Differentiable Simulation of a Liquid Argon TPC for High-Dimensional Calibration

Pierre Granger
APC
Paris
granger@apc.in2p3.fr

Abstract

Liquid argon time projection chambers (LArTPCs) are widely used in particle detection for their tracking and calorimetric capabilities. The particle physics community actively builds and improves high-quality simulators for such detectors in order to develop physics analyses in a realistic setting. The fidelity of these simulators relative to real, measured data is limited by the modeling of the physical detectors used for data collection. This modeling can be improved by performing dedicated calibration measurements. Conventional approaches calibrate individual detector parameters or processes one at a time. However, the impact of detector processes is entangled, making this a poor description of the underlying physics. We introduce a differentiable simulator that enables a gradient-based optimization, allowing for the first time a simultaneous calibration of all detector parameters. We describe the procedure of making a differentiable simulator, highlighting the challenges of retaining the physics quality of the standard, non-differentiable version while providing meaningful gradient information. We further discuss the advantages and drawbacks of using our differentiable simulator for calibration. Finally, we discuss extensions to our approach, including applications of the differentiable simulator to physics analysis pipelines.

This short paper summarizes a work from many people described in more detail in arXiv:2309.04639.

1 Introduction

High-quality simulations of physics detectors are a fundamental piece of infrastructure across a diverse set of scientific disciplines. In high-energy particle physics measurements, detector simulation is particularly crucial to the analysis and reconstruction of an event. To avoid bias in physics results, dedicated calibrations are required to match simulation and real data. In particle physics, a conventional approach for identifying sources of data-simulation difference is to isolate different detector modeling processes using selected control samples. However, this approach does not capture the interplay among entangled detector processes [1–7]. A simultaneous correction of all detector processes and physics models would be an ideal new paradigm for detector calibration.

Gradient-based optimization provides a pathway towards improving calibration. Gradient-based methods are powerful and efficient in high dimensional optimization, supporting a simultaneous fitting of arbitrarily many parameters. Equipping existing detector simulations with gradient information enables the use of gradient-based optimization for calibration, while allowing for trivial application of this calibration directly in the simulation code. The exact calculation of these gradients can be done using a set of techniques called automatic differentiation [8]. In automatic differentiation, computer code is decomposed into a set of fundamental operations with known derivatives. Derivatives of the full program may then be calculated using these fundamental operations and the chain rule. A

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

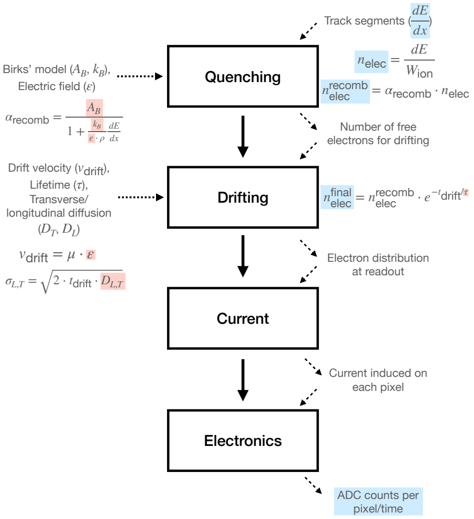

Track segments ($\frac{dE}{dx}$)
Birks' model (A_B, k_B),
Electric field (ε)
Quenching
$n_{\text{elec}} = \frac{dE}{W_{\text{ion}}}$
$n_{\text{elec}}^{\text{recomb}} = \alpha_{\text{recomb}} \cdot n_{\text{elec}}$
$\alpha_{\text{recomb}} = \frac{A_B}{1 + \frac{k_B}{\varepsilon \cdot \rho} \frac{dE}{dx}}$
Number of free electrons for drifting
Drift velocity (v_{drift}),
Lifetime (τ),
Transverse/
longitudinal diffusion
(D_T, D_L)
Drifting
$n_{\text{elec}}^{\text{final}} = n_{\text{elec}}^{\text{recomb}} \cdot e^{-t_{\text{drift}}/\tau}$
$v_{\text{drift}} = \mu \cdot \varepsilon$
$\sigma_{L,T} = \sqrt{2 \cdot t_{\text{drift}} \cdot D_{L,T}}$
Electron distribution at readout
Current
Current induced on each pixel
Electronics
ADC counts per pixel/time

Figure 1: Flow diagram of the simulator, highlighting inputs and outputs of each stage (blue) as well as commonly calibrated model parameters (red).

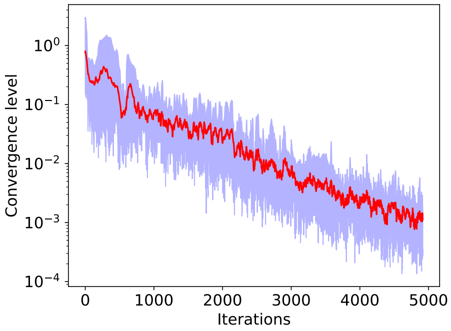


Figure 2: For each 6-parameter fit, the convergence level is defined as the maximum relative distance to the corresponding target value across all the parameters. The red line shows the average convergence level per iteration across the 10 fits. The band boundaries represent the minimum and maximum convergence level per iteration across all of the fits – all 10 convergence levels fall within the band.

variety of software packages, such as PyTorch [9], TensorFlow [10], and JAX [11], are capable of performing this calculation.

We demonstrate the utility of gradient-based optimization for a detector calibration task, using a liquid argon time projection chamber (LArTPC) detector as a case study. LArTPCs are used in a variety of modern particle physics experiments, such as the DUNE experiment, which is currently under construction [12]. Given the significant community and governmental investment in such large experiments, there is high interest and big efforts behind having a high-quality LArTPC simulation, and a significant potential impact in improving the corresponding calibration pipeline.

2 Simulator implementation

A LArTPC is a detector with an applied electric field across a volume of liquid argon. Deposition of energy can cause ionization in the liquid argon, and the ionization electrons drift under the influence of the electric field to be measured as current in an electronics readout. Our simulator uses a detector configuration corresponding to a module design for a DUNE LArTPC prototype.

The simulator developed for the DUNE LArTPC prototype is called `larnd-sim` [13]. Energy depositions are represented as line segments ("particle segments") defined by 3D start and end positions, and an associated energy. The simulator models the production of ionization electrons due to these energy depositions, as well as the drifting of electrons to the pixel readout at each anode plane. A current in the pixel readout can be induced by charges approaching or being directly collected on a given pixel. Each pixel is an independent readout channel with its own setup for trigger threshold and gain. Currents in the pixel readout are digitized to analog-to-digital converter (ADC) counts before being read out. The location of induced current in the pixel plane provides an x, y measurement. Timing information provides information on z. The magnitude of the induced current (ADC counts) gives information on deposited energy. A schematic view of this simulator is presented in Fig. 1.

We have rewritten `larnd-sim` to use differentiable, vectorized operations within PyTorch. To ensure the differentiability of all operations, we have introduced a set of *differentiable relaxations*: continuous, often smooth approximations of non-differentiable or poorly conditioned operations. Two scenarios of particular relevance for this work that require such relaxations are discrete integer operations and hard masking operations. All the truncations from floating point values to integers are removed, while all the masking operations are softened using a sigmoid function.

The simulated output using all of these relaxations was compared with a reference simulation from a snapshot of `larnd-sim`. Both simulators produced very similar results, with an average deviation of 0.04 ADC counts per activated pixel, which is two orders of magnitude below the typical noise level.

3 Parameter fitting

In the following, we apply the differentiable simulator described above to the calibration task. In our context, this means tuning the parameters of our simulator to match some given dataset, which can be either simulated or from a real experimental setup. Let $f(\chi, \theta)$ represent our differentiable simulator, with input particle segments χ and parameters θ. Our focus is to optimize the parameters θ.

For the optimization of θ, we propose an "analysis-by-synthesis" approach:

1. Choose initial values for the parameters, denoted as θ_0;
2. Run the forward simulation with these parameters, $f(\chi, \theta_0)$;
3. Compare the simulation output with target data F_{target}, using a *loss function*, $\mathcal{L}(f(\chi, \theta_0), F_{\text{target}})$;
4. Update parameter values $\theta_0 \to \theta_i$ to minimize the loss, and repeat from step 2 starting from parameters θ_i and forward simulation $f(\chi, \theta_i)$. The differentiable simulator enables a gradient-based update rule for θ_i.

To demonstrate the capability of our differentiable simulator in optimizing model parameters, we focus on a controlled case where F_{target} is generated using our simulator. Simulated targets F_{target} are constructed as $f(\chi, \theta_{\text{target}})$, where particle segments χ are known and may be used directly in the fit. Similarly, θ_{target} are known parameter values producing F_{target}. Successful fits recover fitted values $\theta = \theta_{\text{target}}$. This procedure is therefore known as a closure test.

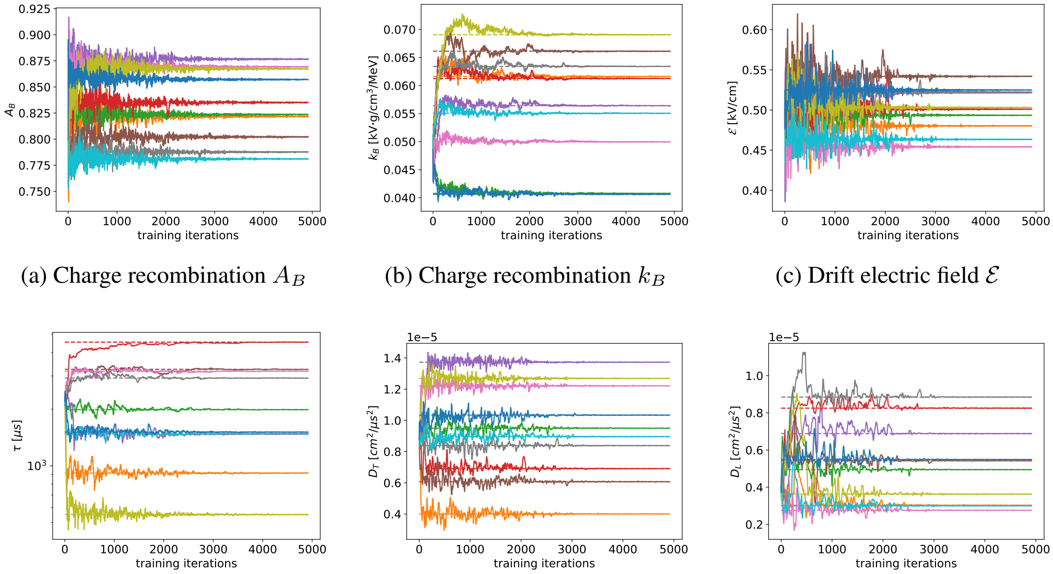
The output of the simulation is the charge q, in ADC counts, read out on relevant pixels of coordinates x and y at time t. Pixel coordinates and readout times are discrete. We use a dynamic time warping discrepancy (DTW) [14] as a loss function. It was designed for time series analysis and can easily deal with misaligned data series, which is crucial for our application, where we try to match sparse outputs. The inputs to the DTW loss function are sequences of ADC counts, hierarchically ordered by the corresponding values of x, y, and z. Dynamic time warping is not nicely differentiable by default. We therefore employ Soft-DTW [15], a smoothed version of DTW, using the implementation from Ref. [16, 17].

For this demonstration, we fit 6 parameters using our differentiable simulator: the Birks model parameters A_B, k_B; electric field $\mathcal{E}$; lifetime τ; and longitudinal and transverse diffusion coefficients D_L and D_T. To mimic a muon control sample that is typically used for LArTPC calibration, we simulate 100 events with about 10 muons per event. Muon injection angles are sampled randomly from an isotropic distribution while their energy is set to 1 GeV.

4 Results and outlook

To select parameter targets, we draw a uniform random value for each parameter from ranges determined by values measured in previous experiments. Each target then corresponds to a single point sampled uniformly from the considered 6D phase space. This setup mimics a realistic experimental procedure, as our best initial guess for each parameter is based on previous measurements.

In Fig. 3, we show closure-test results for fits to 10 different parameter targets using the muon sample described above. Each fit is labeled by a different color, and the convergence of the 6D simultaneous fit for each parameter is shown in a separate panel. The targets cover a wide range of phase space. All fits converge well to their corresponding targets. Fig. 2 shows a combined convergence metric across all 6 parameters, with the distribution calculated across the same 10 fits as in Fig. 3. For each fit, the convergence level is defined as the maximum over all parameters of the relative distance of each parameter to its corresponding target value. The band shows the maximum and the minimum relative distances to the targets among the 10 fits. All fits converge to within 1 % of their target values after 5000 iterations.



(a) Charge recombination A_B

(b) Charge recombination k_B

(c) Drift electric field $\mathcal{E}$

(d) Electron lifetime τ

(e) Transversal diffusion D_T

(f) Longitudinal diffusion D_L

Figure 3: Results from a simultaneous fit of six detector model parameters: A_B, k_B, $\mathcal{E}$, τ, D_T, D_L using the default muon sample. Each color indicates a sampled 6D target parameter point, and all fits start from the same initial parameter values, mimicking a realistic fitting scenario. The dashed lines label the target values of each parameter for each respective fit. The solid lines show the evolution of fitted parameter values with respect to the iteration number in the fit.

The results above demonstrate that a robust calibration can be performed in a multi-dimensional phase space using gradient-based optimization. This novel technique enables us to simultaneously optimize a set of model parameters across the detector simulation. In addition, the number of particles required for this calibration would have a negligible impact on data-taking and could be collected in a very short amount of time. This would allow for frequent verification of the calibration. Furthermore, this procedure is extensible to larger or different sets of parameters, making it suitable for generic use. The optimized model parameters can be immediately applied within the same simulation used for calibration. This benefits experiments by automatically ensuring consistent application across the simulation and analysis chain.

The presented calibration results are simulation closure tests. With real data, we will have to estimate the particle segments event by event. For track-like particle samples, such as muons, pions, and protons, this can be done by fitting the readout hits with lines and breaking those lines into segments. We will then need to reconstruct $\frac{dE}{dx}$ using the charge readout and the reconstructed segment length. However, this reconstruction has some dependence on detector parameters and will likely require iterative updates of $\frac{dE}{dx}$ alongside the parameter optimization fit.

The calibration tests presented here are also a test of the validity of parameter gradients through our simulator. This work therefore further sets the stage for the use of differentiable detector simulation within a broader machine learning context, allowing for e.g. explicit feedback of detector simulation on neural network training, a rich area with many applications, including learning to remove detector effects.

In summary, our work is a first step towards a broader differentiable physics program in particle physics. This differentiable physics program has the potential for a significant impact on the way physics analysis is performed, and there is a broad set of interesting future tasks towards integrating differentiable toolkits within particle physics to expand analysis capability and improve the quality and output of new physics results.

References

[1] C. Adams et al. Calibration of the charge and energy loss per unit length of the MicroBooNE liquid argon time projection chamber using muons and protons. *JINST*, 15(03):P03022, 2020. doi: 10.1088/1748-0221/15/03/P03022.

[2] B. Abi et al. First results on ProtoDUNE-SP liquid argon time projection chamber performance from a beam test at the CERN Neutrino Platform. *JINST*, 15(12):P12004, 2020. doi: 10.1088/1748-0221/15/12/P12004.

[3] P. Abratenko et al. Measurement of the longitudinal diffusion of ionization electrons in the MicroBooNE detector. *JINST*, 16(09):P09025, 2021. doi: 10.1088/1748-0221/16/09/P09025.

[4] R Acciarri, C Adams, J Asaadi, B Baller, T Bolton, C Bromberg, F Cavanna, E Church, D Edmunds, A Ereditato, S Farooq, B Fleming, H Greenlee, G Horton-Smith, C James, E Klein, K Lang, P Laurens, D McKee, R Mehdiyev, B Page, O Palamara, K Partyka, G Rameika, B Rebel, M Soderberg, J Spitz, A M Szelc, M Weber, M Wojcik, T Yang, and G P Zeller. A study of electron recombination using highly ionizing particles in the argoneut liquid argon tpc. *Journal of Instrumentation*, 8(08):P08005, aug 2013. doi: 10.1088/1748-0221/8/08/P08005. URL `https://dx.doi.org/10.1088/1748-0221/8/08/P08005`.

[5] S. Amoruso et al. Study of electron recombination in liquid argon with the ICARUS TPC. *Nucl. Instrum. Meth. A*, 523:275–286, 2004. doi: 10.1016/j.nima.2003.11.423.

[6] C. Adams et al. A method to determine the electric field of liquid argon time projection chambers using a UV laser system and its application in MicroBooNE. *JINST*, 15(07):P07010, 2020. doi: 10.1088/1748-0221/15/07/P07010.

[7] P. Abratenko et al. Measurement of space charge effects in the MicroBooNE LArTPC using cosmic muons. *JINST*, 15(12):P12037, 2020. doi: 10.1088/1748-0221/15/12/P12037.

[8] Andreas Griewank and Andrea Walther. *Evaluating Derivatives*. Society for Industrial and Applied Mathematics, second edition, 2008. doi: 10.1137/1.9780898717761. URL `https://epubs.siam.org/doi/abs/10.1137/1.9780898717761`.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[10] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[11] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

[12] Babak Abi et al. Deep Underground Neutrino Experiment (DUNE), Far Detector Technical Design Report, Volume II: DUNE Physics. 2 2020.

[13] DUNE Collaboration. Highly-parallelized simulation of a pixelated LArTPC on a GPU, 2022. URL `https://arxiv.org/abs/2212.09807`.

[14] Hiroaki Sakoe and Seibi Chiba. A dynamic programming approach to continuous speech recognition. In *Proceedings of the Seventh International Congress on Acoustics, Budapest*, volume 3, pages 65–69, Budapest, 1971. Akadémiai Kiadó.

[15] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series, 2018.

[16] Mehran Maghoumi. *Deep Recurrent Networks for Gesture Recognition and Synthesis*. PhD thesis, University of Central Florida Orlando, Florida, 2020.

[17] Mehran Maghoumi, Eugene Matthew Taranta, and Joseph LaViola. Deepnag: Deep non-adversarial gesture generation. In *26th International Conference on Intelligent User Interfaces*, pages 213–223, 2021.