Learned Models

Cherenkov emission (light produced when charged particles exceed the speed of light in water) and secondary particle interactions produce complex photon distributions that vary nonlinearly with energy. We train a SIREN [9] network $f_\theta : (E, \theta, s) \to w$ that maps particle energy E, emission angle θ relative to the particle trajectory, and distance s along the particle track to photon weights w. The network is trained on 10^4 GEANT4-simulated muon tracks across multiple energies. During inference, we employ adaptive sampling: first evaluating the SIREN on a coarse grid in (E, θ, s) space, then densely sampling N total points within grid cells that exceed a threshold weight.

2.2 Differentiable Ray Propagation

We model PMTs as spheres and employ standard ray-sphere intersection algorithms [7] for detection checks along with ray-cylinder intersection to handle boundary reflections.

Weight-Based Formulation: Each photon carries weight w_i representing its expected contribution. When reaching a surface at distance d, the outcome probabilities are:

$$p_{\text{detect}} = e^{-d/\lambda_s}(1 - R), \quad p_{\text{reflect}} = e^{-d/\lambda_s} R, \quad p_{\text{scatter}} = 1 - e^{-d/\lambda_s} \tag{1}$$

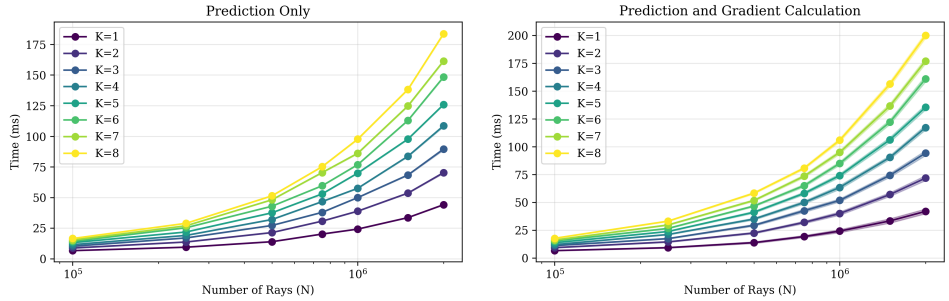
The photon contributes $w_i p_{\text{detect}} e^{-d_{\text{total}}/\lambda_a}$ to the PMT, where d_{total} is the total distance traveled across all bounce iterations. It then continues with weight $w_i(p_{\text{reflect}} + p_{\text{scatter}})e^{-d_{\text{total}}/\lambda_a}$. This process repeats for K bounces.

Path Selection: For continuing photons, we select between volumetric scattering and surface reflection using a Straight-Through Estimator [3]: a one-hot sample selects the physical path in the forward pass, while gradients flow through the underlying probabilities via a stop-gradient surrogate. The reflection direction is computed using the surface normal at the hit location via specular reflection law $\vec{d}_{\text{reflect}} = \vec{d} - 2(\vec{d} \cdot \vec{n})\vec{n}$. Scattering position and direction are obtained via reparameterization [6] of exponential and Rayleigh distributions respectively. Computing expected detector responses rather than sampling individual paths provides dramatic variance reduction, as exponential attenuation ensures >99.9% of weight is absorbed within $K = 8$ bounces, matching Monte Carlo statistics with far fewer computations.

Computational Efficiency: We subdivide the detector surface into spatial patches and precompute which sensors lie in each patch, storing up to 4 PMTs per patch. Since photons travel in straight lines between bounces, we only check sensors in the intersected patch. This reduces per-photon computations from $O(10^4)$ to $O(10)$ and makes the algorithm scale independently of total PMT count. Figure 2 demonstrates the resulting performance gains: processing five hundred thousand photons takes ~ 50 ms, and automatic differentiation adds only $\sim$10–15% overhead to the forward pass, independent of the number of optimized parameters.

[2] An extended version of this work with additional calibration and reconstruction studies is available at [1].

Figure 2: Computational performance of LUCiD on an A100 GPU as a function of the number of rays (N) and the number of propagation bounces (K).



2.3 Photon Detection via Gaussian Relaxation

The discrete PMT hit function produces biased gradients that accumulate errors. We model photons as 2D Gaussians perpendicular to their trajectory, computing overlap probability $P(d)$ with circular PMTs of radius r through a 2D integral over the Gaussian-circle intersection. Since only the photon-PMT distance d varies during simulation (with fixed $\sigma = 0.1r$), we precompute $P(d)$ and its gradient on a dense grid, then use linear interpolation for both the values and the pre-computed gradients during simulation. This provides smooth gradients that preserve the detector's physical response characteristics while enabling each photon to contribute to multiple nearby PMTs, substantially improving gradient quality compared to discrete hit assignments.

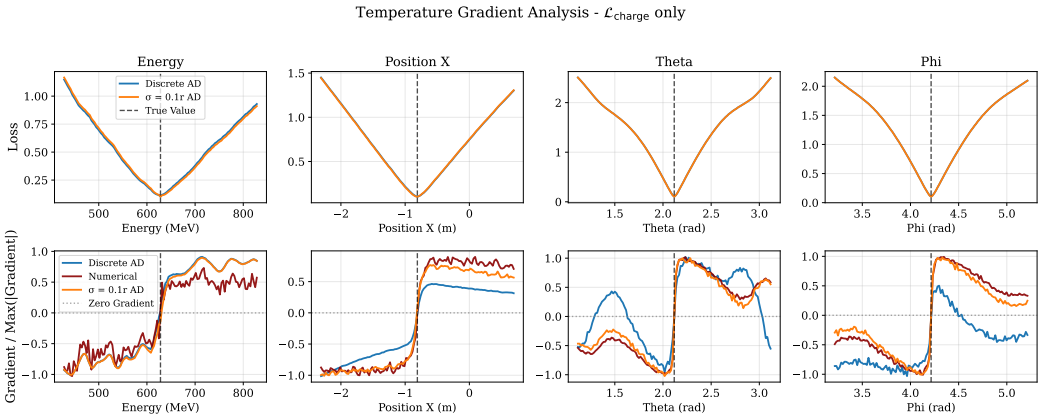


Figure 3: One-dimensional gradient analysis varying individual particle parameters. Top: loss landscapes. Bottom: normalized gradients from discrete AD (blue), Gaussian-relaxed AD with $\sigma = 0.1r$ (orange), and numerical finite differences (red).

PMTs record the arrival time of the leading edge after digitizing and integrating charge. For each PMT, we accumulate all photon contributions across all K bounces, then approximate the first-photon time via intensity-weighted softmin with temperature $\beta = 0.1$ ns, balancing first-photon approximation with gradient stability.

3 Gradient Analysis and Optimization Results

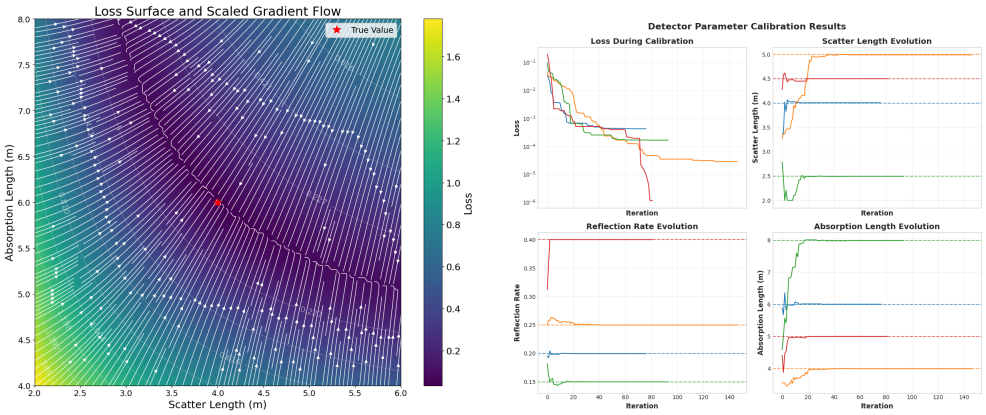
We evaluate our framework using a cylindrical detector with radius 5m, height 8m, and 9720 uniformly distributed PMTs (radius 0.1m). All experiments use 10^6 photons with $K = 8$ bounces. To enable differentiable spatial comparison between observed and predicted hit patterns, we apply Gaussian soft assignment weights $W_{ij} = \exp(-d_{ij}/\tau)/\sum_k \exp(-d_{ik}/\tau)$ with $\tau = 0.1$ m between nearby sensors. This prevents mode collapse, where the simulation could match observed light by concentrating predictions at a subset of sensors. The full loss combines charge pattern agreement,

timing residuals, and total light conservation, weighted via geometric mean to balance contributions without additional hyperparameters.

3.1 Gradient Quality and Reconstruction Performance

Figure 3 compares gradient quality between discrete and relaxed formulations. While loss curves are nearly identical between formulations, the gradients reveal crucial differences: the relaxed formulation closely matches numerical finite differences, whereas discrete gradients exhibit spurious artifacts, particularly in angular parameters. In high dimensions, finite difference computation becomes prohibitive, making our differentiable approach essential for practical optimization.

In closure tests, our framework achieves a vertex resolution of 15.6 cm, an angular resolution of 1.1°, and a momentum resolution of 1.5%. These values are comparable to those of FitQun (15.8 cm, 1.0°, 2.3%) [5], the standard Super-Kamiokande reconstruction algorithm, in a similar simulation sample.



(a) Gradient flow for calibration parameters (b) Multi-parameter optimization convergence

Figure 4: Detector calibration optimization using isotropic source. (a) Loss landscape and gradient flow for scattering versus absorption length. Red star indicates true values. (b) Simultaneous parameter optimization from different initial conditions (colors) converging to true values (dashed lines).

3.2 Calibration Parameter Optimization

Figure 4 demonstrates robust calibration using an isotropic source. The left panel varies two parameters: scattering length and absorption length, which have competing effects that are traditionally difficult to disentangle. Increased scattering reduces direct light, while absorption reduces total light. This competition creates the valley visible in the 2D plot. Gradient streamlines demonstrate robust convergence to true parameter values, even from distant initial conditions. The right panel confirms this robustness across multiple parameter sets and initial conditions, all converging within 150 iterations. This demonstrates the framework's ability to resolve parameter correlations that plague traditional grid-search methods.

4 Conclusion

Through differentiable physics simulation, we enable optimization techniques that were computationally infeasible with traditional Monte Carlo methods. Where Monte Carlo requires extensive sampling to probe parameter correlations, our gradients directly compute sensitivities, achieving comparable precision in orders of magnitude less time. Beyond these immediate gains, the differentiable framework opens new possibilities: gradient descent naturally scales to multi-particle events, avoiding the curse of dimensionality that limits traditional methods. Unknown processes can be parameterized by embedded networks, learning from data while maintaining interpretability. The framework generalizes across monolithic optical detector geometries and media. As experiments push toward ultimate precision, gradients become not just useful but necessary for navigating expanding parameter spaces.

References

[1] Omar Alterkait, César Jesús-Valls, Ryo Matsumoto, Patrick de Perio, and Kazuhiro Terao. End-to-end differentiable calibration and reconstruction for optical particle detectors, 2026.

[2] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey, 2018.

[3] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013.

[4] James Bradbury et al. JAX: composable transformations of Python+NumPy programs, 2018.

[5] M. Jiang et al. Atmospheric Neutrino Oscillation Analysis with Improved Event Reconstruction in Super-Kamiokande IV. *PTEP*, 2019(5):053F01, 2019.

[6] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[7] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation (3rd ed.)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, November 2016.

[8] Hyper-Kamiokande Proto-Collaboration et al. Hyper-kamiokande design report, 2018.

[9] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *arXiv*, 2020.