Reviews in Physics 10 (2023) 100085

Contents lists available at ScienceDirect

Reviews in Physics

journal homepage: www.elsevier.com/locate/revip

Toward the end-to-end optimization of particle physics instruments with differentiable programming

Tommaso Dorigo [a,b,x,*], Andrea Giammanco [a,c,x], Pietro Vischia [a,z,c], Max Aehle [d], Mateusz Bawaj [e], Alexey Boldyrev [a,f], Pablo de Castro Manzano [a,b], Denis Derkach [a,f], Julien Donini [a,g,x], Auralee Edelen [h], Federica Fanzago [a,b], Nicolas R. Gauger [d], Christian Glaser [a,i], Atılım G. Baydin [a,j], Lukas Heinrich [a,k], Ralf Keidel [l], Jan Kieseler [a,m], Claudius Krause [a,n], Maxime Lagrange [a,c], Max Lamparth [a,k], Lukas Layer [a,b,o], Gernot Maier [p], Federico Nardi [a,b,q,g], Helge E.S. Pettersen [r], Alberto Ramos [s], Fedor Ratnikov [a,f], Dieter Röhrich [t], Roberto Ruiz de Austri [s], Pablo Martínez Ruiz del Árbol [a,u], Oleg Savchenko [b,c], Nathan Simpson [v], Giles C. Strong [a,b], Angela Taliercio [c], Mia Tosi [a,b,q], Andrey Ustyuzhanin [a,y], Haitham Zaraket [a,w]

[a] *MODE Collaboration*[1]
[b] *Istituto Nazionale di Fisica Nucleare, Sezione di Padova, Italy*
[c] *Centre for Cosmology, Particle Physics and Phenomenology (CP3), Université catholique de Louvain, Belgium*
[d] *Chair for Scientific Computing, Technische Universität Kaiserslautern, Germany*
[e] *Università di Perugia and INFN, Sezione di Perugia, Italy*
[f] *HSE University, Russia*
[g] *Université Clermont Auvergne, Laboratoire de Physique de Clermont, CNRS/IN2P3, France*
[h] *SLAC National Accelerator Laboratory, USA*
[i] *Department of Physics and Astronomy, Uppsala University, Sweden*
[j] *Department of Computer Science, University of Oxford, UK*
[k] *Physik-Department, Technische Universität München, Germany*
[l] *Center for Technology and Transfer, University of Applied Sciences Worms, Germany*
[m] *CERN, Switzerland*
[n] *NHETC, Department of Physics and Astronomy, Rutgers University, USA*
[o] *Università di Napoli "Federico II", Italy*
[p] *Deutsches Elektronen-Synchrotron (DESY), Germany*
[q] *Università degli Studi di Padova, Italy*
[r] *Department of Oncology and Medical Physics, Haukeland University Hospital, Norway*
[s] *Instituto de Física Corpuscular, UV-CSIC, Spain*
[t] *Department of Physics and Technology, University of Bergen, Norway*
[u] *Instituto de Física de Cantabria, UC-CSIC, Spain*
[v] *Lund University, Sweden*
[w] *Multi-Disciplinary Physics Laboratory, Optics and Fiber Optics Group, Faculty of Sciences, Lebanese University, Lebanon*
[x] *Universal Scientific and Education Research Network (USERN), the World*
[y] *Constructor University Bremen gGmbH, Campus Ring 1, Bremen, 28759, Germany*
[z] *Universidad de Oviedo and ICTEA, Spain*

ARTICLE INFO

Keywords:
Particle detectors

ABSTRACT

The full optimization of the design and operation of instruments whose functioning relies on the interaction of radiation with matter is a super-human task, due to the large dimensionality of the

* Corresponding author at: Istituto Nazionale di Fisica Nucleare, Sezione di Padova, Italy.
E-mail address: dorigo@pd.infn.it (T. Dorigo).

[1] https://mode-collaboration.github.io/

https://doi.org/10.1016/j.revip.2023.100085
Received 25 November 2022; Received in revised form 5 May 2023; Accepted 11 May 2023
Available online 25 May 2023

Differentiable programming
Machine learning
Optimization
Particle physics
Nuclear physics
Astrophysics

space of possible choices for geometry, detection technology, materials, data-acquisition, and information-extraction techniques, and the interdependence of the related parameters. On the other hand, massive potential gains in performance over standard, "experience-driven" layouts are in principle within our reach if an objective function fully aligned with the final goals of the instrument is maximized through a systematic search of the configuration space. The stochastic nature of the involved quantum processes make the modeling of these systems an intractable problem from a classical statistics point of view, yet the construction of a fully differentiable pipeline and the use of deep learning techniques may allow the simultaneous optimization of all design parameters.

In this white paper, we lay down our plans for the design of a modular and versatile modeling tool for the end-to-end optimization of complex instruments for particle physics experiments as well as industrial and medical applications that share the detection of radiation as their basic ingredient. We consider a selected set of use cases to highlight the specific needs of different applications.

1. Introduction

The modern investigation of the fundamental structure of Nature relies on the design and operation of complex instruments whose functioning is based on the interaction of radiation with matter. To extract the maximum scientific value from these experiments, as well as from potential other scientific, medical, or industrial applications, a full optimization of both design and operation of these instruments must be performed. However, the sheer complexity of this class of instruments makes their optimization a super-human task, because of the large dimensionality of the space of possible choices for geometry, detection technology, materials, data-acquisition, and information-extraction techniques, and the interdependence of the related parameters. On the other hand, massive potential gains in performance over standard, "human-experience driven" layouts are in principle within our reach if an objective function fully aligned with the final goals of the instrument is maximized through a systematic search of the configuration space, subject to constraints of various nature. In mathematics and engineering such concept is often referred to as *topology optimization* [1]. In physics, the alignment to the final goals of the instrument is often expressed using the frequentist statistics concept, rooted in hypothesis testing, of significance as the probability, expressed in units of variance of a Gaussian distribution, of observing a value of a test statistic (a function of the data) as large or larger than the observed one, assuming that the reference model is true [2]. The approach to experiment optimization traditionally used in the last half of a century is sequential, where each portion of instrument (often called subdetector or submodule) is optimized separately, and the global set of optimal parameters is defined as the union of the individual optimization results. This is, broadly, the way the state-of-the-art instruments currently active have been designed, especially in the field of particle physics (for instance, for CERN experiments such as the Compact Muon Solenoid [3]). We propose a paradigm shift, where the scientific objective of the experiment is encoded in the aforementioned objective function, and all the design parameters of the instrument are optimized at the same time. This reflects a fundamental mathematical property of multi-parameter optimization: the solution to the joint optimization of a set of parameters $\{\theta_i\}$ yields a set of optimal values $\{\theta_i^{\text{joint}}\}$ which is in general different, for each indexed parameter θ_i, from the set $\{\theta_i^{\text{sequential}}\}$ obtained by solving the one-dimensional optimization problem separately for each θ_i.

Although typically quite complex, similar problems may sometimes still be tractable by standard means, in the sense that a parameterized model of the system allows the definition of a likelihood function $L = p(x|\theta)$, given simulated data x, and a solution by minimization of $-\ln L$ with respect to the modeling parameters θ.

Another important characteristic of the kind of optimization proposed above stems from the stochastic nature of the interaction of radiation with matter. Such interaction is rooted in quantum mechanical processes, that is probabilistic processes where there is no deterministic law connecting the space of model parameters and the space of data. In this case, the optimization problem is intractable: the probability $p(x|\theta)$ of observing data x given underlying parameters θ may not be written explicitly. One has therefore access at best to the generating function of the observed data through forward simulation, a setting commonly referred to as likelihood-free or simulation-based inference [4].

The optimal choice of layout, characteristics, materials, and information-extraction procedures of a measuring instrument constitutes a loosely constrained problem, featuring a very large number of free parameters related by non-trivial correlations.

Over the course of the past eighty years, the intractability of the design optimization problems commonly encountered in particle physics has not prevented physicists from successfully conceiving, commissioning, and operating detectors of huge complexity. The development of increasingly performant instruments followed a robust strategy that, while systematically leveraging technological advancements in electronics and material science, duly exploited well-tested paradigms proven to work by previously acquired experience. For example, a long-standing paradigm for the detection of particles in collider physics experiments is the need to measure the momentum of all electrically charged particles by magnetic bending in gaseous or light materials before exploiting the electromagnetic and hadronic showers produced by both charged and neutral particles in dense matter. Another paradigm is the requirement of significant redundancy in the detection systems, to enable cross-calibration of the different components and offer robustness of the resulting inference. A further typical default is the choice of a symmetric layout of the detection components, such as the equal spacing of scintillating and passive elements along the depth of a calorimeter. While these paradigms have a strong motivation if we look at the past of particle detection practice, they are not necessarily valid for the optimization of very

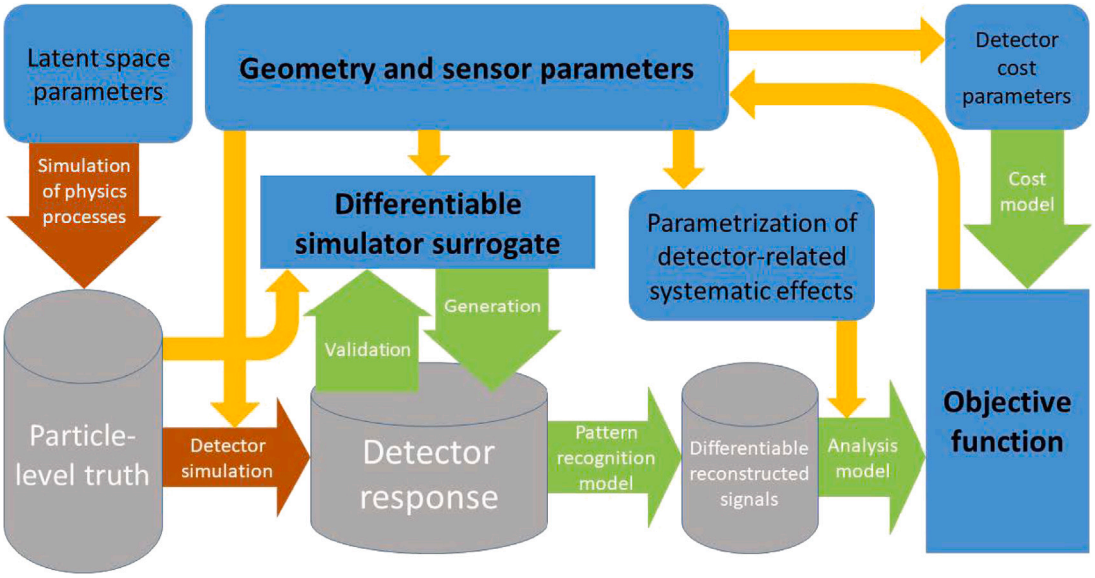


Fig. 1. Block diagram for the optimization of a generic particle physics detector. Data from a simulator (left, cans labeled *Particle-level truth* and *Detector response*) are used to train and validate a differentiable model (*Differentiable simulator surrogate*) of the relevant physical processes. Models of pattern recognition, inference extraction, cost of components, and a loss function may then become a function of detector geometry and construction layout parameters. A back-propagation loop of loss derivatives through the functional elements of the system allows their optimization.
Source: The figure is adapted from Ref. [5].

complex future instruments, where the sheer complexity of the task induces sever non-convexities in the loss function, resulting in a non-trivial structure of the set of optimal and near-optimal parameters. Paradigms that may still be valid will be recovered automatically by this new class of optimization techniques.

Computer science has undergone a fast progress in the past twenty years. The development and rapid expansion of deep neural network and gradient-optimization based software, denoted as *differentiable programming*, has been made possible by the concept of automatic differentiation. Such algorithms are based on nested differentiation based on the chain rule. Deep networks and complex backpropagation problems require several nested applications of the chain rule: symbolic differentiation methods become unfeasible because the resulting expression becomes too expensive to compute, whereas numerical differentiation incurs in rounding and truncation errors due to reaching machine precision in the representation of floating numbers. Automatic differentiation is a technique to obtain fast and exact numerical derivatives with no overhead: its astounding performance powers up virtually any modern deep learning or, broadly, gradient-based algorithm. We describe automatic differentiation, as well as the other approaches to computer derivatives, in deeper detail in Section 2.

The fast progress of computer science in the past twenty years, together with the development of deep neural networks and optimization software based on differentiable programming, offers us an unprecedented opportunity to rethink the foundations of our design strategies, and to identify and investigate novel, possibly revolutionary solutions we have been unable to figure out by ourselves. The typical design problems we face involve the choice of hundreds, if not thousands of parameters defining the placement and geometry of materials and detection layers, their specifications and performance, and their monetary cost. The full exploration of this high-dimensional space of design solutions is an entirely super-human task: to move forward, we must turn to the differentiable programming tools that make this exploration possible.

It must be noted that the breadth of the space of design solutions has also been increasing with our technological advancements. Nowadays, we can 3D print scintillation detectors [6], as well as design more complex detection elements with thin layers of AC-coupled resistive silicon sensors [7]. These advancements can best be exploited if we endow ourselves with the capability of performing continuous scans of the geometry space of the devices we wish to construct: this is something we achieve by developing differentiable programming pipelines.

Another reason for revisiting our detector design paradigms while accounting for the availability and development of new computer science tools is the evolution of the pattern recognition and inference procedures we have been adopting in the extraction of information from raw detector readouts. The demands posed to our instruments are continuously increasing, as we move, *e.g.,* toward the high-luminosity (HL) phase of the Large Hadron Collider (LHC) [8], or toward larger and larger detection volumes in cosmic ray and neutrino physics. The LHC and its experiments have been conceived in the nineties of last century and are designed to operate until the late thirties of the current century, passing through several phases of improved operations. For instance, at the HL-LHC, in a few years the ATLAS and CMS experiments will be reconstructing high-energy particle collisions within $\mathcal{O}(200)$ pileup interactions taking place during the same bunch crossing; the performance of standard reconstruction algorithms for charged tracks will be strongly reduced in the presence of an exponential increase of the combinatorial background. If deep learning methods will

be employed for those pattern recognition tasks (such as those described in Refs. [9–29]), the question arises of whether the detectors have been conceived to be optimal for those tools. Such a potential misalignment between design and exploitation is even more evident if we look further into the future, to the construction of colliders, such as the proposed Future Circular Collider (FCC) [30], characterized by higher center-of-mass energies than the current machines: since we are currently sitting on a rapidly growing curve of performance of artificial-intelligence (AI-)powered methods [31], in order for our future detectors to be most effective we need to consider their design as an optimization problem that includes a model of the pattern recognition and inference extraction procedures available at operation time, however hard it may be to envision their power today.

The above considerations motivate us to pursue a wide-ranging plan of investigations that has the primary purpose of educating ourselves and our community on how to best integrate all the elements of a detector design problem—from the modeling of the stochastic quantum phenomena to the description of detector layout, geometry, and performance; from the pattern recognition to the inference extraction procedures; and from the interplay of geometry and systematic uncertainties to the physical and economic constraints—into a single optimization problem, as schematized in Fig. 1. We believe that the capability to compute derivatives of the objective function with respect to any one of the parameters of the system, provided by implementing the whole pipeline using differentiable programming, will be key to enable the successful exploration of the large space of design choices, and the discovery of innovative solutions.

What we are facing is an extremely tall order if we consider a detector of the scale of collider experiments such as ATLAS or CMS. In fact, it is doubtful that we have today the resources, expertise, and skills required to attack a problem of that complexity. We must therefore proceed in steps, by considering less ambitious and more achievable goals. In this document we propose, and discuss in some detail, a series of design optimization tasks that are interesting in their own right, and whose solution via the above plan may enable us to build a framework of methods and software tools that together may constitute the building blocks for solving harder problems. While the specificity of a detector leaves little room for reuse of the differentiable surrogate models of particle interaction with active and passive components that may have been developed to study them, there is instead significant device independence in recently developed reconstruction algorithms empowered by deep learning [32], and a clear possibility of reusing the models developed for the monetary cost of the components, for the interaction between geometry- and detector-related systematic uncertainties, and for inference extraction.

At the core of any optimization procedure lays a carefully defined objective function, which should encode as closely as possible the explicit goals of the instrument we are designing. For a large scientific endeavor, specifying this function may appear an impossible task, given the multi-purpose nature of the detectors, the breadth of physics studies they enable, and the arbitrariness of the relative value of different scientific objectives of the experiment. However, we argue that the exercise of appraising those goals and proposing an evaluation metric *can* be beneficially carried out, and an objective function—or a family of objective functions that address different points of view (multi-objective optimization)—can profitably be specified. Indeed, such an exercise is not altogether different from the one of defining a trigger menu for a collider physics experiment, which produces a list of triggers with relative selection strategies, bandwidths, and prescaling factors. We stress that the resulting optimization study cannot be expected to produce a final answer, but rather that it may indicate advantageous combinations of design choices and "sweet spots" in the space of design parameters, guiding our hand toward robust and effective decisions. This corresponds to an "assisted optimization" paradigma, where the seasoned expert, which usually is in charge of optimizing the design of an instrument, used a landscape of near-optimal solutions provided by our optimization algorithm, while ultimately using their domain knowledge to take the final decision on which design works best, incorporating also considerations on aspects that are not (or not easily) parameterizable.

The present white paper, which builds on the ideas succinctly described in Ref. [5], is structured as follows. In Section 2 we provide an overview of the state of the art of the computer science ingredients that can be used to construct the software pipeline for an end-to-end optimization study, and we provide a brief survey of today's solutions to optimization problems in other fields of research. In Section 3 we outline the concrete way of defining a detector optimization problem and the way of assembling a set of modules to construct a closed-loop pipeline employing differentiable programming techniques. In Section 4 we provide a discussion of example applications and the specific needs of each, and assess their feasibility and requirements. In Section 5 we discuss the hardware and software requirements for the solution of the typical problems we have considered. We offer some conclusions in Section 6.

2. The state of the art in design optimization and differentiable programming

Scientists and engineers have leveraged the steady growth of available computing power over decades to continuously improve the accuracy of their numerical simulations. Nowadays, in many technical disciplines and for setups too complicated for traditional theoretical approaches, simulations can answer the question "what happens when we do X?" The oftentimes excellent agreement that simulations may achieve with physical experiments, at a tiny fraction of the cost of the latter, makes it natural to pose the next question, "which design X gives the *optimal* outcome?"

The "model pupil" among the technical disciplines that deal with this kind of question is certainly computational fluid dynamics (CFD). While first attempts to numerically improve the shape of airfoils [33,34] were based on finite differences, the analytic derivation of sensitivity equations [35–37] proves much more efficient when the number of design parameters is large. During the last two decades, automatic differentiation[2] (AD) has been integrated into various CFD codes [38–40] and enabled a large number of design optimization studies [40–43].

[2] Also called algorithmic differentiation, or simply autodiff.

"Optimality" can relate to much more than the technical performance of a design. In a wider sense, parameter and model fitting optimize the predictive quality of a theoretical model given empirical data. AD has enabled parameter fitting studies in fields as diverse as ice sheet modeling [44], optimal control [45], and quantitative finance [46]. Deep learning [47,48] can be understood as a special case of parameter fitting with an emphasis on representation learning and specialized model architectures for AI-related tasks such as computer vision [49], natural language processing [50], and reinforcement learning [51]. AD capability in machine learning (ML) frameworks such as PyTorch [52] and TensorFlow [53] is at the core of many recent advances in the machine learning community [54].

The relatively new term *differentiable programming*[3] is used to capture the essence of deep learning practice, where one constructs differentiable computer code—via AD—in order to solve various tasks, and optimizes them via gradient-based optimization of an objective based on training data. The term emphasizes the general-purpose programming aspect of recent machine learning approaches, in contrast with conventional neural networks. In this context, neural networks—viewed as compositions of a series of non-linear transformations—are only a member in the more general family of differentiable programs, which covers the space of all differentiable algorithms with, *e.g.,* control flow, loops, and recursion, and crucially includes differentiable numerical simulators in science and engineering disciplines.

In particle physics, measuring instruments rely on the extraction of inference from the data they collect, and thus base their operation on the aforementioned functionalities: therefore, their design optimization is closely connected to the precision of the models and the effectiveness of those parameter estimate procedures. However, instruments that rely on the interaction of radiation with matter in their data-collection mechanisms, such as particle detectors, add more complexity to the task, in that they introduce an element of intrinsic stochasticity from the quantum nature of the involved physical processes, and thus necessitate the deployment of special solutions, such as those described in Section 3. Hence, the full optimization of particle detectors and accelerators is a frontier topic examined by only a few examples in the literature; see *e.g.* Refs. [5,58–65].

There are two main ways to make a simulation differentiable. The first one consists of using AD directly in the simulation code, making use of the AD tools based on operator overloading or source code transformation in the programming language in which the simulator is implemented. As we will detail below in Sections 2.6 and 2.7, the complexity of such a task varies widely from case to case, sometimes making it not viable. The second way to have a differentiable simulation consists of using deep learning techniques to produce a differentiable surrogate model of the simulator, using supervised training data sampled by running the original simulator. We detail the advantages of such an approach in Section 2.8.

The remainder of this section is organized as follows. Section 2.1 introduces the notation for a general optimization problem. The optimization algorithms listed in Section 2.2 rely on the gradient of the objective function. Section 2.3 discusses multiple ways to compute derivatives, and Sections 2.4 and 2.5 illustrate the mathematical background of the forward and reverse mode of AD, respectively. Section 2.6 summarizes the general implementation aspects, while Section 2.7 lists various considerations that might make program-specific adaptations necessary. Finally, Section 2.8 introduces the notion of surrogate models.

2.1. Optimization

The most generic formulation of a mathematical optimization problem is

$$\min_{x \in \mathcal{X}} f(x) \tag{1}$$

where

- $\mathcal{X}$ is a space of possible choices, usually a subset of some $\mathbb{R}^d$, and
- $f : \mathcal{X} \to \mathbb{R}$ is an *objective*, *cost*, or *loss* function, quantifying the idea of an optimal choice.

The terms *loss* and *cost* usually refer strictly to a function defined on a data point; *e.g.,* measuring a distance between a prediction and a target value. All throughout this document, however, we will refer at this objective function as *loss*: we will use the term *cost* to refer to the cost of the instrument (monetary or of other nature). *Objective* is the most general term for any function we would like to optimize, and can cover cases where there is no explicit notion of a loss between a prediction and a target; it can also include regularization terms and other constraints. Note that these terms are sometimes used interchangeably by some authors. Both $\mathcal{X}$ and f are problem-specific and need to be determined by domain experts. Regarding the end-to-end optimization of detectors, $\mathcal{X}$ could represent the space in which each dimension corresponds to a numerical parameter affecting the detector hardware and software design.

The function f encodes the accuracy and technical and financial constraints. This involves a whole pipeline of software modules that simulate the physical processes inside the detector, generate the detector response, post-process it, and finally assess the objective. We describe these physics-specific aspects in Section 3 and present concrete examples of detector optimization projects in Section 4. This section deals with the theory for solving a general minimization problem as defined in Eq. (1).

2.2. Gradient-based optimization

Most of the design variables x_i of the detector systems of our interest (*e.g.* those considered *infra*, Section 4, as well as others of similar scope) can be chosen continuously and influence the objective function f in a smooth way, making them subject to

[3] A term initially proposed by Christopher Olah [55], David Dalrymple [56], and Yann LeCun [57] from a deep learning point of view.

```python
import torch
import scipy.optimize

def f(x): # returns a tuple (value, gradient)
  x_ad = torch.tensor(x, requires_grad=True)
  f_ad = x_ad[1]*x_ad[1]+torch.sin(x_ad[0])
  f_ad.backward()
  return ( f_ad.item(), [x_ad.grad[0], x_ad.grad[1]] )

x0 = (2,-3)
res = scipy.optimize.minimize(f, x0, method='BFGS', jac=True)
```

(a) Using SciPy [66] for optimization. See Fig. 2e for how the gradient is obtained.

```python
import math

def func(x, y):
  return y*y + math.sin(x)

x = 0.
y = 5.
z = func(x,y)
```

(b) *Primal*, *i.e.* undifferentiated, program.

```python
import ad_tool

def func(x, y):
  return y*y + ad_tool.sin(x)

x = ad_tool.DualNumber(0.,1.)
y = ad_tool.DualNumber(5.)
z = func(x,y)
```

(c) Program differentiated in *forward mode* AD using the *ad-hoc* tool in Fig. 2d.

```python
import math

class DualNumber:
  def __init__(self, primal, tangent=0):
    self.primal = primal
    self.tangent = tangent

  def __add__(self, other):
    return DualNumber( \
        self.primal+other.primal, \
        self.tangent + other.tangent)
  def __mul__(self, other):
    return DualNumber( \
      self.primal*other.primal, \
      self.primal*other.tangent + \
        self.tangent*other.primal)
  # ... define further operations

def sin(val):
  return DualNumber( \
      math.sin(val.primal), \
      math.cos(val.primal)*val.tangent)
# ... define further operations
```

(d) *Ad-hoc* tool for forward AD.

```python
import torch

def func(x, y):
  return y*y + torch.sin(x)

x = torch.tensor(0.,
                 requires_grad=True)
y = torch.tensor(5.,
                 requires_grad=True)
z = func(x, y)
z.backward()

print("z    =", z.item())
print("dz/dx=", x.grad)
print("dz/dy=", y.grad)
# alternative:
# torch.autograd.grad(z, [x,y])
```

(e) Program differentiated in *reverse mode* AD with PyTorch [52].

Fig. 2. Applying AD and optimization to a simple function $f(x) = y^2 + \sin(x)$. See the text for more detail.

differentiable programming and gradient-based optimization. In case a design variable is intrinsically discrete, such as the number of active layers in the muon tomography apparatus described in Section 4.3, alternative strategies must be used, typically consisting either in a smart parameterization of the discrete variable in terms of a vanishing continuous efficiency, or in the optimization of that parameter based on unsupervised techniques such as reinforcement learning [51].

Gradient descent algorithms have a general form where a differentiable function f is minimized over a number of iterations by starting from an initial parameter $x^{(0)}$ and generating a sequence of updated parameters $x^{(1)}$, $x^{(2)}$, etc. by using an update rule

$$x^{(k+1)} = x^{(k)} - \eta_k \cdot \nabla_x f(x^{(k)}) \,. \tag{2}$$

A small step into the direction of the negative gradient decreases the value of f, as $-\nabla_x f(x^{(k)})$ points into the direction of steepest descent of f around $x^{(k)}$. The step size $\eta_k > 0$ has to be chosen in advance, or adjusted in each step, to make sure that the gradient descent algorithm does not step "beyond" the minimum.

If f is twice differentiable, the Hessian matrix $H_x f$ can be used in the (damped) Newton's method:

$$x^{(k+1)} = x^{(k)} - \eta_k \cdot H_x f(x^{(k)})^{-1} \cdot \nabla_x f(x^{(k)}) \,. \tag{3}$$

Quasi-Newton methods such as BFGS [66–68] and L-BFGS-B [69,70] only require the knowledge of $\nabla_x f$ and form a internal approximation of the Hessian.

Stopping criteria indicate that the present solution $x^{(k)}$ is close to a *local minimum*, *i.e.* a point that minimizes f within some neighborhood. For example, the norm of the gradient $\nabla_x f(x^{(k)})$ can be tested to fall below a user-defined threshold. In this case, no further improvement is expected and the algorithm terminates.

A general function f can have many local minima. The one selected by an optimization algorithm not necessarily is the *global minimum* as well. A finite set of iterates $x^{(1)}, \ldots, x^{(k)}$ cannot comprehensively explore the full design space $\mathcal{X}$. Better local minima might therefore remain undiscovered, no matter how ingeniously the optimization algorithm uses the information on the value and derivatives of f at the iterates. For applications, solutions that do not perfectly or provably reach the global minimum may still present a valuable improvement over previous designs. Ref. [71] reviews the state of the art in the mathematical understanding of the loss function landscape for hyperparameterized systems.

We must remark that the optimization of complex detector design typically involves a nonconvex (with respect to the design parameters) loss function landscape, akin to that of deep learning applications. In this case, stochastic gradient descent (SGD) [72] is not guaranteed to find the global minimum. Empirically, though, SGD seems to find correctly the global minimum even for nonconvex loss functions. A review of recent literature in mathematics concerning convergence properties of SGD in nonconvex landscapes can be found in Ref. [73].

Many popular optimization algorithms have been implemented in open-source packages such as SciPy [74]; Fig. 2(a) shows an example calling a SciPy optimizer with the gradient of the objective supplied by the PyTorch's AD module. Essentially, the user must provide an initial solution and code to evaluate both f and its derivatives automatically. Three ways to obtain the derivative are listed in the next subsection.

2.3. Computing derivatives: An overview

Numerical differentiation can approximate the components of a gradient $\frac{\partial f}{\partial x_i}(x^{(k)})$ by evaluating f at several points around $x^{(k)}$. The most commonly used formulas are the forward and central finite difference quotients,

$$\frac{f(x^{(k)} + h \cdot \mathrm{e}^{(i)}) - f(x^{(k)})}{h} \quad \text{and} \quad \frac{f(x^{(k)} + h \cdot \mathrm{e}^{(i)}) - f(x^{(k)} - h \cdot \mathrm{e}^{(i)})}{2h}, \tag{4}$$

where $\mathrm{e}^{(i)}$ is the ith unit vector, and $h > 0$ is a small real number. The approximations in Eq. (4) converge to the true derivative of f at the limit $h \to 0$. In numeric code h cannot be chosen to be arbitrarily small and the error in the approximation can never be eliminated because of truncation and round-off errors in floating-point operations [54,75]. Hence a suitable value for h must be selected whenever computing a finite-difference quotient to minimize this error. Numerical differentiation is usually easy to implement and its time complexity scales linearly with the number of input variables. Nevertheless, the aforementioned presence of truncation and rounding errors makes numerical differentiation an undesirable technique.

Analytic differentiation by hand, and *symbolic differentiation* by a computer algebra system such as Mathematica, provide exact derivatives as mathematical (symbolic) expressions. These are however usually only applicable to conventional closed-form expressions that *e.g.* cannot easily describe programming-language concepts such as loops and control flow. Symbolic differentiation also involves expression swell, where the derivative expression obtained can be significantly more costly to compute than the original expression in terms of computational complexity.

Automatic- or algorithmic-differentiation (AD) extends the computer code implementing a given objective function f by additional arithmetics to compute specific partial derivatives of the involved variables (input, intermediate, and output). AD is exact up to floating-point accuracy, and its so-called *reverse mode* beats numeric differentiation also with respect to (asymptotic) time complexity. AD and gradient-based optimization are the two main ingredients of *differentiable programming*, where solutions to optimization problems are implemented as computer code, differentiated via AD, and optimized using gradient-based algorithms.

We present a summary of the two *modes* of AD in Sections 2.4 and 2.5, and implementation aspects in Section 2.6. For a more complete introduction to AD, see the textbook by Griewank and Walther [76] as a classical resource from the numerical simulation community, or a recent survey by Baydin et al. [54] from a machine learning perspective.

2.4. The forward mode of AD

The forward mode of AD [77] extends each variable a by a variable $\dot{a}$ for its partial derivative in some direction, also called a *tangent*. This is typically achieved by introducing a new dual data type, as exemplified with the *ad-hoc* AD tool sketched in Fig. 2(d), which defines a new class that has a member `tangent` for $\dot{a}$ besides the member `primal` (the ordinary ordered computation or primal program) for a. Whenever some a is evaluated via an operation like $a = b_1 \cdot b_2$, $\dot{a}$ can be evaluated alongside according to the analytic rules of differentiation, like $\dot{a} = \dot{b}_1 \cdot b_2 + b_1 \cdot \dot{b}_2$. Therefore, the elementary operations like `__mul__` have been overloaded in Fig. 2(d). Fig. 2(c) shows how to use such a tool: to compute $\frac{\partial f}{\partial x_i}(x^{(k)})$, all $\dot{x}_j$ are initialized with 0 except for $\dot{x}_i = 1$. Then $y = f(x^{(k)})$ is evaluated using the extended arithmetics and $\frac{\partial f}{\partial x_i}(x^{(k)})$ can be read from $\dot{y}$. The time complexity to compute the full gradient vector $\nabla_x f(x^{(k)})$ is proportional to the time complexity to evaluate f times the number of input variables. Any particular directional derivative $\nabla_x f(x^{(k)})^T \cdot v$ can be computed within a time proportional to the evaluation of f by initializing $\dot{x}_j = v_j$ for all j.

2.5. The reverse mode of AD

The reverse mode of AD [78–80] consists of two phases: first, the program is executed using the ordinary arithmetic operations, but all statements are recorded, usually in a computational graph or a stack-like data structure called the *tape*. After this *primal* run, each primal variable a is extended by an *adjoint variable* $\bar{a} := \frac{\partial f}{\partial a}$. The adjoint variables are successively updated while revisiting the statements in reverse, starting from the output $y = f(x)$ and going towards the inputs x_i. For instance, a primal statement $a = b_1 \cdot b_2$ that forms an intermediate step of the computation of f in the recording phase translates to the following updates in the reversal phase:

$$\bar{b}_1 \mathrel{+}= \bar{a} \cdot b_2, \qquad \bar{b}_2 \mathrel{+}= \bar{a} \cdot b_1 \,. \tag{5}$$

The update of adjoints $\bar{b}_1$ and $\bar{b}_2$ in Eq. (5) reflect the difference between the derivatives of the output f with respect to the values of b_1 and b_2 prior to and after $a = b_1 \cdot b_2$. The adjoint $\bar{a}$ accounts for the dependency of the output f on the value of a, and is itself computed as a result of reverse propagation of adjoints from f to a. Through the combination of forward and reverse propagations for $a = b_1 \cdot b_2$, this two-phase algorithm computes the partial derivatives of f through the chain rule $\frac{\partial f}{\partial b_1} = \frac{\partial f}{\partial a}\frac{\partial a}{\partial b_1} = \bar{a}\frac{\partial a}{\partial b_1}$ and $\frac{\partial f}{\partial b_2} = \frac{\partial f}{\partial a}\frac{\partial a}{\partial b_2} = \bar{a}\frac{\partial a}{\partial b_2}$ and accumulates them in the adjoints $\bar{b}_1$ and $\bar{b}_2$ respectively.

All the adjoint variables are to be initialized with 0, except for the output variable $y = f(x)$, which has $\bar{y} = 1$. After the reverse pass, the adjoint input variables $\bar{x}_i$ contain the derivatives $\frac{\partial f}{\partial x_i}$. The time complexity of the reverse mode relative to the primal computation is independent of the number of input variables, making it faster than forward-mode AD or numerical differentiation when computing the gradient of a scalar-valued function with a large number of inputs variables. However, recording a tape requires a significant amount of memory. The reverse mode of AD is also significantly more difficult to implement. In Fig. 2(e) we differentiate a simple function using the machine learning framework PyTorch [52] using the reverse mode.

2.6. Implementation aspects of AD

Depending on the programming language, the primal program can be extended by AD arithmetics in different ways.

The most straightforward strategy is certainly to substitute any arithmetic operations by calls to an AD library that implements the additional AD arithmetics. Just like the *ad-hoc* forward AD tool of Fig. 2(d), many AD tools make use of polymorphism and *operator overloading* [81–84]. This is a feature of many contemporary programming languages, by which the compiler or interpreter automatically dispatches any calls to arithmetic operators or functions to custom implementations if one of the involved variables is of a custom type. It makes adopting AD as easy as replacing the floating-point datatype (*e.g.,* `double` or `float`) by a type from the AD library.

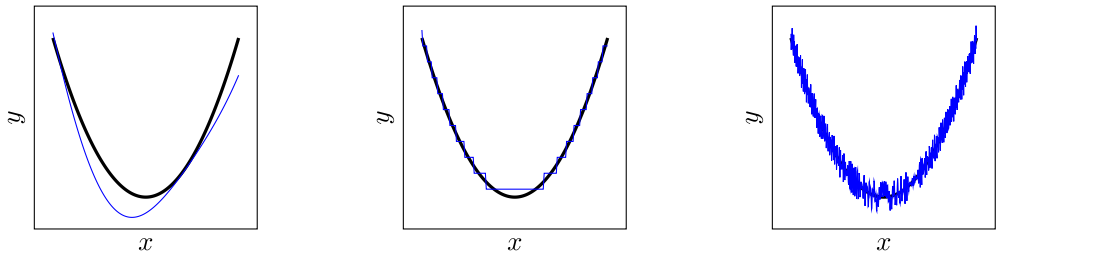
Alternatively, AD arithmetics can be added by a modified or special compiler [85,86] or through *source-to-source transformation* tools [87–90]. An extensive overview of AD tools can be consulted online [91]. In some cases, the entire simulation code needs to be rewritten in the AD framework, which easily makes this approach prohibitive.

In Ref. [92], AD was added to the general purpose matrix element generator MadGraph [93] by using `JAX` [94]. However, some modifications of the original code were necessary for the implementation to work; this is very typical for complex codes, as we outline *infra*.

2.7. Adaptations to the primal program

One common goal in the development of AD tools is to make their integration into an existing primal program as automatic as possible. Problem-specific adaptations of the primal program and the AD workflow can however be necessary, for various reasons such as the following:

- The input/output of the program must be extended to initialize and output the tangent or adjoint variables.
- The primal program might call *external functions* from *e.g.* numerical libraries that are only available in compiled form. Here an analytic, numerical, or surrogate derivative must be provided.



(a) Good. (b) Bad: The derivative is 0, if it exists at all. (c) Bad: The derivative is dominated by high-frequency noise.

Fig. 3. Value-wise good approximations can, but do not need to be, good derivative-wise as well.
Source: Reproduced from Ref. [97].

- Recording every statement might violate memory limits in reverse mode. *Checkpointing* [95] allows the program to re-execute parts of the program so they can be taped just before this chunk of the tape is needed. *Preaccumulation* consists in immediately finding the derivative of blocks with few input and output variables, instead of recording them on the tape. Some numerical algorithms use many operations to solve a mathematically simple problem, and the knowledge of their analytic derivative can be used via external functions;
- When differentiating shared-memory parallel code in reverse mode, shared reads in the forward run translate to concurrent writes in the adjoint updates, Eq. (5). While the AD tool can use atomic updates as a general solution to prevent race conditions, the programmer may manually disable them where the data access patterns allow [96];
- If the primal program only approximates the objective function, there is no guarantee that the (accurate) derivative of the primal program is related to the derivative of the actual objective, as illustrated in Fig. 3. Manual adaptations might be necessary to ensure that we are in the case of Fig. 3(a).
- Related with this caveat and independent of AD, the objective function itself can have properties that make it not amenable to gradient-based optimization, such as having many discontinuities or a very large number of local minima. In such a case, the objective function might be replaced by a surrogate model, as discussed in the next section.

2.8. Surrogate models

Algorithmic differentiation can also be applied to a *surrogate model* instead of the original objective function [58,98]. While the latter can be a very complex code dealing with the very specific design problem at hand, a surrogate model is usually taken from a simple and generic class of functions, such as various neural network architectures commonly used in deep learning. The set of parameters (or weights) specifying the surrogate is determined by means of a fitting (or training) procedure that makes it imitate the original objective.

With a surrogate based on a deep-learning architecture, AD is immediately available within the machine learning framework used to train the surrogate. Note that the surrogate can be differentiable even if the original function is not. In addition, evaluation of the surrogate (and also its derivatives) is usually several orders of magnitude faster [99] than the evaluation of the "true" model, mainly due to vectorization and access to hardware parallelism of GPUs and TPUs available in machine learning libraries. However, training the surrogate requires a substantial number of evaluations of the original function, which does scale at least with the number of design parameters. Also, a poorly trained surrogate that does not reproduce the original function well enough can introduce a bias in the subsequent analysis. We discuss several kinds of suitable neural network architectures in Section 3.2.2.

3. Problem description and possible solution

In this section, we consider the problem of optimizing a customizable objective function for an instrument that employs the interaction of radiation with matter as part of its data-generating processes. The above abstract definition embraces a variety of detectors and instruments and a wealth of use cases in fundamental physics research (including high-energy particle physics, astroparticle physics, nuclear physics, and neutrino physics) as well as industrial applications ranging from hadron therapy or irradiation facilities, to muon tomography scanners for border control, geological monitoring, and archaeological prospections.

Besides the stochastic element provided by particle interactions with matter, which is common to all, the above applications share some specific tasks and lack others. In the following we discuss separately the most important and critical of those tasks, partitioning them in such a way that their interplay may be expected to be the loosest: they may then lend themselves to be studied separately both in a modeling phase and in their separate initial optimization on a reduced set of parameters while fixing (*freezing*) all the others, before a global optimization loop can be proficuously carried out on the full unfrozen system of parameters, together with the other ingredients of the problem.

3.1. Problem statement

An end-to-end detector design optimization task can be briefly formalized in the following way. We start with a simulation of the physics processes of relevance for the considered application, which generates a multi-dimensional, stochastic input variable x, distributed with a probability density function (PDF) $f(x)$. The input is turned by the simulation of the detection apparatus into sensor readouts z distributed with a PDF $p(z|x,\theta)$, which constitute the observed features of the physical process; readouts z depend through $p(\cdot)$ on parameters θ that describe the physical properties of the detector and its geometry. Note that in general $f(x)$, and consequently z, also depend on other latent features—parameters that characterize the underlying physics process and that may not be known precisely. These latent features constitute an additional potential source of systematic uncertainty in the measurement task in addition to the detector-related uncertainties we discuss here; we ignore their existence in the following simplified treatment, noting that the inclusion of their effect in the problem is comparatively straightforward in most cases.

The observations z are used by a reconstruction model $R(\cdot)$ that produces high-level features,

$$\zeta(\theta) = R[z, \theta, \nu(\theta)] \tag{6}$$

(*e.g.* particle four-momenta, in a collider application), by employing knowledge of the detector parameters as well as of the systematic uncertainties affecting the problem, modeled as *nuisance parameters* $\nu(\theta)$ that affect the pattern recognition task [100,101]. In turn, high-level features $\zeta(\theta)$ constitute the typical input of the data analysis step: this is an optional further dimensionality-reduction task, typically performed by a classifier or regressor $A(\cdot)$ powered by a neural network (NN). Once properly trained for the task at hand, the NN produces a low-dimensional summary statistic $s = A[\zeta(\theta)]$ with which inference can finally be carried out to produce the desired goal of the experiment. Suitable optimization metrics derivation follows from that final step: *e.g.,* the power $1-\beta(s)$ of a hypothesis test on the presence of smuggled material in a container, if we are discussing muon tomography for border control (see *infra*, Section 4.3.1); or the total uncertainty in the cross section of production of a new particle, as a simplified proxy to a relevant experimental goal for a collider detector use case.

In general, one may formally specify the problem of identifying optimal detector parameters as that of finding estimators $\hat{\theta}$ that satisfy

$$\hat{\theta} = arg\ min_\theta \int L[A(\zeta), c(\theta)] p(z|x,\theta) f(x) dx dz\ , \tag{7}$$

where we omitted for simplicity to specify the dependence of the high-level features ζ on z and the nuisances ν. Moreover, in the solution above $c(\theta)$ is a function modeling the cost of the considered detector layout of parameters θ, and the loss function $L[A,c]$ is constructed to appropriately weight the result of the measurement in terms of its desirable goals, as well as to obey cost constraints and other use-case-specific limitations. For example, for the aforementioned search of high-atomic-number material smuggled in a container, one might write:

$$L = (1 + e^{(k(c-c_0))}) \sum_Z [u(Z) w_{50}[s(Z)]]\ , \tag{8}$$

where k is an external parameter describing the importance of preventing the cost c from exceeding a given budget c_0, $u(Z)$ is defined *a priori* to weigh the relative importance of successfully detecting material of atomic mass Z in a given benchmark search case involving a set of possible Z values, and $w_{50}[s(Z)]$ is the mass of concealed material for which the power $1-\beta(s)$ to accept the alternative hypothesis (that there is concealed material) in a test at a fixed type-I error rate α (*e.g.,* 0.05) is 50%,

$$w_{50}[s(Z)] = \beta^{-1}(0.5)\ . \tag{9}$$

Since in the cases of interest the PDF $p(z|x,\theta)$ is not available in closed form (as the considered models are implicit), we must rely on forward simulation to sample from it. The problem is solved by approximating $\hat{\theta}$ with

$$\hat{\theta}_a = arg\ min_\theta \frac{1}{n} \sum_{i=1}^{n} L[A(R(z_i)), c(\theta)]\ , \tag{10}$$

where z_i is distributed as $F(x_i,\theta)$ to emulate $p(z|x,\theta)$, as x_i is sampled from its PDF $f(\cdot)$ by the simulator. One may thus obtain an estimate of the loss function and the detector parameters that minimize it.

In order to cast the problem formulated above in a differentiable framework, which makes it possible to search for optimal solutions by gradient descent, it has been demonstrated how it is viable to approximate the non-differentiable stochastic simulator $F(\cdot)$ with a local surrogate model, $z = S(y,x,\theta)$, that depends on a parameter y describing the stochastic variation of the approximated distribution [58]. This allows to descend toward the minimum of the approximated loss $\hat{L(z)}$ by following its surrogate gradient,

$$\nabla_\theta(\hat{L(z)}) = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta L[A(R(S(y_i, x_i, \theta))), c(\theta)]\ . \tag{11}$$

The above recipe requires one to learn the differentiable surrogate $S(\cdot)$: this task can be carried out independently of the optimization procedure.

It should transpire from the above succinct description that the components of the final optimization goal are sufficiently decoupled from one another to allow for modular solutions. So, *e.g.,* the development of a detailed model of event reconstruction

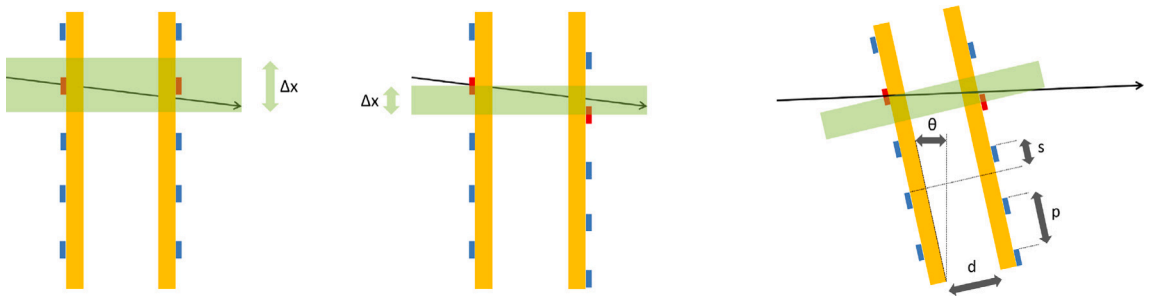


Fig. 4. Left and center: a double-sided silicon strip sensor produces twice smaller resolution Δx on single-strip hit position for an orthogonally incident particle if strips on the two sides are staggered by half the strip pitch. Right: the four parameters affecting single-strip hit position resolution (tilt angle θ, strip pitch p, sensor distance d, staggering s).

that closely matches present-day capabilities in advanced pattern recognition tools, and that allows to obtain high-level features $\zeta = R[z, \theta, \nu(\theta)]$ from the detector outputs z (or approximations thereof, learned by the generative model), can be performed by a separate learning task and then incorporated in the architecture. A similar note concerns the triggering and data acquisition parts of the detection apparatus, most relevant to high-energy physics (HEP) applications: while the online identification and triggering of processes of interest constitute a valid subject for an independent optimization task, they may be incorporated in a simplified way as an independent modeling block with similar techniques to those describing pattern recognition and offline event reconstruction; the output may then be encoded into a set of efficiency maps $\epsilon(z, \theta)$, and the latter included as collected events weights in the global optimization task.

3.2. Modeling of particle detectors

Several tasks of widely varying complexity fall within the scope of the modeling of particle interactions with matter. These range from the simple propagation of individual particles subjected to multiple scattering, which is relevant *e.g.* to muon tomography applications (as discussed in Section 4.3), or the similarly straightforward modeling of charge deposition and collection in silicon strip sensors (which was done in a study of MUonE, see Section 4.1.6), to the enormously complex description of hadronic showers in a calorimeter for a collider detector (Section 4.1.2), or the modeling of beam-induced backgrounds in a detector for a high-intensity muon collider (Section 4.1.5).

3.2.1. Modeling the geometry of detection elements

A continuous model of the layout of detection elements is straightforward to construct for most applications, and the methods to do so are rather general. Indeed, for most of the use cases of our interest (a selection of which is provided in Section 4 *infra*), we may employ solutions similar to those of existing implementations developed for full [102–105] or fast [106] detector simulations. Detector simulators are typically based on Monte Carlo simulations, that is, they are intrinsically stochastic: there are, however, recent ongoing efforts to make simulators like GEANT4 fully differentiable [97]. The challenge is however to identify what other parts of the global pipeline are most affected by variations of the geometry of the instrument, such that we may choose a parameterization that allow a simpler functional mapping of the geometry parameters into those dependent features. An example will clarify this point.

Consider a set of silicon tracking layers made up of two strip sensors glued back-to-back, such as those that constitute the MUonE detector (described in Section 4.1.6). Almost any relevant figure of merit for such a detector is heavily dependent on the resolution of the track parameters, which may be extracted from a fit to the reconstructed hit positions. We define as *hit* the binary signal given by an electronic module that fires when a particle passes through it. The latter will in turn be subjected to systematic uncertainties due to positioning and alignment of the sensors: there is thus a clear path through which these systematic uncertainties affect the utility function. For silicon strip sensors, the root-mean-square hit resolution on the coordinate orthogonal to the strips equals $p/\sqrt{12}$, where p is the pitch, if the particle ionization causes the system to detect a single-strip hit[4]; however, a significant resolution gain is obtained if particles create multiple-strip hits, as charge sharing then provides added information on the track intercept. For particles moving along the beam axis (here taken as the z coordinate) or very close to it (the vast majority in MUonE, and also those that are most relevant to extract the final measurement of high-q^2 scattering cross section, which is the goal of that experiment), the fraction of multiple-strip hits has a functional dependence on four geometry parameters: pitch, z-spacing of the glued sensors, staggering of strips on one sensor relative to the other, and (if allowed) tilt angle of the sensors with respect to the z axis, as illustrated in Fig. 4. An optimization study of the placement of those detection elements which varied those four variables independently would waste significant resources to investigate their functional dependence at resolution minimum, which may be instead determined in isolation from the rest of the problem (and potentially more accurately) through a dedicated full simulation, and thus inform

[4] Estimated single-strip hit positions distribute with a uniform density function, which in a differentiable study can easily be parameterized by using a constant connected to sigmoid functions at the extrema.

a constrained parameterization that keeps the four parameters bounded within the subspace of optimal charge-sharing conditions. The smaller-dimensional description would also allow a simpler tracing of position and alignment-related systematic uncertainties. This example shows how domain knowledge—in this case, insights in the operation of a detector along with the inference extraction procedures applied to its output—may help find the most suitable data representations for an optimization task, and how this may be beneficial on the whole.

Surrogates based on deep generative models learn the distribution of data (either explicitly or implicitly, see below) and sample new events from that distribution. Data either come from experiment or from dedicated simulations, for example using GEANT4 [102–104] or other alternatives [107], depending on the specific case at hand.

Energy depositions in the continuous space of the detector are then discretized into voxels representing read-out channels, forming the feature space z in which training is carried out. The number of these voxels, usually of the order $\mathcal{O}$(100–1000), should be chosen to be as large as possible to have fine-grained description of the detector module at hand. However, deep generative models do not work with an arbitrarily high number of dimensions, so the number of voxels cannot be too large. Current state-of-the-art calorimeter shower simulations uses at most $\mathcal{O}(10^4)$ voxel: Ref. [108] uses 65k voxels and Ref. [109] uses 27k voxels. The problem can be mitigated to some extent by using aggressive parallelization and memory offloading based on CUDA kernels,[5] but more radical and effective solutions will require dedicated hardware such as neuromorphic chips [110] for energy-efficient processing or quantum computers [111] for quantum supremacy (optimizing a problem of a given complexity faster or with less computational resources).

In order to be differentiable in the additional parameters θ that describe the physical properties of the detector, its geometry, or other features of interest for the optimization task, the models have to be trained with θ as a conditional label [112]. Examples for θ are discussed in Section 4.1.2 and include thickness, position, angle, or size of detector modules (as long as the number of voxels remains constant); tuning parameters of GEANT4; or atomic number Z of the detector material. Conversely, there are a few parameters that cannot be studied. Since the dimensionality of the feature space is hard-coded into the models and given by the training data, the number of voxels is fixed. The resulting augmented dataset (z, θ) is then used to train one of the deep generative models described in the next subsection.

3.2.2. *Learning the simulated particle interactions with matter through surrogates*

The models which we will consider for learning the datasets mentioned *supra* strongly depend on the situation at hand. As an illustration, we focus on modeling the shower produced by an energetic particle in a calorimeter for a collider detector with deep generative models. Based on the similarity of voxelized calorimeter showers and pixelated digital images, all deep generative models used to generate artificial images can also be used to learn calorimeter showers, albeit some modifications improve the quality of the samples. Note that the task is always to reproduce the shower of a single incoming particle in a given submodule of a larger detector. Such showers are statistically independent of each other and several of them can be combined into the simulation of a more complicated, single event.

Current, state-of-the-art deep generative models are based on one of the following three main architectures: Variational AutoEncoders (VAEs) [113–115], Generative Adversarial Networks (GANs) [116,117], and Normalizing Flows (NFs) [118–120]. Each of them has its own advantages and disadvantages, ranging from memory footprint, to training time and stability, to sampling time, and sampling quality. The best surrogate model is one that is both fast and faithful, however, in practice there is a trade-off between these metrics.

VAEs [113–115] build upon simple autoencoder (AE) structures. AEs consist of two neural networks, an encoder e and a decoder d. The encoder maps the n-dimensional data to an m-dimensional latent space, where usually $m \ll n$. The decoder maps the latent vector back into data space, making the entire AE architecture (the combination of encoder and decoder) a bottleneck. Training of an AE is done by minimizing the reconstruction loss between the input data x and the encoder–decoder-transformed data: $d(e(x))$. By randomly sampling points in the latent space and passing them through the decoder, the architecture becomes a generative model. In a perfect AE, the NN learns a lossless compression of data to latent space. This, however, can become problematic if the NN overfits and simply remembers a one-to-one mapping between data and latent space. In this case, the AE performs poorly as a generative model, as the latent space only remembers the data and newly-sampled points confuse the decoder. A VAE avoids these issues by giving structure to the latent space, thereby regularizing it. The decoder now maps the input to $2m$ numbers, that are understood as mean and standard deviation of an m-dimensional, multivariate Gaussian. A pass through the decoder–encoder chain now involves a sampling step in the latent space. The loss function of the VAE consists of two terms: the first one is the reconstruction error of the AE architecture, the second one is a Kullback–Leibler (KL) divergence [121] that compares the latent space distribution to an m-dimensional standard normal distribution. A relative weight between the two terms emphasizes a more regular latent space, *i.e.* a smooth morphing of one data point into another through a trajectory in the latent space, or a higher sample quality. Sampling is usually fast, as generation only involves a single pass through the decoder network. Without further modifications or post-processing, samples from VAEs usually have comparatively low quality. This setup, including modifications and post-processing, has been considered for calorimeter simulations in HEP [109,122–127].

GANs [116,117] train two NNs, a generator and a discriminator, in an adversarial objective. The generator tries to generate realistically-looking images while the discriminator tries to separate them from real ones. The resulting saddle-point optimization objective is harder to train, resulting sometimes in mode-collapse and artifacts in samples. A significant improvement in performance can be obtained by using a Wasserstein GAN (WGAN) [128,129]. In this case, the discriminator is replaced by a critic that evaluates the Wasserstein distance of a given generated sample from the training data. Model selection and evaluation of GANs is a difficult

[5] https://developer.nvidia.com/blog/cuda-refresher-cuda-programming-model/.

task [130], as the loss value of the critic in training does not always correlate with the sampling quality. Once trained well, GANs generate realistically-looking samples across various domains. Hence, there have been several applications of GANs to calorimeter shower simulation [108,109,122,126,131–154].

NFs [118–120] learn a bijective mapping between two distributions, which are usually the (complicated) data distribution on one side and a (computationally easy) base distribution—such as a standard normal distribution—on the other side. The bijector does not only provide the coordinate transformation between the two distributions, but also the Jacobian of the transformation, making NFs suitable for many different applications: when points from the data-space are mapped to the base distribution, the Jacobian of the transformation, together with the probabilities of the points under the base distribution, give a probability of the original points, thus making NFs density estimators. In the inverse direction, noise generated under the base distribution can be mapped to data-space: the NF then acts as a generative model. Since the log-likelihood of the data points is available by construction, NFs can be trained by minimizing the negative log-likelihood. The resulting training is usually very stable and gives a reliable density estimator, if the NF is expressive enough. Such a model, when used as a generative model, does not suffer from mode collapse. In addition, the log-likelihood does provide a good metric for model selection that directly correlates with the quality of the fit to the data distribution. Requiring the transformations to have an analytic inverse and tractable Jacobian puts constraints on the NN architecture realizing them. State-of-the-art bijectors consist of a series of spline-based transformations [155], the parameters of which are predicted by NNs. To further ensure that the Jacobian can be evaluated in linear time, the architectures must be either bipartite [156] or autoregressive [157,158]. The former run equally fast in both directions (density estimation for training and sampling generation for application), the latter strongly favor one direction over the other. Masked Autoregressive Flows (MAFs) [157] are fast in density estimation, but slower in sampling by a factor d, given by the dimension of the space to be learned. Inverse Autoregressive Flows (IAFs) [158] are fast in sampling, but a factor d slower in estimating the density of data points. Only recently, NFs (based on MAF and IAF architectures) have been applied to calorimeter shower simulation [159,160], surpassing the quality of showers generated by an older GAN trained on the same dataset [132,133].

Modeling calorimeter showers with deep generative models is a very active area of research with considerable interest of the community, as can be seen at the "Fast Calorimeter Simulation Challenge 2022" [161].

3.3. Modeling of pattern recognition and event reconstruction procedures

While the simulation of multiple particles with matter can be factorized on a particle-by-particle basis and the corresponding deposits can be superimposed, the subsequent reconstruction of their patterns can be highly affected by correlations between particles, *e.g.* through spatial overlap. These correlations can in principle even span over whole detectors or sub-detectors. This poses more stringent requirements on the resources to model this step.

While typical detectors or sub-detectors consist of hundreds to millions of individual sensors to achieve the resolution needed to measure the quantities of primary interest, the latter usually belong to a set orders of magnitude smaller than the number of detector inputs. Therefore, the deposits in sensors left by particles traversing the detector or being stopped by detector elements (hits) are used to reconstruct those very same particles as so-called physics objects, which relate more directly to the quantity of primary interest, such as track hits in tracking detectors, or full particle candidates in particle-flow algorithms [8,162–170]. This concept of pattern recognition has proven to be very powerful, however most algorithms suffer from inherent limitations with respect to their differentiability, as they rely heavily on seeding mechanisms that select a certain area or point of interest as starting point for a reconstruction algorithm based on a certain particle type assumption. Then, the reconstruction is subsequently refined in steps, each individually optimized and coming with its own selection thresholds. As such, this reconstruction procedure contains many nondifferentiable steps, and furthermore does not generalize easily if the detector geometry or the individual sensor properties are changed. Therefore, these algorithms can introduce biases towards those detector designs they were originally developed for, which makes them not applicable for a generic differentiable detector design optimization.

For a truly generic reduction of dimensionality from hits to physics objects, the algorithm needs to be geometry agnostic, be differentiable, encode generic physics considerations, and adhere to the given computing resources. Machine-learning algorithms can offer this flexibility as they are typically differentiable by construction, and can also easily adapt to changing conditions. However, neither those algorithms that rely on a regular geometry (such as convolutional neural network based approaches [171–175]), nor algorithms based on dense neural networks alone that make no assumptions at all on the structure of the problem, are applicable: the former cannot generalize to irregular geometries, and the latter cannot fit resource constraints. In addition, recurrent algorithms cannot be made really generic, as conceptually they rely on a certain ordering of the inputs.

The only viable option to date to handle this high-dimensional and irregular input space are graph neural networks [176] (GNN). Compared to other classes of neural networks, GNNs are a quite recent development, but have already proven to be very powerful, also in the physics domain [177–184]. GNNs do not require a certain type of input ordering or regularity; they solely rely on a set of input points that can represent the detector hits, and connections between them. These connections can simply be chosen as nearest neighbors in a physical or a latent space. This procedure has two advantages: it encodes a notion of locality directly into the network architecture, and therefore maps the physics of locally propagating particles through the detector; and it also comes with advantages in terms of computing resources, since the number of operations to be performed does not grow quadratically with the number of inputs anymore, as it is the case for a dense neural network approach. The exception are fully connected graph neural networks, which might be applicable for a small number of inputs, but cannot conceptually work for more complex detectors. Such neural network architectures have already shown their potential for tracking [180] and calorimetry [177,181].

However, none of the neural network architectures themselves perform a reduction of dimensionality; the dimensionality reduction is performed in a second step. First approaches use edge classifiers, which learn a score that determines whether or not a connection between two neighboring hits corresponds to a connection between hits of the same object, and then the objects get segmented by following edges with scores above a certain threshold [180]. The object properties are then derived in a second step based on the selected hits. This is a natural way to approach a tracking problem, with clearly separable objects, but the paradigm breaks with larger overlaps, and when object sizes are similar to the spatial detector resolution, *e.g.* in calorimeters. The object condensation approach has been recently proposed as a way to overcome this limitation [32]. It is being used already for reconstruction in the CMS HGCAL [181,185] and machine-learning driven particle flow in CMS [186]. Here, the object properties are directly accumulated in representative so-called condensation points, and a simple clustering in a learned clustering space resolves ambiguities and performs the dimensionality reduction, without the need for full segmentation. This is done by collecting hits around points with a large condensation score in the learned clustering space.[6]

Neither the graph neural networks nor the training procedures discussed above make any conceptual distinction between hits in different detector systems. Therefore, these approaches also provide a basis for investigating designs that break with existing paradigms, such as the hybrid calorimeters discussed in Section 4.1.3.

Once the patterns of individual particles are identified using information from the whole detector, and the information is reduced in dimensionality by orders of magnitude, it can be either passed on directly to information-extraction procedures, or a more classical approach can be taken *e.g.* by clustering jets and deriving higher-level quantities such as, *e.g.*, missing (transverse) momentum in collider experiments. The latter, being a simple 4-vector sum, can trivially be part of a fully differentiable pipeline; the jet clustering procedure, on the contrary, incorporates non-differentiable assignments of a particle to either one jet or the other by introducing cut-off parameters. However, while the assignment is not differentiable, gradients for all particle properties can be passed through the calculation of the final jet properties. Moreover, by introducing weights in the clustering, it is also possible to equip the cut-off parameters of the jet clustering themselves with effective gradients. Therefore, the full pipeline from hits to higher-level inputs to the information-extraction procedures used to determine the final quantities of interest can be made differentiable.

3.4. Modeling of information-extraction procedures and detector-related biases

A crucial ingredient in the global optimization task for a measuring instrument is a precise model of the conversion of high-level information (produced by the pattern recognition and event reconstruction tasks described *supra*) into the summary statistics which either are by themselves the final product of the measurement procedure, or constitute the direct input for its extraction. Here the problem displays a new layer of complexity, as the design choices of a measuring instrument have implications on the existence of biases and imprecision in the measurements, which can only partly be corrected by calibration procedures or detailed simulation studies. The resulting uncertainties propagate directly into a worsening of the final performance, and must therefore be included in an optimization pipeline. In this section, we discuss how those effects can be tamed by methods that themselves employ differentiable programming solutions. The inclusion of these inference extraction procedures in the optimization task can thus not only properly account for the impact of design choices on the final metrics, but also allow for an alignment of the optimization of the whole system with the most performing inference strategies.

3.4.1. Systematic-aware summary statistics

In the last decade, classification and regression models have become very popular in HEP to construct powerful summary statistics that are used for inference. This is largely due to the presence of high-fidelity simulators that provide us with accurate models of underlying physical processes, which we can use as training data. However, standard machine learning-driven loss functions become misaligned with respect to physics goals when the simulated observations are affected by systematic uncertainties. Given this, it is then desirable to seek an objective function that is "systematics-aware", which can optimize any set of free analysis parameters, including learnable components like neural networks.

This issue has been first addressed by the INFERNO algorithm [187], that aims at directly minimizing the expected variance of the parameter of interest (POI), accounting for the effect of relevant nuisance parameters, the latter describing the systematic effects. The parameters of a neural network are optimized via SGD using automatic differentiation. A sketch of the INFERNO algorithm is shown in Fig. 5. An inference-aware summary statistic is learnt by optimizing the parameters $\boldsymbol{\phi}$ of a neural network f performing a reduction of dimensionality of the input data $\boldsymbol{x}$:

$$f(\boldsymbol{x};\boldsymbol{\phi}) : \mathbb{R}^d \rightarrow \mathbb{R}^b \ . \tag{12}$$

The network is trained with batches of simulated samples G_s obtained from a simulator g with parameters $\boldsymbol{\theta}_s$. The number of nodes in the last layer of the network determines the dimension b of the summary statistic. Since histograms are not differentiable, the original algorithm uses a softmax function as a differentiable approximation for the neural network output y:

$$\hat{s}_i(\boldsymbol{x};\boldsymbol{\phi}) = \sum_x \frac{e^{f_i(\boldsymbol{x};\boldsymbol{\phi})/\tau}}{\sum_{j=0}^{b} e^{f_j(\boldsymbol{x};\boldsymbol{\phi})/\tau}} \tag{13}$$

[6] In a fully differentiable pipeline one could omit the ambiguity, by resolving clustering and feeding the points with high condensation score directly to subsequent steps.

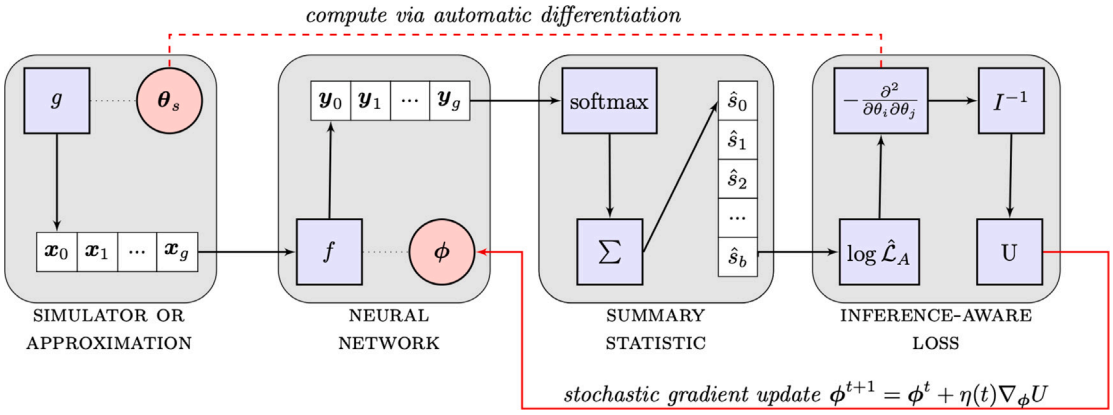


Fig. 5. Sketch of the INFERNO algorithm. Batches from a simulator are passed through a neural network and a differentiable summary statistic is constructed that allows to calculate the variance of the POI. The parameters of the network are then updated via SGD.
Source: The figure is reproduced from Ref. [187].

where the temperature hyper-parameter τ regulates the softness of the operator. In the limit of $\tau \to 0^+$, the probability of the largest component will tend to 1 while others to 0. With this approximation it is possible to construct a summary statistic for each batch by computing the Asimov Poisson-count likelihood $\hat{\mathcal{L}}_A$:

$$\hat{\mathcal{L}}_A(\boldsymbol{\theta};\boldsymbol{\phi}) = \prod_{i=0}^{b} \text{Pois}\left(\hat{s}_i\left(G_s;\boldsymbol{\phi}\right) \mid \hat{s}_i\left(G_s;\boldsymbol{\phi}\right)\right) \ . \tag{14}$$

where with the HEP jargon term *Asimov* we indicate that the value of $\hat{\mathcal{L}}_A$ is computed with the expected values based on the simulated samples G_s, such that the maximum likelihood estimator for the Asimov likelihood is the parameter vector $\boldsymbol{\theta}_s$ used to generate the simulated dataset G_s. In Statistics, this is also referred to as a saturated model. From the Asimov likelihood the Fisher information matrix is then calculated via automatic differentiation, according to

$$\boldsymbol{I}(\boldsymbol{\theta})_{ij} = \frac{\partial^2}{\partial\theta_i\partial\theta_j}\left(-\log\hat{\mathcal{L}}_A(\boldsymbol{\theta};\boldsymbol{\phi})\right) \ . \tag{15}$$

The covariance matrix can be estimated from the inverse of the Fisher information matrix if $\hat{\boldsymbol{\theta}}$ is an unbiased estimator of the values of $\boldsymbol{\theta}$:

$$\text{cov}_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}) \geq I(\boldsymbol{\theta})^{-1} \ . \tag{16}$$

It is also possible to include auxiliary measurements that constrain the nuisance parameters, characterized by likelihoods $\left\{\mathcal{L}_C^0(\boldsymbol{\theta}), \ldots, \mathcal{L}_C^c(\boldsymbol{\theta})\right\}$, by considering the augmented likelihood $\hat{\mathcal{L}}'_A$:

$$\hat{\mathcal{L}}'_A(\boldsymbol{\theta};\boldsymbol{\phi}) = \hat{\mathcal{L}}_A(\boldsymbol{\theta};\boldsymbol{\phi}) \prod_{i=0}^{c} \mathcal{L}_C^i(\boldsymbol{\theta}) \ . \tag{17}$$

The loss function used for the optimization of the neural network parameters $\boldsymbol{\phi}$ can be any function of the inverse of the Fisher information matrix at $\boldsymbol{\theta}_s$, depending on the concrete inference problem. The diagonal elements $I_{ii}^{-1}\left(\boldsymbol{\theta}_s\right)$ correspond to the expected variance for the parameter θ_i. Thus, if the aim is efficient inference about one of the parameters $\omega_0 = \theta_k$ a possible loss function is

$$U = I_{kk}^{-1}\left(\boldsymbol{\theta}_s\right), \tag{18}$$

which corresponds to the expected width of the confidence interval for ω_0 accounting also for the effect of the other nuisance parameters in $\boldsymbol{\theta}$. The algorithm performance was originally studied with a synthetic example inspired by a typical cross section measurement. In that setup, it was shown that the confidence intervals obtained using INFERNO-based summary statistics were narrower than those obtained using binary classification, and tended to be closer to those expected when using the true model likelihood for inference. The improvement over binary classification was also seen to increase when more nuisance parameters were considered. Recently, INFERNO has been re-implemented in `PyTorch`, allowing for its use in a drop-in fashion [188] and enabling the use of approximately differentiable histograms with sigmoid functions. By making use of interpolation techniques, and a suitable preprocessing of the systematic variations, the INFERNO algorithm has been extended to run with realistic HEP-like systematics. First studies of the algorithm with real data based on CMS Open Data indicate that the algorithm is able to mitigate the impact of systematic uncertainties also in realistic LHC analyses [189].

More recently, several methods have been developed that build upon the ideas behind INFERNO. A promising approach is taken by the authors of `neos` [190,191], who aim at directly optimizing the expected sensitivity of an analysis. This also serves to minimize the uncertainty on the maximum likelihood estimate differently compared to INFERNO, since the sensitivity is calculated from the

distribution of a test statistic that is monotonically related to the MLE, as shown in Ref. [192]. To approximate histograms, `neos` uses a binned version of a kernel density estimate (KDE), where the smoothness is determined by the bandwidth parameter of the KDE. To get a differentiable analysis sensitivity as targeted by `neos`, one can leverage fixed-point *implicit differentiation* in order to calculate expressions for the gradients of maximum likelihood estimates, as found in the profile likelihood. In particular, these gradients only involve one update step of the optimization procedure, and do not require the costly unrolling of the entire optimization loop during back-propagation. `neos` takes advantage of recent progress in the `pyhf` package, which implements full HistFactory likelihoods and their inference using many different automatic differentiation backends, *e.g.* `TensorFlow`, `PyTorch`, and `JAX` [193,194].

Despite their close similarity, differences clearly exist in the implementations of INFERNO and `neos` from a software standpoint. To this end, work is being done on a library that serves as a toolbox of drop-in differentiable operations that are designed for use in this kind of workflow, called `relaxed` [195]. This will facilitate design of a pipeline with easily exchangeable components, allowing for more flexibility in algorithm design, and easier replicability through a unified implementation. Moreover, both INFERNO and `neos` have been separately implemented with this library, and one can trivially combine elements of each approach to reach a potentially more optimal analysis pipeline.

In the context of a complete detector optimization task, the integration of a differentiable analysis workflow in a complete pipeline as above may be tricky. A way to perform this in practice is to simplify the problem, by considering the inference extraction task as one to first order independent on the system's parameters, and to independently optimize the inference step assuming frozen values for those geometry and detector-related parameters (such as, *e.g.,* calibration errors, imperfect efficiency maps, alignment and positioning accuracy) which introduce potential imperfections in the model. The trained dimensionality reduction model may then be integrated in the global pipeline, and only updated when significant changes occur to the value of parameters to which the model is most sensitive. Future studies are needed to test the most advantageous ways to include the inference extraction step in an end-to-end optimization task.

3.5. Modeling the cost of detectors

Monetary cost plays a key role in the conception of any detector and acts as a major constraint in terms of technology and design choices. In this context, detector optimization cannot rely exclusively on physics performance features such as resolution or efficiency. Along with case-specific technical constraints, construction cost has to be implemented in the loss function (see Section 3.1) to set boundaries to the parameter space and guarantee the feasibility of the project. In order to preserve the adaptability of the optimization to any experiment, one can compute the effect of construction costs on the loss function $L_{cost} = c(\theta, \phi)$ in two main steps, each of them dependent on different sets of parameters:

- Local cost parameters θ are specific to the technology used: *e.g.* active components material, photo-detection and light transport techniques.
- Global cost $c(\theta, \phi)$ can be expressed as a function of local cost parameters θ and a set of parameters ϕ describing the overall detector conception, such as number and size of detector modules and their respective positions.

Modeling the dynamics of global cost with respect to local costs may seem unfeasible for large scale detectors such as experiments at the LHC or future colliders (see Section 4.1), but can surely be done for setups of moderate complexity such as cosmic-muon trackers for muon radiography (see Section 4.3), and it is indeed one of the features being included in the TomOpt package described *infra* (Section 4.3.3).

For most applications, a similar detection performance can be achieved by several different technologies. For each of them, one must establish relations between their local cost parameters θ and their physics performance γ. In a first approximation, this can be done by considering a detector as made up of two separate modules: an active detection system and its related electronics.

- The choice of an active detector module fixes most of the detector performance parameters γ, such as spatial and time resolution. Its cost C should be expressed as a function of these performance proprieties, and normalized to its active area/volume and number of readout channels:

$$C_{technology} = C(\gamma)\ [\ m^{-2}.\ readout^{-1}\ ] \tag{19}$$

- Electronics: In many detectors, front-end electronics along with data acquisition systems account for the largest share of detector cost, which makes their cost estimation critical. To ensure compatibility with active detector cost, electronics cost should be estimated per channel.

Such a splitting of the cost is done under the assumption that it scales linearly with surface or volume and the number of readout channels, which is likely to be a fair approximation for simple setups; for large-scale detectors a more complex model should be considered.

A complication of the above scheme comes from the fact that the total monetary cost of a detector is not only a function of specification and number of its components: other fixed expenses such as infrastructure, laboratory equipment, and maintenance can occasionally be significant. Nevertheless, modeling such costs and including them in the optimization process is not necessary as long as their dependence on detector performance parameters is limited. Besides, infrastructure costs might vary from one technology to another, hence it would be more pertinent to evaluate them aside from the optimization phase. These fixed costs can then be added to the variable cost modeled in the loss function.

3.6. Modeling of external constraints

Often a detector design task needs to consider, in addition to performance and cost of the instrument, external constraints coming from a number of specific conditions that characterize the project. For example, the construction of the detectors of the LHC was conditioned by engineering constraints connected with the placement and operation of the instruments in underground caverns; these, *e.g.,* affected the largest size and weight of elements that could be assembled on the surface before lowering them down in the caverns; pieces built externally by participating laboratories or contractor industries also posed considerable logistic challenges connected to their transportation to CERN. Other physical constraints are often connected with power consumption and payload (factors of high relevance for detectors to be operated in space), operation temperature and cooling infrastructure, online computing and data acquisition limits. Further, for large-scale projects even the sheer availability of construction materials may play a role, and require its consideration as a constraint in a global optimization task.

A different kind of external constraints often comes from the timeline of the projects. At the end of the nineties, when upgrades to the CDF and DZERO detectors were being planned prior to the start of Tevatron Run 2, the construction schedule of the LHC was a very important ingredient affecting the decisions that the laboratory took on how much to invest in upgrades which could grant Fermilab a fighting chance to discover the Higgs boson before the European machine would take over with its larger energy and luminosity. A careful modeling of commissioning timelines may in similar situations heavily affect the absolute value of attainable goals depending on their expected time to completion, and cannot therefore be ignored in a serious optimization study.

While the above examples could fuel criticism toward the naivety of the idea that an automated scanning of detector design configurations can provide significant help to the hand of the expert detector builder, we argue that in fact everything which adds to the complexity of the task only strengthens the value of complete modeling approaches. In fact, all the mentioned constraints are relatively straightforward to insert in a differentiable pipeline, and in most cases they are also only weakly coupled to the other ingredients, making their inclusion less impacting and simpler. At any rate, as we already argued in Section 3.5, we believe that whenever some external factor is impossible or inconvenient to include in an optimization pipeline, one needs not worry about it too much. It is already very helpful to optimize the system under design for various scenarios, to pin down the most optimal set of parameters in each of them and reduce the complexity of the decision to a comparison between a few discrete options, where non-quantitative and even political or sociological arguments can find their place.

4. Example use cases

In this section, we consider the specific modeling needs of instruments designed for a variety of different goals, ranging from pure research to industrial and medical applications. Our choice of illustrative use cases, which is far from being exhaustive, is driven by the need to clarify, to ourselves and to the reader, how a modular optimization pipeline such as the one described in Section 3 may be customized and adapted to very different problems, with only a minor reconfiguration of its basic ingredients and minimal changes in the optimization methodology. These problems all constitute interesting benchmarks to us because of our specific focus on those research areas.

We start in Section 4.1 with a discussion of a representative selection of use cases for AD-powered design optimization taken from accelerator experiments. In Section 4.2 we discuss some use cases from astro-particle and neutrino physics, which offer a variety of additional complications and intriguing problems to solve. In Section 4.3 we consider in detail the use case of muon tomography, which is an excellent test-bed for the development of a full optimization pipeline, given the relatively simple physics involved and the well-defined nature of the optimization target. Section 4.4 describes the special optimization challenges of instruments designed for proton-computed tomography. In Section 4.5 we offer an example drawn from research in low-energy particle physics, where one tries to optimize the transport of cold neutrons. We complete our pot-pourri of potential applications of differentiable programming to design optimization in Section 4.1.8 with the consideration of how to optimize the calculations in lattice QCD.

4.1. Experiments at accelerators

Fundamental research has driven the need of accelerating particles and collide them at ever-increasing energies, to study the products of the resulting collisions with the hope of understanding the fundamental structure of nature and detect the presence of physics processes never observed before, if they exist. Physicists have used the data from these collisions to identify elementary particles, directly or indirectly observable, and understand their interactions [196,197]. To do so, they have built particle detectors of ever-increasing size, which detect final-state particles by exploiting their interaction with matter. The particle detectors needed to collect data taken from collisions at the energies reached by the LHC [8] have reached an unprecedented level of complexity; a full optimization of the next generation of particle detectors with a differentiable pipeline is a very ambitious, long-term goal for our studies [5]. Particle accelerators and detectors have originally been designed for fundamental research, but the technologies that drive their functioning have quickly been adapted to a vast range of other applications in scientific, medical, and industrial fields.

The optimization of an entire accelerator or detector is a daunting task that is probably still beyond our present-day capabilities. Nevertheless, differentiable-programming-based optimization has been successfully applied to the optimization of portions of these machines and to their automated control. In Section 4.1.1 we outline existing work and future perspectives for the design and control of particle accelerators; in Section 4.1.2 we consider the optimization of the electromagnetic calorimeter of the LHCb experiment; in Section 4.1.3 we describe how one could approach the task of optimizing a hybrid calorimeter design for a future collider; and

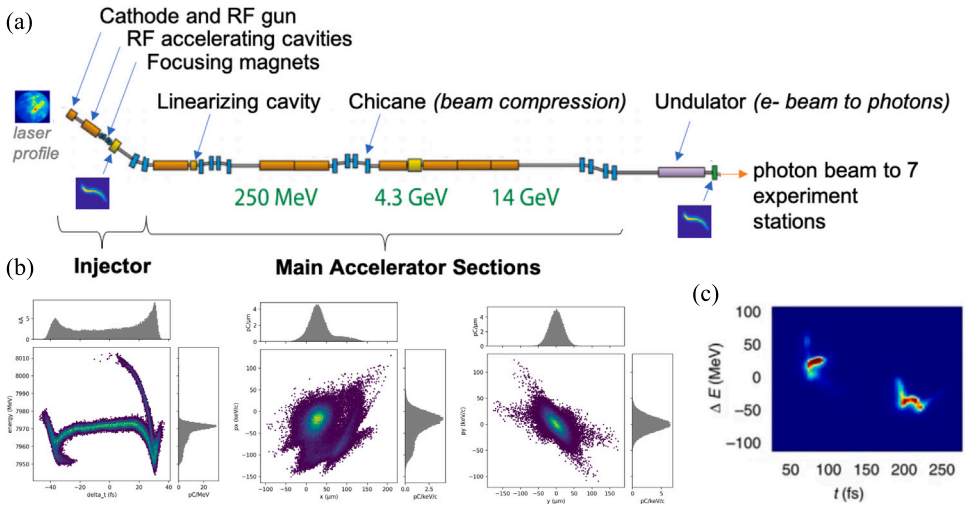


Fig. 6. Example of a linear accelerator and major components (in this case the Linac Coherent Light Source) (a). Examples of 2D projections of 6D beam phase space, from physics simulations in `Bmad` [198] (b). Example of measured longitudinal (duration vs. energy) phase space in an operating mode with two electron bunches (c).
Source: Reproduced from Ref. [199].

in Section 4.1.5 we discuss the optimization of an electromagnetic calorimeter for a future muon collider experiment. We conclude our survey with Section 4.1.6, where we describe the optimization of MUonE, a proposed detector with a relatively simple tracking and calorimetry geometry, and Section 4.1.7, where we describe the perspectives for a cost-effective optimization of the MilliQan detector.

4.1.1. Particle accelerator design and control

Particle accelerators are employed for a wide range of scientific, medical, and industrial applications, including high-energy particle physics experiments. Optimization and control of particle accelerators is challenging: these instruments have many interconnected sub-systems (*e.g.* low-level dynamic control of radio-frequency cavities, high-level optimization of focusing/steering magnets and cavity settings), which rely on adjustable settings often consisting of hundreds-to-thousands of variables, and in many cases display highly non-linear beam responses to different combinations of input settings. In addition, there are time-varying inputs and responses that must be taken into account, both due to unintended drifts (*e.g.* due to temperature changes) and deliberate changes in state (*e.g.* to achieve different beam parameters). The challenge of optimizing these systems both in design phase and during operation increases as we push toward the energy and intensity frontiers of beam physics, where the beam responses become increasingly nonlinear and sensitive to machine settings, noise, and other sources of uncertainty.

The beam itself is typically represented in physics simulations as a cloud of particles in 6D position-momentum phase space. In the past, using bulk statistical scalar metrics of the beam distribution (such as energy spread or bunch duration) was sufficient for optimization in many applications. However, increasingly the full information about the phase space distribution is needed to meet the tolerances of applications at the energy and intensity frontiers. For example, in high-energy particle beams, the beam halo at the edges of the beam distribution can contain enough energy to damage accelerator components. Advanced phase space manipulation techniques to produce, for example, beams that are highly elongated in one dimension is often challenging due to the sensitivity of the many correlated responses of the beam to different accelerator components. The need to precisely optimize (in design phases) and then control the beam phase space distribution in detail is common across a wide variety of accelerator types and applications, ranging from light source user facilities, to isotope production facilities, to colliders. An example of an accelerator and particle beam phase space distribution is shown in Fig. 6.

Challenges for accelerator design and online optimization. The initial design of accelerators is driven by detailed physics simulations of the accelerator components (such as the type of RF cavities) and, to varying degrees of detail, simulations of the nominal range of accelerator settings and placement of components as a whole. Automated multi-objective optimization is an essential component of this process and has traditionally used heuristic methods, such as genetic algorithms [200] and particle swarm optimization, coupled with high-performance computing. Accelerator physicists then examine beam parameter tradeoffs at different working points when making design decisions (for example, examining the achievable beam intensity at different beam energies).

While this design paradigm has a long history of success, the computational expense of most simulations that include the majority of expected nonlinear collective beam effects makes this process a slow and resource-heavy endeavor. Simulations that include nonlinear collective beam effects typically use computations on hundreds-of-thousands of individual particles in a particle beam to predict its evolution along the accelerator. Increased execution speed can be achieved by making simplifications to simulations

(*e.g.* not including all expected beam effects, or including only the most significant accelerator variables in optimizations) at the cost of accuracy and comprehensiveness.

The design process and online optimization are also segmented at present: many accelerator design optimizations examine the tradeoff between only two-to-three scalar beam properties at a time, online optimization often examines only a few output parameters as objectives at a time, and sub-systems are often optimized sequentially or semi-independently (*e.g.* injector systems are optimized separately from down-stream sections). This approach is known from simulation studies to produce sub-optimal results, in contrast to simultaneously optimizing settings across the entire accelerator [201]. Emerging accelerator designs that have more stringent tolerances on the beam parameters and involve operation at the energy and intensity frontiers cannot rely on such simplifications to the same degree as older or more conventional accelerator systems where the sensitivity of the beam dynamics and impacts of nonlinear beam collective effects is lower. For novel acceleration schemes such as plasma-based acceleration, the computational burden and level of complexity of the physics effects is even more substantial.

The computation burden of detailed accelerator physics simulations has numerous impacts that suggest where future improvements can be made: (1) it drives a need for more sample-efficient and high-dimensional optimization methods, (2) it limits in practice the extent of detailed analyses, such as comprehensive treatment of uncertainties due to component misalignment and noise, or inclusion of the majority of available variables in optimization, and (3) it limits the extent to which simulations are used online with operational accelerators, including the extent to which physics simulations are updated to match the as-built accelerator system. Many of these challenges are starting to be addressed with advanced computational methods, including the use of machine learning and differentiable simulators.

In addition, at present, there is a need for truly "end-to-end" accelerator system modeling that enables more comprehensive design and control. Looking to the future, this should include co-design of the accelerator and its controls along with the experimental equipment (such as particle detectors) and their associated analyses. New computational techniques for accelerator simulations and optimization leveraging differentiable programming, coupled with corresponding developments in particle detector design and physics analysis, provide promising avenues toward bringing this type of integrated approach to fruition. Much emphasis in recent years has been on machine learning applications that can directly aid accelerator operation (*e.g.* see Ref. [64] for some applications to the LHC), but substantial opportunities also exist for design optimization. In addition to the brief notes offered below, more detail on accelerator modeling challenges and emerging opportunities is given in Ref. [202].

Recently, a variety of computational techniques have begun to open up new capabilities in accelerator simulation and design. Many of these techniques are in their infancy with respect to application to accelerators and represent a significant opportunity for future development.

ML-based optimization algorithms have demonstrated increases in sample-efficiency and ability to extend to higher dimensionality, resulting in higher-quality solutions and reduced computational burden. For example, multi-objective Bayesian optimization [203] and Bayesian parameter space exploration [204] enable more sample-efficient optimization and characterization of accelerators (see, *e.g.,* Fig. 7), which is important both in design optimization and online optimization. These techniques can be used even when no previous data is available, making them appealing for commissioning of new systems. They can also be combined with learning where constraint violations on output parameters are likely to occur, to prevent undesirable conditions (such as losing the beam online, or wasting computational resources on poor simulation runs) [204,205]. Bayesian optimization can also leverage expected correlations from physics models to help increase convergence speed and improve sample-efficiency in model learning [206,207]. Feed-forward corrections with ML system models also have been used to speed up optimization convergence, both online and in the context of experiment design [62,208,209].

ML-based surrogate models trained on physics simulations have been shown to increase execution speed of accelerators system models by orders of magnitude [62,210,211]. These fast-executing models can be used in conjunction with standard optimization and feedback algorithms to speed up design. They can also be used online to provide fast predictions of unobserved beam behavior (*i.e.* "virtual diagnostics") [209,212–215]. This is of great interest in many accelerator applications where diagnostics are invasive to the beam and cannot be used during downstream delivery. Recent lines of inquiry have also examined the use of Bayesian neural networks [216] and ensembling [217] to obtain uncertainty estimates from these models. Such uncertainty estimates are essential when making design or control decisions, and when using model predictions to inform physics analysis or convey information about the beam to downstream user experiments. On a more granular level, numerous individual simulation calculations involved in accelerator physics codes can also be sped up and made differentiable with machine learning. For example, calculation of the impact of coherent synchrotron radiation (CSR) on each particle in the beam at each simulation time step can be very computationally intensive; preliminary studies have shown that computation of the CSR wakefield can be significantly sped up using a neural network [218].

Differentiable physics simulations have not yet been explored extensively in accelerators. While automatic differentiation has a long history in accelerator physics [219], the major simulation codes do not support arbitrary computation of gradients. Making these codes differentiable will be a challenge, especially in instances where simpler analytic or transport matrix based representations do not suffice and more detailed simulations (such as those based on particle-in-cell calculations) are needed. However, some of the potential benefits of differentiable physics models have been explored with differentiable ML models that are trained on detailed physics simulations. The differentiability of ML models enables them to be readily calibrated to match measurements from operational accelerator systems, for example to increase prediction accuracy and identify sources of systematic errors (such as unknown initial beam distributions, *e.g.* in Ref. [63,220]). This is important for understanding beam dynamics effects in the observed system. The differentiability of ML models also enables them to be combined with other ML algorithms directly. For example, in Ref. [221] back-propagation through a forward ML model trained on simulations of an injector system was used to train a neural

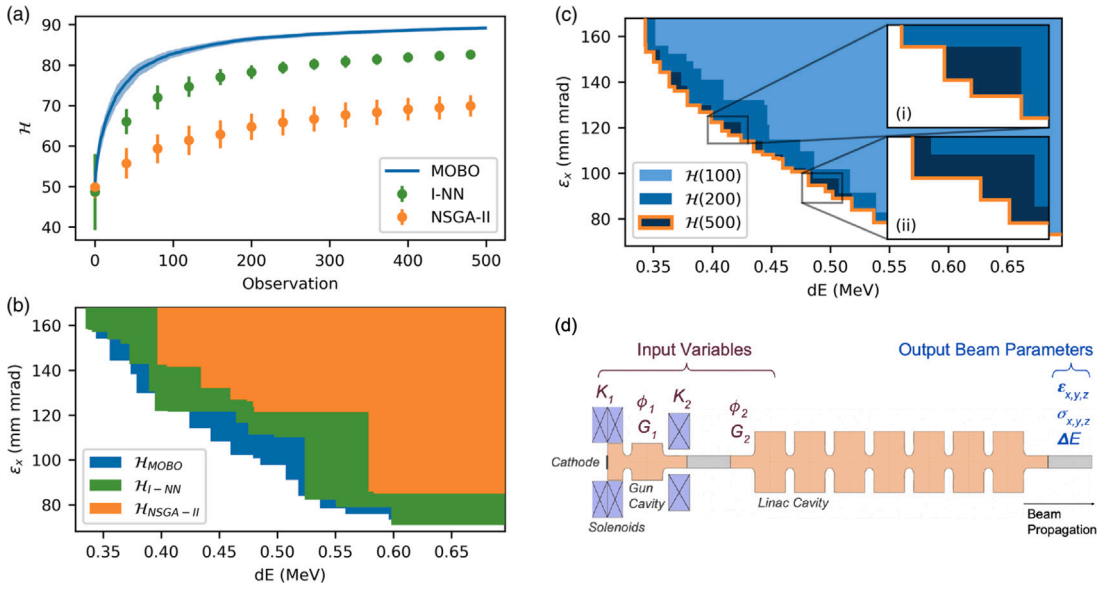


Fig. 7. Multi-objective Bayesian optimization of the Argonne Wakefield Accelerator (reproduced from Ref. [203]). (a) Multi-objective Bayesian optimization is more sample-efficient in converging to the Pareto-optimal front in a 7D objective space consisting of desired beam parameters (here denoted by the convergence in the hypervolume) than classical genetic algorithms [200] and neural network boosted genetic algorithms [62]. (b) The optimal front is outlined with higher resolution due to the uncertainty-aware sampling (an example for one of the 2D projections of the front is shown). (c) Running multi-objective Bayesian optimization for more iterations results in strategic refinement of the Pareto front resolution. (d) Schematic of the Argonne Wakefield Accelerator injector, along with input settings and output beam objectives used in this problem. For more detail, see Ref. [203].

network controller to quickly switch between beam energies while maintaining optimal beam size and divergence into an undulator. In that case, the forward model had to be updated occasionally with new data as the simulation entered new regions of the parameter space (where the forward model was no longer accurate). In contrast, having a differentiable physics simulation for this type of problem would enable one to avoid this intermediate step, and also would enable more flexibility in adjusting the problem to tackle other combinations of input and output variables. There are numerous accelerator control problems that could benefit from this type of approach. As another example, in Ref. [207] the Hessian of a neural network system model and physics simulation (numerically estimated) is used to easily incorporate expected correlations between input variables into a Gaussian process model, resulting in faster Bayesian optimization. Having differentiable physics simulations of accelerators would greatly enhance the ease and speed with which this type of approach could be applied to arbitrary accelerator optimization tasks.

Recent work has also highlighted the utility of using differentiable accelerator physics models by exploring this approach with simpler analytic representations. In Ref. [222] a physics-based model of hysteresis is made differentiable and used to describe accelerator focusing magnets. The differentiable physics model is then used to learn a magnet's hysteresis response based on measured data, enabling rapid solving of the hysterion density. This procedure can even be done indirectly by using, for example, beam size measurements; this capability is essential for applying this technique to existing accelerator beamlines where the individual magnetic field responses cannot be directly measured. It is then shown that this approach can be combined directly with Bayesian optimization for improved precision in optimization of an accelerator quadrupole triplet that exhibits hysteresis. Hysteresis is a major concern for accelerator optimization in practice, especially for high-energy systems, such as final focus systems for future colliders. Similarly, many high-level accelerator beam dynamics problems can be modeled in bulk analytically with transport matrices and Taylor matrices. These idealized representations can be cast into a differentiable form and used with gradient-based optimization of free parameters to make the model more closely match measured data, as was done in Ref. [223] for a Taylor map representation of beam dynamics in a ring.

Looking forward, differentiable physics simulations would help enable (1) tighter integration of physics simulations of accelerators with ML-based optimization routines, (2) gradient-based calibrations of predictions and uncertainty estimates when comparing simulations to measured data, (3) identification of unknown input parameters and sources of error, and (4) fine gradient-based optimization of beam phase space distributions at the particle level. Having differentiability for simple transport matrix-based accelerator codes and codes used for detailed predictions of beam phase space that include nonlinear collective effects (often requiring expensive particle-in-cell computations) would open up a host of new capabilities in this area. This could substantially benefit higher-precision modeling, characterization, and tuning of accelerators. Linking these modeling and optimization tasks directly with end user experiments (such as particle detectors) would further enhance the overall precision with which we can design, characterize, and control the entire end-to-end experiment.

Optimization and simulation infrastructure for accelerators and associated experiments. Finally, analogous to efforts for particle detector experiments, there is much work in the accelerator physics community to make software tools and standards that allow interoperability as ease-of-use when running simulations for different types of accelerator sub-systems, and exchanging data between them. This is essential for running end-to-end simulations and optimizations, as different parts of accelerators typically are simulated with different codes depending on the relevant physics effects for each section. Examples include frameworks such as LUME [224] to run different simulations and stitch them together, and Xopt [225] to drive high-level optimization and save datasets. This also includes standards for describing the beam distribution (such as OpenPMD [226–228]) and other aspects of the simulation and data sets (such as those used in LUME [224]). Efforts to develop standards for exchanging information with user experiments is also underway, for example in doing end-to-end simulations of accelerator-based light source user experiments. This provides a solid foundation for beginning to integrate accelerator simulations with particle physics detector simulations and analyses.

4.1.2. Calorimeter optimization

Calorimeters are a crucial component for most detectors at modern colliders. Their tasks include identifying and measuring the energy of photons and neutral hadrons, recording energetic hadronic jets, and contributing to the identification of electrons, muons, and charged hadrons. To fulfill these many tasks while keeping costs reasonable, the calorimeter construction requires good and thoughtful balancing with other components of the detector.

In practice, the design of a calorimeter depends on the choice among many options, which affect drastically both the physics performance of the instrument and its overall cost. Such options include:

- granularity of the detector;
- materials for absorber and active component;
- mechanical construction of absorber and active component (for sampling designs);

In the case of detectors based on light-producing effects, the following are also important factors:

- light collection and transport techniques;
- photo-detecting techniques;
- photodetector signal acquisition and processing.

The performance of the calorimeter may usually be described in terms of its:

- energy resolution;
- spatial resolution;
- time resolution;
- readout time;
- sustainability to in-time and out-of-time backgrounds;
- radiation hardness of components and related issues (aging, backgrounds, etc.).

In the following, we present a calorimeter optimization task for the upgrade [229] of the electromagnetic calorimeter [230] of the LHCb detector at the LHC [231]. The upgraded calorimeter is expected to operate in the challenging conditions of LHC's high luminosity Run 5 and beyond [232].

Problem statement and current optimization approach. The ultimate goal of the optimization of a calorimeter is to achieve the best performance to fulfill the physics program of the experiments, while fitting within the available budget. Taking into account the different tasks for the detector as mentioned above, the list of requirements is a trade-off between different properties, including:

- radiation hardness to sustain the expected lifetime span;
- energy and spatial resolution for good photon reconstruction and electron identification;
- high granularity to facilitate better precision, both spatial and in energy, which in turn improves reconstruction algorithms;
- good timing resolution to facilitate pileup suppression in high-occupancy areas as well as better matching of separate signal components.

Thus, the optimization problem statement is to find a technology for the construction of calorimeter modules, *e.g.* homogeneous, Shashlik [233], or SpaCal [234,235] (see Ref. [236] for a review on calorimetry technologies for particle physics), and to choose the construction details for the chosen technology, such as materials, granularity, geometries, etc.

To evaluate the physics performance of the different possible configurations, a detailed GEANT4-driven [102–104] simulation of the detector modules is needed. Thus, the first task in an optimization pipeline consists in evaluating necessary target low-level performance metrics like energy, spatial, and timing resolutions from physics-based first principles, and validating with simulation if a particular configuration fulfills the requirements. The challenge of this approach is that, although the individual detector channel response may be simulated, achieving even a low-level performance requires the development of a reconstruction algorithm tuned to the particular configuration under study. Moreover, this approach decouples the local optimization of the calorimeter low-level performance metrics from the global optimization of the physics performance of the entire detector. To include the ultimate physics performance into consideration, a more comprehensive optimization loop is necessary [237].

The typical workflow for the optimization of calorimeter components is sketched in Fig. 8 and proceeds as follows:

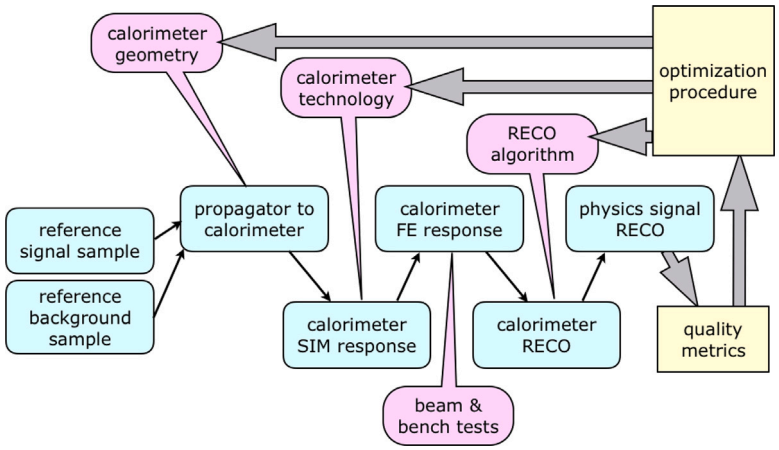


Fig. 8. A general pipeline for the calorimeter optimization includes several steps. Blue blocks indicate data processing pipeline steps; pink bubbles represent configurations and conditions for pipeline steps; yellow blocks close the optimization loop. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- Selected event samples, both signal and background, are used to initiate an optimization cycle for comparing the performance in terms of signal recovery and background suppression;
- the calorimeter is usually installed at some separation from the interaction point and downstream of other detection elements, so a propagation of events from their origin to the calorimeter is necessary. This step is dependent on the properties of the elements of the detector between origin and calorimeter. Additionally, if the calorimeter detector has a non-homogeneous configuration, the details of the global geometry are to be accounted for in this step for the optimization to be based on physics performance quality metrics;
- the construction technology used for the individual calorimeter modules is a central point for the detector R&D. To evaluate the impact of a choice of construction technology, we need to simulate its effect on observable event characteristics. This is done using response simulation models, typically based on GEANT4. The details of the calorimeter technology used to drive such simulation;
- the behavior of the front-end electronics is another important contribution to the physics quality of the detector. Although the properties driving the behavior are hard to simulate, suitable data samples may be obtained from beam or bench tests;
- a reconstruction algorithm is absolutely necessary to evaluate the quality of converting the detector response into physics objects;
- physics quality metrics may be calculated using reconstructed objects and it can be used as a target function for the optimization procedure;
- all design aspects of the calorimeter may be optimized: the details of the calorimeter technology, the geometrical layout, and the possible reconstruction algorithms.

Such an optimization cycle, built on top of an event-processing pipeline, makes it possible to obtain physics-motivated optimal values for the detector parameters.

To evaluate the physics performance of a particular configuration for a possible future calorimeter detector, one needs to run the optimization cycle described above. A good fine-tuning of the individual blocks is important to properly propagate the properties of the configuration under study to the ultimate physics performance. For the regular operation of a stable detector, these blocks are carefully tuned based on the actual detector configuration. In contrast, for the R&D of a new detector, many different possible configurations are studied simultaneously during the optimization stage. Nevertheless, reasonable representations of the simulation and reconstruction steps, which are tuned for each of the configurations studied, are necessary for inferring consistent conclusions about the physics performance of these configurations. This is a time-consuming task, if done manually. Fortunately, these studies use well-labeled data sets either from MC simulation or from test beam measurements. Surrogate models may therefore be built and trained on labeled data using regular ML approaches. This makes it possible to speed up model building for different pipeline steps. Importantly, such training may be automated and requires minor expert supervision.

The big slowdown factor for running an optimization cycle is the necessity of fine-tuning reconstruction algorithms for every new calorimeter technology and geometry configuration. ML may help to tune the reconstruction automatically. Indeed, as soon as a sample of calorimeter responses is available, the corresponding regressor may be trained to extract physics information from the raw response.

Fig. 9 demonstrates the quality of the spatial reconstruction of the calorimeter cluster for the case of the LHCb 4 cm and 12 cm modules [238]. An automatically trained ML model (based on XGBoost [239], in this case), agnostic to particular calorimeter details, produces a slightly better performance than the manually selected parametric model. Importantly, the automatically trained generic regressor provides a performance comparable with that of the manually tuned one. This justifies the use of this surrogate regressor

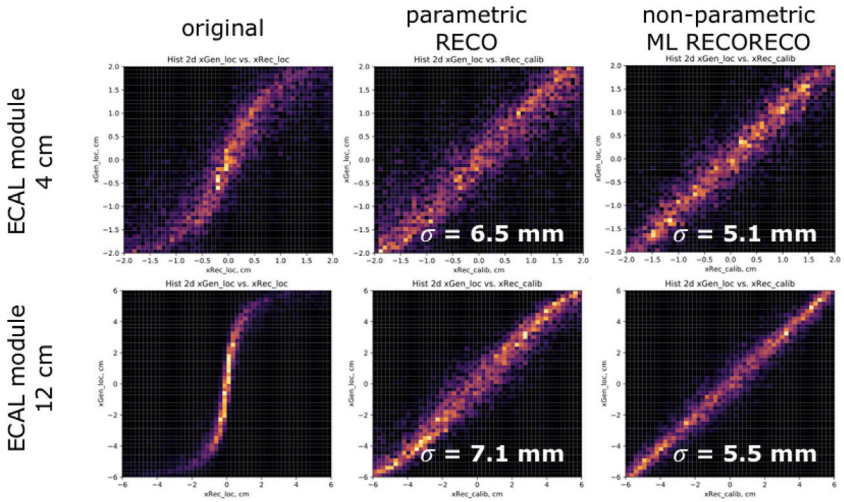


Fig. 9. ML-based reconstruction of the calorimeter cluster position provides spatial resolution similar to the customized reconstruction procedure, but without *a priori* knowledge about the particular spatial properties of the calorimeter under study. Left — correlation between cluster center and the true track position; middle — correlation corrected using parameterized correction; right — correlation using ML trained regressor.

in the optimization cycle in place of a well-tuned reconstruction algorithm for extracting physics observables from the calorimeter response.

Figures of merit. The local figures of merit (FOM) for the calorimeter optimization are the characteristics mentioned above: energy, spatial and timing resolutions, sustainability to high pile-up and huge radiation doses, etc. However, the ultimate figure of merit for the optimization process is the physics performance achievable using a given configuration of the detector. Moreover, in typical use cases the FOM is actually not a single number, but rather a dependency of the physics performance on the cost of the configuration of the detector under study.

Examples for optimizations. To give an idea of the optimization procedure, let us consider the future calorimeter of the LHCb experiment. Different regions of the detector are very different in terms of requirements to the precision of the signal measurement and background conditions. We can use different technologies for calorimeter modules (Shahlik, SpaCal), different materials (lead, tungsten) with different granularity in different areas to optimize the overall performance.

Let us consider one of the target physics processes for the optimization, namely the decay $B_s \to J/\psi(\mu^+\mu^-)\pi^0(\gamma\gamma)$ with emphasis on the B_s signal reconstruction performance, which is driven by the photon reconstruction in the calorimeter. Using simulated signal and background events, signal photons may be propagated directly from the π^0 decay point to the calorimeter, and the expected calorimeter response for a given module technology and configuration may be extracted. This response needs to be merged with the estimated contributions of in-time and out-of-time pileup collisions.

To evaluate a realistic reconstruction performance for photon energies, position, and timing, we trained ML-based regressors similar to those presented in Fig. 9. Using reconstructed photon parameters we then evaluated the ultimate performance for the physics signal—estimated, in this particular case, as the significance of the reconstructed B_s—for different physics selections. Fig. 10 shows the significance curves obtained for different choices of the physics reconstruction parameters, from which we can select the best B_s selection algorithm for a given calorimeter configuration, resulting in an optimal physics performance for the configuration under scrutiny.

From another perspective, for each calorimeter configuration, one can evaluate the monetary cost of building such a detector. In our case, the overall detector cost is mostly driven by the total number of readout channels. Thus, the physics performance obtained above for each configuration is associated with the corresponding cost of building the detector using the configuration under exam. Fig. 11 illustrates the benefits brought to the physics performance by the number of readout channels used for different configuration of calorimeter cells, their granularity, and readout schemes. This kind of information vastly simplifies the problem of choosing the optimal balance between physics performance of the designed detector and the cost one has to pay to achieve that performance.

4.1.3. Hybrid calorimetry for future particle colliders

High granularity has recently become a new paradigm in calorimeters for particle colliders, following the realization that the hadronic decay of highly boosted heavy particles can be distinguished from backgrounds through detailed studies of jet substructure. Another advantage of granular detection of hadronic showers lies in the possibility of increasing the performance of particle-flow algorithms, which exploit the coupling of tracking information to reconstruct all the primary constituents of hadronic jets, thereby improving the overall energy resolution of the detector as well as its capability to help in discriminating the originating partons. In parallel with the desirability of increasingly granular designs, technological advancements have been progressively reducing the

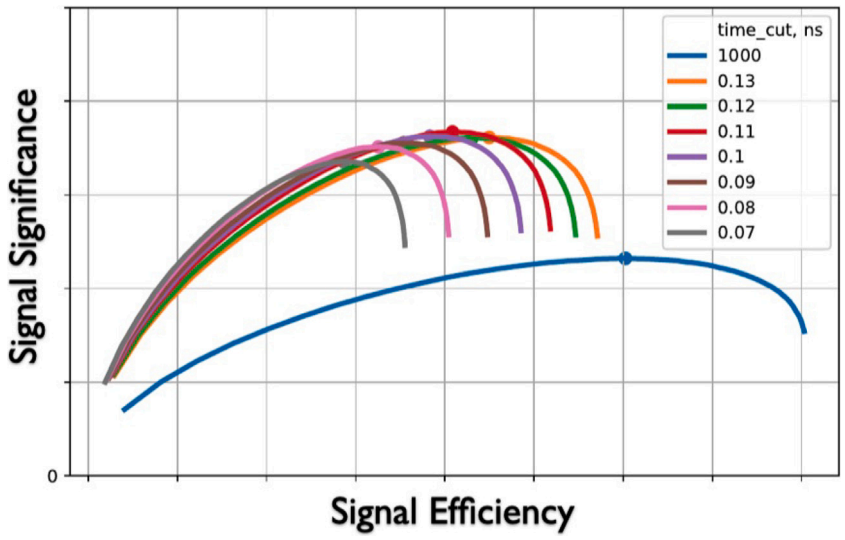


Fig. 10. Illustration of the physics signal significance for different efficiencies of the signal selections, evaluated using dynamically trained, ML-based calorimeter reconstruction algorithms. Different curves correspond to different timing discrimination windows in this example.

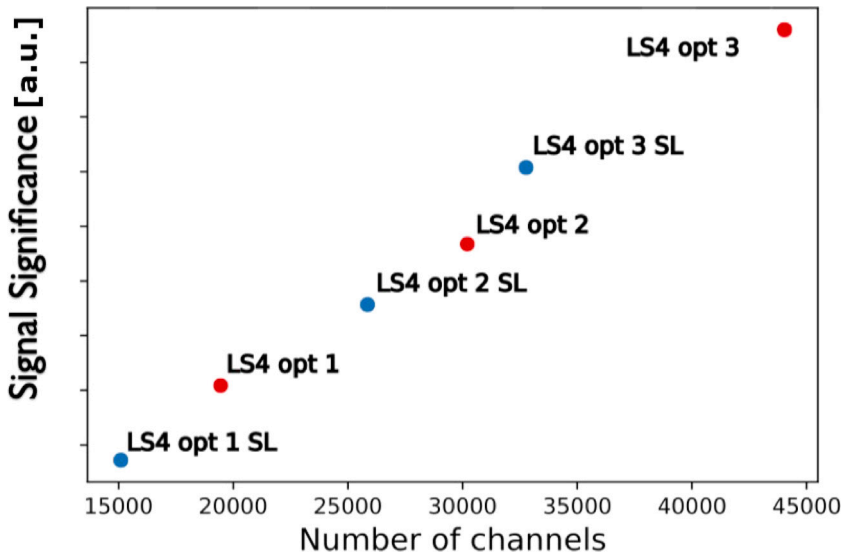


Fig. 11. Dependence of the optimized significance (for a given reference signal of interest) for different calorimeter configurations on the total number of readout channels used in those configurations. This information helps one make an educated decision about the optimal detector configuration which provides the best balance between ultimate physics performance and construction costs. The vertical axis is in arbitrary units.

related construction costs, and in some case enabled solutions previously not feasible, such as with the now possible 3-D printing of scintillating materials into arbitrarily complex shapes.

The above trend is obviously leading toward a rethinking of the dichotomic split between a tracking detector, meant to measure charged particles without perturbing their path or modifying their identity, and a calorimeter meant to destroy both charged and neutral particles and deduce their energy from the resulting shower of secondaries. Machine learning reconstruction may help foster this evolution, if it can show that nuclear interactions in the tracker material can actually be exploited rather than suffered. In fact, different hadrons have different cross section for nuclear interactions, as well as distinguishable phenomenology. If those differences can be leveraged to improve particle identification, the concept of tracker and calorimeter may become less distinct, potentially improving the overall performance of event reconstruction. In addition, today, our understanding of the fragmentation properties of energetic partons is mature enough to offer itself as prior information that can be incorporated in the reconstruction task of a particle flow algorithm. This may seem hazardous, as it brings in additional modeling uncertainties, but today we routinely exploit

our understanding of hadronic showers in the extraction of information from calorimeter signals, so exploiting also our knowledge of parton showers should be considered an obvious next step.

Finally, it is also worth mentioning that the magnetic measurement of muon momenta above a few TeV becomes unfeasible within a tracker; a future 100-TeV-class particle collider would thus lose the ability to exploit the excellent information power of very energetic muons for the hunt of heavy resonances decaying to muon pairs, if it did not offer alternative ways to measure the energy of those particles. Granular calorimetry has been shown to provide a complementary way of inferring the energy of muons by exploitation of the pattern of soft photon radiation produced along the path of those particles in dense materials [240].

The above ideas can only be tested within a general study of hybridization of tracking and calorimetry, which is a long-term but compelling plan. One of the ingredients of such a study is the fast simulation of calorimeter showers, which we discuss briefly below.

4.1.4. Generative models for calorimeter showers

The optimization of a calorimeter for a future particle collider requires a reliable description of particle showers from electromagnetic and hadronic cascades in the material of the calorimeter. If the detector layout is to be optimized based on the calorimeter shower structure, the corresponding differentiable pipeline must be interfaced with a differentiable surrogate of the calorimeter shower simulation.

To achieve a differentiable description of calorimeter showers, all three types of generative networks mentioned in Section 3.2.2 (GANs, VAEs, and Normalizing Flows) can be used. There are, however, differences in sampling time and quality that depend on the used architecture. Given this trade off, current applications of deep generative models at the LHC and beyond usually focus on a higher sampling speed, since a suboptimal performance in shower quality tends to matter less in downstream tasks [126]. For the purpose of optimizing a calorimeter, however, a better sampling quality, *i.e.* having more realistic showers, is more important. Therefore, we will be focusing on the generative model with the best sampling quality. However, as already stated in Section 3.2.2, the quantitative evaluation of the quality of generative models is a difficult task [130]. This is especially true if models based on different architectures (VAEs, GANs, and normalizing flows) are compared to each other. One metric for comparison is given by a binary classifier to distinguish "real" (meaning based on GEANT4 training data) from "fake" (meaning generated by the generative model) samples [241]. By using the Neyman–Pearson lemma, we can argue that $p_{\text{real}} = p_{\text{fake}}$ if a powerful classifier cannot distinguish the two sets. This test can therefore be seen as the "ultimate" test, as it considers not only all voxels, but also their correlations [159].

One of the first applications of deep generative modeling to calorimeter simulation was the CaloGAN [132,133]. There, a three-layer deep, simplified version of the ATLAS LAr detector was considered. The three layers were segmented into 288 + 144 + 72 voxels respectively, yielding a feature space of 504 dimensions. Showers initiated by e^+, γ, and π^+, shot perpendicularly to the detector surface and of uniform incident energy in the $[1, 100]$ GeV range, could be generated by CaloGAN at a factor 20,000 faster than with GEANT4. Histograms of high-level features of these showers were close to their GEANT4 counterpart. However, it was shown in Ref. [159] that these showers still failed the classifier test and could be separated from the GEANT4 produced showers with almost 100% accuracy. Samples of more advanced setups, like the Bib-AE [109], were also shown to be separable by classifiers, see Ref. [148].

The first generative model capable of generating samples that would confuse a classifier was based on normalizing flows [159, 160]. This approach, called CaloFlow, used the same detector geometry as the CaloGAN. The authors used a multistep approach to generate high-quality samples. The first step uses a small NF to learn the probability $p_1(E_i|E_{\text{inc}})$, the distribution of the deposited energy into the three layers, E_i, conditioned on the incident energy E_{inc}. The second step uses a larger NF to learn the normalized energy depositions into all 504 voxels, I, conditioned on the layer and incident energy: $p_2(I|E_i, E_{\text{inc}})$. The normalization of energy depositions in each calorimeter layer helped to capture the different energy scales of the E_i, coming from the large size of layer 1 compared to layers 0 and 2. After generation, showers in each calorimeter layer were renormalized to have the correct energies E_i that were used in generation. Such a step is necessary since generative models, even when trained on normalized showers, generate samples that are not perfectly normalized. There are two different architectural choices for the NF in the second step that were explored in Refs. [159,160]. A MAF-based flow (see Section 3.2.2) could be trained with the log-likelihood objective, resulting in a stable training without artifacts in generation and an optimal model selection based on the log-likelihood of a held-out test set. While this MAF was a factor 500 faster than GEANT4, it was still rather slow compared to other deep generative models. An IAF-based flow (see Section 3.2.2) improved the sampling speed by a further factor 500, making CaloFlow as fast as CaloGAN. Due to memory constraints, such an IAF cannot be trained using the log-likelihood anymore. Instead, a method called probability density distillation, originally developed for speech synthesis in Ref. [242], had to be used.

Fig. 12 shows histograms comparing the energy depositions in the three calorimeter layers of showers from GEANT4, CaloGAN, and CaloFlow. Fig. 13 shows the time needed to generate showers as a function of the number of requested showers. All deep generative surrogates outperform GEANT4 for the shower energy considered in the study (the speed-up could be smaller at lower energies).

4.1.5. Electromagnetic calorimeter of a Muon Collider experiment

The next generation of experiments in particle physics finds its common goal in expanding the phase space of events by increasing the energy scale of collisions beyond the TeV. Within this context a new international collaboration has been forming to study the design, challenges and outreach potential of a Muon Collider, which could be envisioned in Europe by the late 2040s [243].

A Muon Collider comes with three main advantages, due to the properties of muons themselves: (I) a higher subprocess center of mass energy for each collision with respect to those produced by hadron colliders of same beam energy, since all the energy of

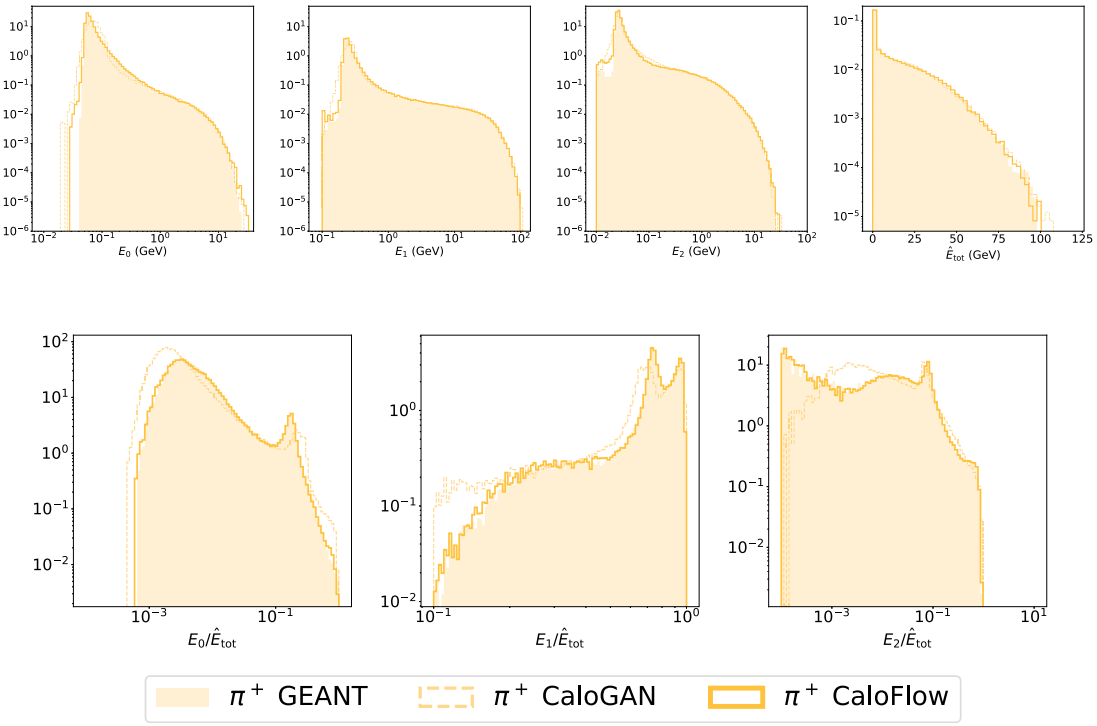


Fig. 12. Distributions of energies in the three calorimeter layers and total deposited energy (top) and ratio of layer energies to total deposited energy (bottom) for incident π^+ particles, comparing GEANT4 to CaloGAN [132,133] to CaloFlow [159,160].

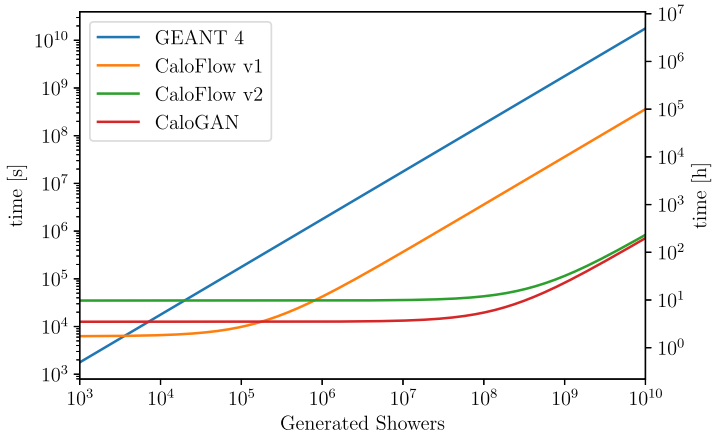


Fig. 13. Comparison of shower generation times of GEANT4, CaloGAN [132,133], CaloFlow v1 [159], and CaloFlow v2 [160].

projectiles is available for the hard subprocess; (II) few design limitations due to synchrotron radiation, allowing one to use today's accelerator technology for a circular lepton collider; (III) the highest luminosity per energy used, which translates into the highest energy efficiency for a collider experiment. It does however come with a set of challenges that need to be considered, especially due to muons not being stable particles: this generates a cloud of decay products that runs alongside the main beam line and interferes with detectors and instrumentation (Beam-Induced Background — BIB). The proposed design of the detector, largely borrowed from the ILC [244], already includes a double cone-shaped tungsten nozzle for shielding, which reduces low-energy backgrounds inside the detector volume (see Fig. 14).

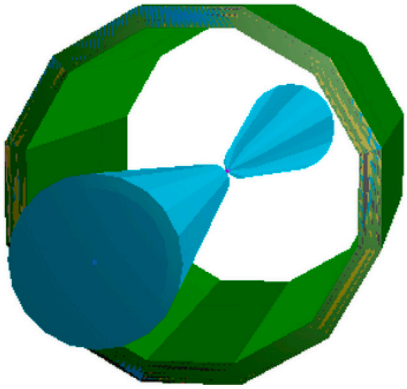


Fig. 14. Muon Collider detector — nozzle and Crilin barrel design.

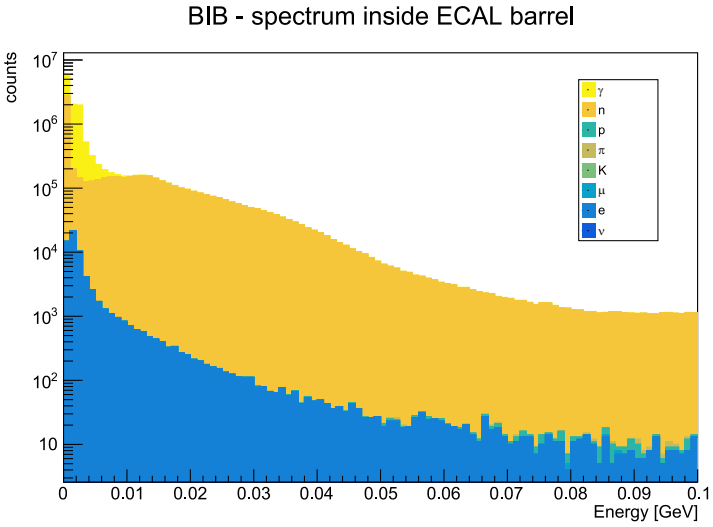


Fig. 15. BIB spectrum components inside the muon collider detector ECAL modules.

A new design for the electromagnetic calorimeter (ECAL) has been proposed, named Crilin (crystal calorimeter with longitudinal information) [245], which implements an array of PbF_2 crystals which reduce the cost of instrumentation without loss in resolution from the original design. This solution is still in development phase and no definitive design has been proposed yet, which offers a chance to perform an optimization study systematically and find the best possible setup to maximize the physics potential of the experiment when BIB effects are accounted for.

Fig. 15 shows the BIB broken into its main components, obtained with a MARS15 [246] simulation of its interaction with the nozzle. The main contributions come from photons and neutrons; for a preliminary study, we chose to focus on the former. We used GEANT4 to study how the bulk material reacts to BIB photons, with the goal of obtaining a continuous model for the detector response which may be used in a full differentiable model. This may be implemented in the geometric configuration of the detector, to get a toy model that can be dealt with for optimization studies. This toy model may also enable studies of event reconstruction using techniques such a Object Condensation [32].

Given the asymmetric properties of the large BIB flux, it appears advisable to construct a full model of the calorimeter and its performance in detecting interesting physics signatures. This may be accomplished with a pipeline which includes the signal and background modeling, their reconstruction, and a suitably defined loss function. Various parameters such as crystal granularity and resolution can be analyzed, as well as how timing information may be exploited for backgrounds reduction.

4.1.6. Optimization of the MUonE detector

The MUonE detector [247] is an experiment proposed to be constructed and operated at the CERN North area, where it would intercept an intense 150-GeV muon beam directly upstream of the COMPASS experiment. The experiment aims at precisely measuring the q^2-differential cross section for elastic muon–electron scattering, which would allow to estimate the next-to-leading order hadronic contribution to the scattering process. The estimated value of that quantity would directly constrain one of the

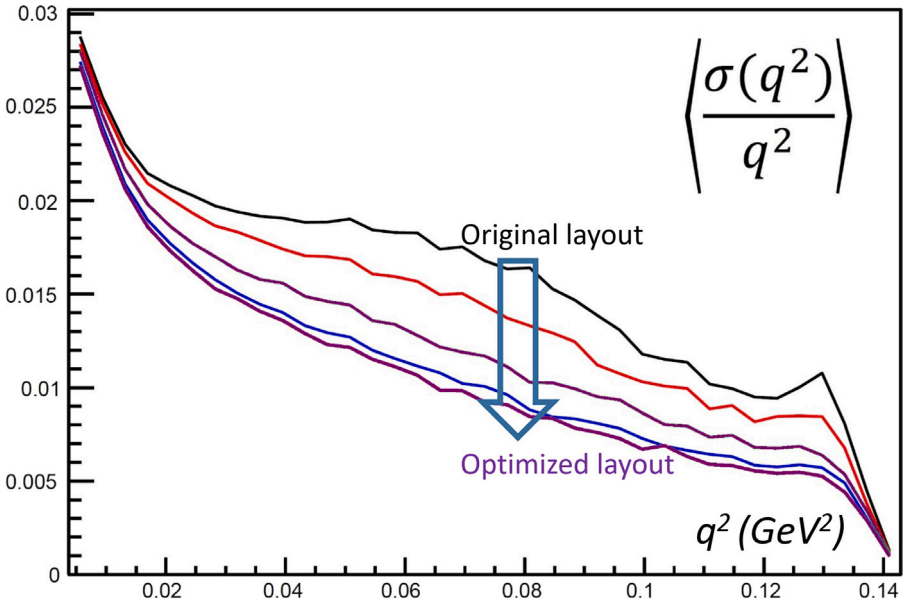


Fig. 16. Relative resolution in q^2 as a function of q^2 of muon–electron elastic scattering. The curves, colored from black to purple, show the improvements in resolution brought by separate optimization steps. The black curve corresponds to the original design, the yellow one to the result of a full optimization. A gain of a factor of 2 is achievable in relative resolution, which is the relevant figure of merit especially in the high-q^2 region.

leading sources of uncertainty in the determination of the anomalous magnetic moment of the muon,

$$a_\mu = \frac{g-2}{2}, \tag{20}$$

(where g is the gyromagnetic ratio of the muon), improving the precision of comparisons of theoretical predictions with the experimental determination of that quantity, currently underway at Fermilab. Due to a long-standing disagreement of theory and measurement of a_μ, which might originate from new physical processes contributing to the magnetic moment with quantum loop diagrams, the interest of improving the theoretical precision with the MUonE study is significant.

In its proposed setup, MUonE is composed of a set of 40 identical tracking stations, each 1-m long, followed by an electromagnetic calorimeter. The stations measure the tracks of incoming muon and outgoing muon and electron in a set of silicon strip planes, and include beryllium targets for the scattering reaction. Given the well-defined goal of the experiment, the simple layout of the detector, and the very straightforward reconstruction of the event kinematics from measured particle trajectories, MUonE lends itself quite well to an exercise in optimization that may consider the full detector geometry, including positioning of active and passive layers as well as detailed characteristics of the detection elements, in the maximization of an objective that fully describes the final goal of the experiment. In fact, given the fully constrained kinematics of elastic scattering reactions, the stochastic elements of the problem are confined to multiple scattering in the material of the tracker, charge collection processes in the silicon strips, the production of soft photons emitted in semi-elastic events, and the description of the detection of electron showers in the calorimeter. However, the details of the physics of the calorimeter, its performance and response may be neglected in a first study of the tracker optimization alone, since the calorimeter provides redundancy in the measurement and a disentanglement of the symmetrical scattering situations where muon and electron emerge with the same divergence from the incoming muon direction. The information may be encoded as a probabilistic function that describes the correct identification of the electron shower, and a roughly Gaussian density function describing the estimate of the electron energy.

More detail can be found in Ref. [59], where a simple grid-scan approach was followed, demonstrating that large gains are reachable (see Fig. 16). Although that brute-force procedure was found sufficient for the study of considered design parameters, a more refined approach (as advocated by this document) would allow a more complex and precise multidimensional optimization with smaller computational cost.

As already noted in Section 3.1, the creation of a differentiable model of the geometry of silicon strips and target layers is straightforward. The other elements of the problem are listed below:

- A sufficiently precise differentiable model of multiple scattering of electrons and muons in the tracker material. This is simple to produce.
- A differentiable model of the charge deposition in silicon strips by the three charged particles involved in the reactions. In a first approximation, this may be produced with a simple charge transport model as discussed in Ref. [59]; a more careful treatment may involve local generative surrogates.

- A model of the self-calibration procedures that enable the constraining of position uncertainty of detection and passive layers, as described in the above-mentioned source.
- A model of the information-extraction procedures that allow to derive the quantity of interest and its uncertainty $\delta\alpha_{had} \pm \sigma_{\delta\alpha_{had}}$ from a given amount of identified elastic collision events and their estimated q^2.

At the time of writing this document, the MUonE detector geometry is already defined, and construction of layers and stations has begun.[7] The above discussion is therefore mainly of academic value, yet this simple use case is illustrative of a number of possible similar problems involving silicon tracking detectors, to which many of the points made are relevant.

4.1.7. Searches for milli-charged particles

It has always been intriguing why elementary particles in the standard model have electric charges that are a fraction $(\pm 1, \pm 1/3, \pm 2/3)$ of the charge of the electron. For instance, one can add a new gauge field A'_μ that couples to a new fermion χ with coupling e' and kinetic mixing (with coefficient κ) with the hypercharge field B_μ. Then, by choosing a basis that will eliminate this kinetic mixing (A'_μ and B_μ) the standard model will lead to an electric charge for the new fermion χ that can be tuned by tuning $\kappa e'$. Having small values of $\kappa e'$ could be interpreted as a centicharged or even a millicharged particle.

Searches for millicharged particles range from astrophysical observations to fixed targets experiments (SLAC-mQ [248]), to accelerator facilities searches like the LHC. In astrophysical proposals, millicharged particles can be produced either in the atmosphere or during propagation through the earth. Given the weak interaction of the millicharged particles with matter, neutrino detectors like Super-Kamiokande were proposed as typical detector. The common denominator of most proposed neutrino-dedicated experiment is the liquid detector structure. In Ref. [249] the data from ArgoNeuT (a liquid Argon Time projection chamber) experiment at Fermilab was reanalyzed to check for mCP. The authors look for mCP in events triggered with data acquisition set in coincidence with the NuMI beam spill signal. The majority of events do not signal neutrino propagation, due to the low neutrino interaction cross section and the spatial limit of the detector. Such "neutrino-empty" signals are explored for mCP presence. In a recent mCP detection proposal using a neutrino-purposed experiment [250], the Super-Kamiokande (50 kton liquid Cherenkov detector) and the Jiangmen Underground Neutrino Observatory (JUNO, a 20 kton liquid scintillator detector) were proposed to detect mCP coming from/through the atmosphere. A dedicated mCP detector, MilliQan [251] has been installed in a CMS drainage gallery. The initial detector design was a simple scintillator bars staked on top of each other and oriented toward the interaction point of CMS. GEANT4 simulations were made to understand the behavior of the detector and the layout of the different scintillator layers. Several lessons were learnt from the installed demonstrator in 2018. A new design has been proposed and is under construction for LHC Run 3 [252]. Besides the upgraded bar detector design shown in Fig. 17 (left), a new slab detector has been added to increase angular acceptance, as illustrated in Fig. 17 (right). Muons constitute a major background for mCP detection. The scale of this kind of detectors is sufficiently contained, and the R&D turnaround is sufficiently fast, that similar considerations may apply as in muon tomography (Section 4.3), and an adaptation of TomOpt (Section 4.3.3) is being considered for further optimization of the detector.

4.1.8. Error analysis of Monte Carlo data in lattice QCD

Lattice QCD is a computational framework based on discretizing space and time on a hypercubic lattice of spacing a. This distance plays the role of a cutoff, providing a regularization of the field theory. Although not very useful from a purely analytic point of view, the appeal of this technique comes from the fact that such discretized versions of QCD can be *simulated* on a computer using Monte Carlo methods, even in the notoriously difficult non-perturbative regime of QCD. In the last years, Lattice QCD has matured significantly and is able to determine many key quantities for HEP phenomenology (see Ref. [253] for a summary).

Here we will focus on the applications of automatic differentiation to data analysis in lattice QCD along the lines described in Ref. [254]. While this topic straggles away from the strict boundaries of detector design optimization, we believe that a discussion the employed tools may be useful in the context discussed in this publication.

Data analysis in lattice QCD consists in processing the *input*: simulation data in the form of Monte Carlo averages over ensembles, experimental inputs, etc. We aim at producing some prediction (*i.e.* the *output* of the analysis), together with its uncertainty (see Fig. 18). It is clear that the prediction is a function of the *input*, but in general this function is very complicated: it involves several nonlinear fits, interpolations, and other iterative processes like root finding. The challenges are the following:

Correlations between inputs Since producing Monte Carlo ensembles is numerically very expensive, several quantities are measured on each ensemble. This produces statistical correlations between different *inputs* that affect the error estimates.

Auto-correlations of the data Because of the very nature of Markov Chain Monte Carlo simulations, subsequent measurements of a quantity are not statistically independent.

We use the Γ-method [255,256] to quantify the effect of these auto-correlations on the error estimates. It remains to be decided how uncertainties are propagated from the inputs to the predictions taking into account correctly the different correlations.

[7] Of the two solutions advocated in Ref. [59] to improve single-hit resolutions, the collaboration chose the tilting of sensors as in the right panel of Fig. 4, which at the price of some complication in the geometry allows reusing the construction jigs for the assembling of double-sided sensors, which do not allow for a staggering of strips on the two sides.

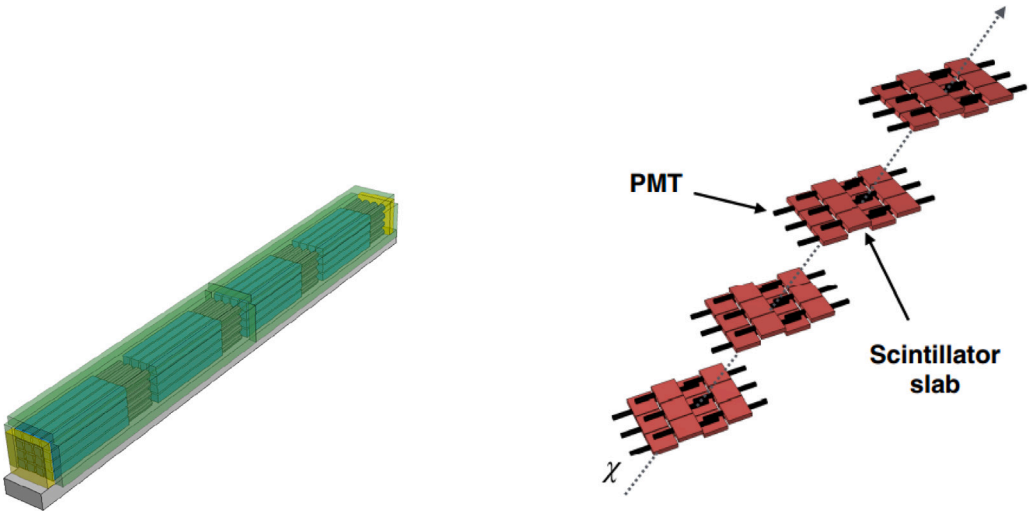


Fig. 17. Left: Bar detector for the MilliQan experiment, consisting of four layers each is composed of 4 × 4 rectangular scintillator bars 5 × 5 × 60 cm^3 each with PMT on its end. Right: slab detector for the MilliQan upgraded detector, consisting of four layers, each endowed with four slabs of dimensions 5 × 5 × 60 cm^3 (right).
Source: Reproduced from Ref. [252].

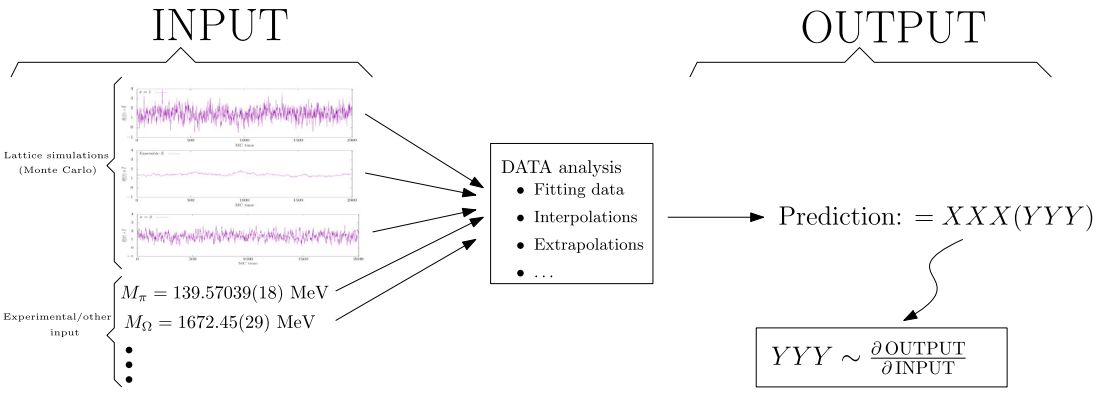


Fig. 18. In lattice QCD several inputs (both experimental and the result of large scale Monte Carlo simulations) are used to determine several quantities of interest. In Ref. [254] it is proposed to determine the uncertainties by computing the derivatives of the results with respect to the inputs using automatic differentiation.

Our prediction F is a generic function of the inputs $F \equiv f(A_i^\alpha)$. The index i labels the input and the index α labels the source of uncertainty. Correlations between different inputs arise from the fact that several input (values of the index i) can share the same source of uncertainty (value of the index α). The final value of the prediction is estimated via

$$\bar{F} = f(\bar{a}_i^\alpha)\,, \tag{21}$$

where $\bar{a}_i^\alpha$ are the central values of the inputs, or Monte Carlo averages

$$\bar{a}_i^\alpha = \frac{1}{N}\sum_{t=1}^{N_{\rm MC}^\alpha} a_i^\alpha(t) \tag{22}$$

in case that the input consists on $N_{\rm MC}^\alpha$ Monte Carlo measurements. In order to estimate the error in $\bar{F}$ we use linear error propagation

$$f(A_i^\alpha + \epsilon_i^\alpha) = F + f_i^\alpha \epsilon_i^\alpha + \mathcal{O}(\epsilon_i^2)\,. \tag{23}$$

where the derivatives $f_i^\alpha = \partial f/\partial A_i^\alpha$ are in practice evaluated at $\bar{a}_i^\alpha$

$$\bar{f}_i^\alpha = \frac{\partial f}{\partial A_i^\alpha}\Big|_{\bar{a}_i^\alpha}\,. \tag{24}$$

The error in our prediction is estimated using the *per-ensemble* auto-correlation function

$$\Gamma_F^{\alpha}(t) = \sum_{ij} \bar{f}_i^{\alpha} \bar{f}_j^{\alpha} \Gamma_{ij}^{\alpha\alpha}(t), \tag{25}$$

with $\Gamma_{ij}^{\alpha\alpha}$ being the product of the uncertainties of the inputs i and j, or the auto-correlation function in case of Monte Carlo input (see Ref. [254] for more details).

The proposal [254] consists in using automatic differentiation to compute the derivatives appearing in Eq. (24). A key issue is how to efficiently compute these derivatives when the function f consists on an iterative process. The typical case is fitting the data to some model, that we will examine in detail. The optimal value of the fit parameters $\bar{p}_i\,(i = 1, \ldots, N_{\text{parm}})$ are given by the minimum of the function

$$\chi^2(p_i; d_a), \qquad p_i\,(i = 1, \ldots, N_{\text{parm}}), \quad d_a\,(a = 1, \ldots, N_{\text{data}}), \tag{26}$$

where d_a is the data (*i.e.* input) that is fitted. Error propagation requires the derivatives of the fit parameters with respect to the data with the condition that the χ^2 function has to stay at the minimum. These derivatives can be exactly computed by the following equation [254]:

$$\frac{\delta p_i}{\delta d_a} = -\sum_{j=1}^{N_{\text{parm}}} (H^{-1})_{ij} \partial_j \partial_a \chi^2 \Big|_{(\bar{p}_i; \bar{d}_a)}. \tag{27}$$

where $H_{ij} = \partial_j \partial_i \chi^2 \Big|_{(\bar{p}_i; \bar{d}_a)}$ is the Hessian of the χ^2 at the minimum.

This Hessian can also be computed using automatic differentiation techniques. The advantage of this approach is that one can rely on external efficient libraries for the minimization of the χ^2 function (*i.e.* to determine the central values of the fit parameters). Error propagation is performed *once the minimum is found*, by explicitly computing the Hessian of the χ^2 function. Similar expressions can be easily obtained for any problem that can be formulated as a root-finding problem (the minimization of the χ^2 is nothing but finding the root of the gradient).

We offer a few concluding comments for this section. First, using automatic differentiation for error analysis results in *robust* error estimates: as long as the central values of the output are correctly computed, the exact nature of automatic differentiation guarantees that errors will be correctly propagated. Second, automatic differentiation techniques produce computationally efficient analysis codes. They are significantly cheaper than resampling techniques (see Ref. [254] for an explicit comparison). Finally, some of the ideas discussed here, such as how to deal with Monte Carlo data, and how to determine efficiently derivatives of complicated iterative processes, can potentially be applied to other problems.

Numerical implementations of these analysis techniques are freely available:

Fortran https://gitlab.ift.uam-csic.es/alberto/aderrors

Julia https://gitlab.ift.uam-csic.es/alberto/aderrors.jl

Python https://github.com/fjosw/pyerrors

4.2. Astro-particle physics and neutrino experiments

In this section, we sketch a common optimization problem which the astrophysics field will face in the near future. Multi-messenger observations of transient astrophysical events [257,258] aim to involve all available observatories. This common effort, started with transients coming from gamma ray bursts [259], was expanded by announcements from GW detectors. Historically the Gamma-ray Coordination Network, now also known as Transient Astronomy Network (GCN/TAN), collects and redistributes low-latency circulars about astronomical transient events such as: supernovae, gamma-ray bursts, kilonovae or binary mergers. Each event is carefully characterized by the nature of the event, its arrival time, localization and the precision of the localization. This information is sent to the observatories interested in studying the evolution of the system after the transient. A quick evaluation of the best follow-up strategy guarantees that multi-messenger observations are extremely fruitful minimizing its cost at the same time. Development of differentiable models of existing observatories and their characteristics, together with a suitable loss function would help in a quick calculation of optimal observation strategy for each event. Publication of such a strategy could be included in the GCN/TAN alert mechanism.

This section is devoted to the optimization of objective functions related with experiments.

4.2.1. High-energy gamma-ray astronomy

The field of very-high energy gamma-ray astrophysics studies the non-thermal universe at photon energies above tens of GeV to hundreds of TeV [260]. These gamma rays trace acceleration, propagation, and interaction of relativistic cosmic particles and provide insight into the most extreme environments around exploding stars and compact objects like black holes or neutron stars. Measurements of gamma rays from distant sources allow characterizing magnetic and photon background fields on intergalactic scales, and provide discovery potential in fundamental physics areas such as the search for dark matter or for potential quantum gravity effects. Today's observatories have discovered about 10^3 sources of gamma rays at energies above 1 GeV and roughly 150 sources of gamma rays above 100 GeV.

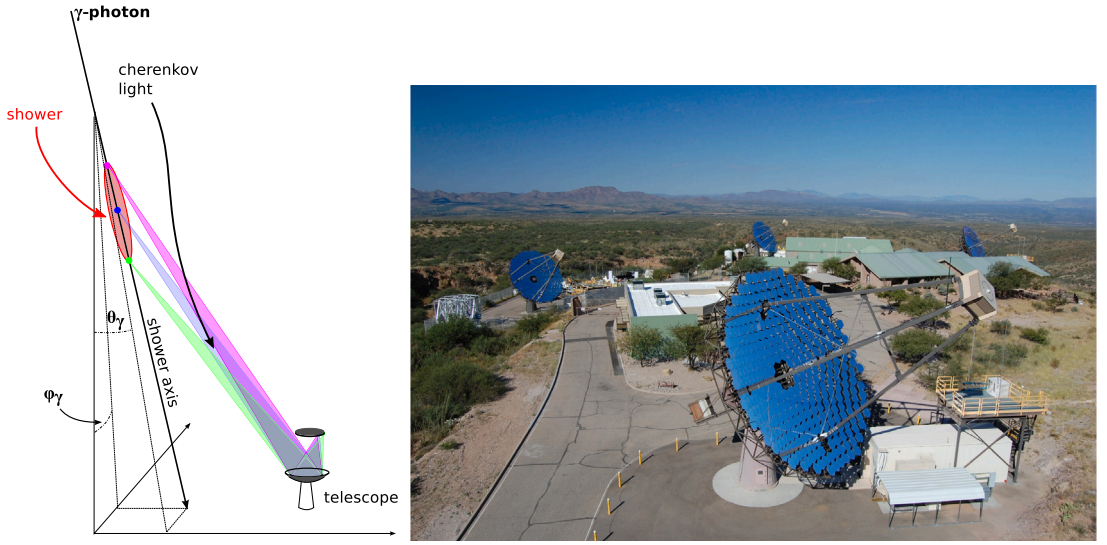


Fig. 19. Left: detection principle of imaging atmospheric Cherenkov telescopes. The sketch indicates the interaction of the primary gamma photon in the atmosphere, the development of the particle cascade, Cherenkov light emission and measurement by the telescope [261]. Right: the VERITAS observatory, consisting of four 12-m diameter telescopes at the Fred Lawrence Whipple Observatory in southern Arizona.

The key characteristic of all high-energy astrophysical sources are low fluxes (less than 10^{-11} photons/cm^2/s) which implies the necessity of detectors with effective areas far beyond the scale of space-based instruments. Imaging atmospheric Cherenkov telescopes detect the Cherenkov light emitted by particle cascades initiated by the high-energetic photons when entering the atmosphere. This detection technique provides a calorimetric and imaging measurement, allowing to reconstruct the energy and direction of the primary photon and provide strong rejection power to the far more numerous background particle cascades initiated by charged cosmic rays.

The measurement principle and the main components are indicated in Fig. 19. All modern Cherenkov observatories consist of several telescopes. This is advantageous for two reasons: first to suppress background events by high-energy muons, and secondly for stereoscopic reconstruction of the main air shower axis, providing improved direction and energy reconstruction. The effective area of ground-based Cherenkov observatories is in the range of $10^5 - 10^7$ m^2.

Major operating instruments are the MAGIC observatory on the island of La Palma consisting of two 17-m telescopes, the H.E.S.S. array in Namibia with five telescopes, and the VERITAS observatory in southern Arizona with four 12-m diameter telescopes (see Fig. 19). The next generation instrument is the Cherenkov Telescope Array (CTA), with two sites in Chile and La Palma with respectively 50 and 13 telescopes of different types. CTA is currently under construction, and it is expected to become operational in the mid 2020s.

State of the problem and current approach of optimization. The complexity of the measurement process and the numerous background sources require detailed Monte Carlo simulations to understand the performance of an array of imaging Cherenkov telescopes. The Monte Carlo simulations include the full development of air showers in the atmosphere. All relevant hadronic and electromagnetic interactions, and the characterization of atmospheric properties for Cherenkov photon generation and propagation (molecular density profile, aerosols) are part of the simulations. The telescope simulations comprise ray tracing of Cherenkov and background photons through the optical components, and the simulation of the photon detection and readout chain.

The current optimization approach relies on brute-force search covering a representative part of the parameter space by simulating a large number of telescopes of different types (*e.g.,* different optical or camera designs) at different positions on the ground followed by a selection of a subset of telescopes at the analysis stage representing a candidate realization for the observatory. This approach is inefficient, slow, and it involves a large amount of computing. As an example, a typical Monte Carlo production for CTA requires the simulation of ≈100 billion gamma ray, protons, and electron events using about 1.5 PByte of storage for the detector simulation output and 100×10^6 HS06 CPU hours [262].

It should be noted that this type of optimization is not only applied during construction, but also during operation: the optimal instrument configuration (*e.g.,* number of telescopes used, pointing mode, trigger settings) is different for the numerous types of astrophysical targets observed by gamma-ray observatories.

Figures of merit. The figure of merit are:

1. detectability of an astrophysical signal for a given observation time. This is usually expressed in differential flux sensitivity, meaning the minimum flux per energy required to obtain a detection;
2. sensitivity energy range and especially the required minimal energy threshold;

3. precision of the direction and energy measurement;
4. sky area covered by a single observation;
5. systematic uncertainty of flux, energy, and direction measurement.

Examples for optimizations. Imaging atmospheric Cherenkov telescopes are complex instruments with a large number of parameters influencing the performance. A telescope consists of one or two main mirrors with a collection area of the order of 100 m^2, a photo-detection plane consisting of an array of $10^3 - 10^4$ photo-multipliers or SiPM detectors, and a fast trigger and readout chain which allows capturing the nanosecond-long illumination from air showers. Fast trigger algorithms based on pre-defined patterns allow suppressing signals from night-sky background photons. Observatories can consist of a large number (>50 for CTA) of telescopes observing the same part of the sky simultaneously.

Example 1 - optimization of telescope distances: assume in this example an array of N imaging atmospheric Cherenkov telescopes ($4 < N < 100$). What is the optimal distance between those telescopes to achieve optimal performance for the figures of merits described before? A good starting point is the area illuminated by the Cherenkov light, which is roughly a circular area of 140 m radius. Increasing the telescope distance will enlarge the sensitive effective area and provide increase sensitivity at energies above several TeVs. Decreasing the telescope distance increases the telescope multiplicity per event and therefore improves the direction and energy reconstruction.

Example 2 - optimization of optical collection area with different telescope types: assume a gamma-ray observatory with up to three different types of telescopes, allowing to cover energy ranges from 10 GeV to 300 TeV. This setup is corresponds to the implementation of CTA, which will consist of three different telescope types (large-size telescopes with 23-m diameter mirrors; mid-sized telescopes with main dishes of 12 m diameter, and small-sized telescopes with about 4 m diameter dishes). How many of each telescope types are necessary for the optimum performances across the required energy range?

Example 3 - optimization of the observation mode for surveys, for which a large sky area is observed under consistent observation conditions. Parallel pointing of all telescopes restricts the observed area of the sky to the field-of view of each telescopes while maintaining large telescope multiplicities for precise reconstruction. In contrast, divergent pointing has each telescope pointing slightly offset from its neighboring pointing, allowing to cover a larger area with reduced telescope multiplicity.

Outlook and possible implementations. The optimization of the configuration of ground-based arrays of imaging Cherenkov telescopes would benefit significantly from a differential model. The possible objectives would not only include finding the optimal instrument, but also to optimize the observation mode of operating gamma-ray instruments. The simulation and analysis chains are implemented in modules and the new optimization approach can be implemented step-by-step (*e.g.,* by parameterizing the single telescope contribution and optimize the array configuration only).

4.2.2. Interferometric gravitational-wave detectors

The design of gravitational wave (GW) detectors [263,264], has given spectacular results only recently when in 2015 the first direct detection of gravitational waves [265] was announced by the end interferometric ground base detectors LIGO/Virgo/Kagra (LVK). However in this moment there are other ambitious designs going on in this field. Among them there are space missions like the Doppler space-craft tracking or the GW in-space antenna LISA [266]. This list can be extended to the next generation detector Einstein Telescope (ET) [267]. At present, these technologies are at different stages of development. Doppler space-craft tracking was implemented and tested in the past in several space-missions; LISA has proven its concept by completing all the tests of the LISA Pathfinder mission in 2017 [268]. LVK has concluded the third observing run O3 which brought a set of new observations [269] and permitted general studies on the Universe [270]. LVK members are currently preparing for the O4 by upgrading their detectors and planning its activity in the future [271] with the aim to continue the search for known coalescence signals and looking for yet undetected waveforms like continuous gravitational waves [272] and waves originating from binary sources with extreme configuration [273]. In the following paragraph we will focus on the technology of ground-based interferometric antennas, in order to find possible applications to differentiable programming in improving their design.

State of the problem and current approach of optimization. AdvancedVirgo+ [274–276] is conceptually a quantum-enhanced doubly recycled large Michelson interferometer (ITF) with Fabry–Perot arm cavities, capable of detecting GWs in the frequency range between 10 kHz to 20 kHz. Fig. 20 depicts the configuration of seven mirrors which are the main optical components. Despite its apparent simplicity, the detector contains more than one thousand optical components divided in subsystems as follows: pre-stabilized laser, injection, detection, thermal compensation system and quantum noise reduction system. Auxiliary optics are placed on optical tables close to the main optics. Gravitational wave detection is based on the sensing of the relative position of the main optical components (mirrors). Isolation of these components from vibrations is fundamental since the ambient where the detector is located is far from being quiet. Main mirrors hang on fibers made of the same material as the substrate forming a monolithic block called a monolithic suspension [277]. The entire set is subsequently suspended on the multi-stage vibration isolator called super attenuator [278]. This sophisticated, actively controlled mechanical structure provides vibrations attenuation by a factor of 10^{13} in the ITF frequency operation range. Many of Virgo optical benches are also suspended on dedicated smaller suspensions. Suspended elements are encapsulated in vacuum tanks which ensure cleanliness, additional layer of both acoustic and electromagnetic insulation, and stabilization of the light rays path at the same time. The amount of components in the detector, the required precision of their relative position, and the presence of suspended optical components requires implementation of active control loops. Many of the Virgo components are controlled actively or continuously monitored. For this purpose, Virgo uses a genuine, distributed data acquisition and signal processing system (DAQ) [279,280]. During ITF operation the DAQ assures the proper alignment, automatic lock, and lock recovery, and registers scientific data. In the following we consider a possible application of differentiable programming in the ITF design.

Fig. 20. Schematic diagram of the second generation interferometric gravitational wave detector. For simplicity, only the main optic elements are depicted. Left: laser source and detection are schematically depicted as black box and orange semi-circle. Right: Virgo aerial view. The two 3-km arms of the interferometer meet the central building where the laser is generated, and five mirrors, including the beam-splitter mirror, are hosted. Optical benches used to acquire the interference pattern are also hosted in this building. On the left part, the other buildings host EGO offices. Credits: the Virgo Collaboration/N. Baldocchi. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Figures of merit. GW detector input signal is the effect of the gravitational wave on the stretched/compressed space, gravitational wave strain. Strain amplitude depends only on the amplitude of the incoming gravitational wave. All noises in the gravitational wave detector are expressed in terms of coupling to the strain. This makes the distribution of noise in the noise budget and the detector observation range a direct measure of the ITF figure of merit. LVK collaboration developed `gwinc` [281], a top-level simulation tool which includes and sums up all noise contributions. However, the important information provided by `gwinc` is not sufficient to optimize an individual subsystem, and more detailed simulators are involved in the design process. We can distinguish three types of design in ITFs: mechanical, optical, and electronic. In general, most of the design tasks belong to more than one class. Considering the long period an ITF spends in the observing run, the second figure of merit must be attributed to the stability of the overall noise and to the detector duty-cycle. Below, we describe possible cases of application of differentiable programming in this context.

Examples of optimizations. At a first glance the ITF optical system and ITF base topology are the main subjects of possible optimization. This is certainly the case for the ET design. In case of Virgo, its optical setup has been already heavily optimized for a wide spectrum of parameters. The optimization was performed manually, with support by the three main types of software: FFT simulators, ray-tracing simulators, and modal simulators [282–284]. None of the mentioned tools are able to simulate all aspects of the detector and none of them was implemented with differentiable programming. In this field, the first necessary effort is the development of differentiable programming-compliant simulators. The current Virgo base topology is itself a perfect reference for testing the future software. Unlike the ITF base topology, the optical subsystems surrounding the main optical elements certainly belong to the first class of design task, which can be optimized in view of their future upgrades. In particular, a possible candidate for optimization is the development bench for EPR squeezing [285,286].

The second class of challenging design tasks is constituted by mechanical structures. Among the most common parameters of mechanical systems, like resonant frequencies or breaking load, interferometric GW detectors rely on the characteristics of thermal noise. In Virgo, mechanical design is aided by analytical models specific for each design task, such as super-attenuator, and payload and monolithic suspension. In some cases, the finite-elements method was used to increase simulation completeness [287]. Careful analysis of existing models and their upgrade to a differentiable model is the first step to make. In our opinion, payload optimization is a good candidate to test the automatic optimization approach for mechanical design tasks.

The Einstein Telescope is a new generation GW detector, currently in a phase of conceptual design and funding. Its design is strongly inspired by the LIGO/Virgo/Kagra success. Certainly, ET will inherit many technologies from existing interferometric GW detectors. At variance with the Virgo case, ET can benefit from differentiable programming models already at the conceptual design stage, where both base topology and specific subsystems can be optimized by gradient-based methods. This outcome strongly motivates the development of differentiable simulators for interferometric GW detectors.

Outlook and possible implementations. The most efficient way to benefit from differentiable programming based optimization is probably the one of following the list of noise contributions and to providing a suitable software tool at the beginning of the design process. However, this approach may not be immediately implementable, hence the workflow may be conceived differently; specific simulators may be built and then exploited to solve complex design tasks.

4.2.3. Radio detection of high-energy neutrinos

Neutrinos are considered a perfect cosmic messenger as they traverse the universe unimpeded, and their flight direction points back to their sources. Ultra-high energy (UHE) neutrinos will provide insight into the inner working of the most violent phenomena in our universe, those that happen in the vicinity of super-massive black holes (*e.g.,* in active galactic nuclei), in neutron star mergers, or gamma ray bursts. The detection of these ghostly, extremely energetic elementary particles will be one of the most important discoveries in astro-particle physics in the 21st century.

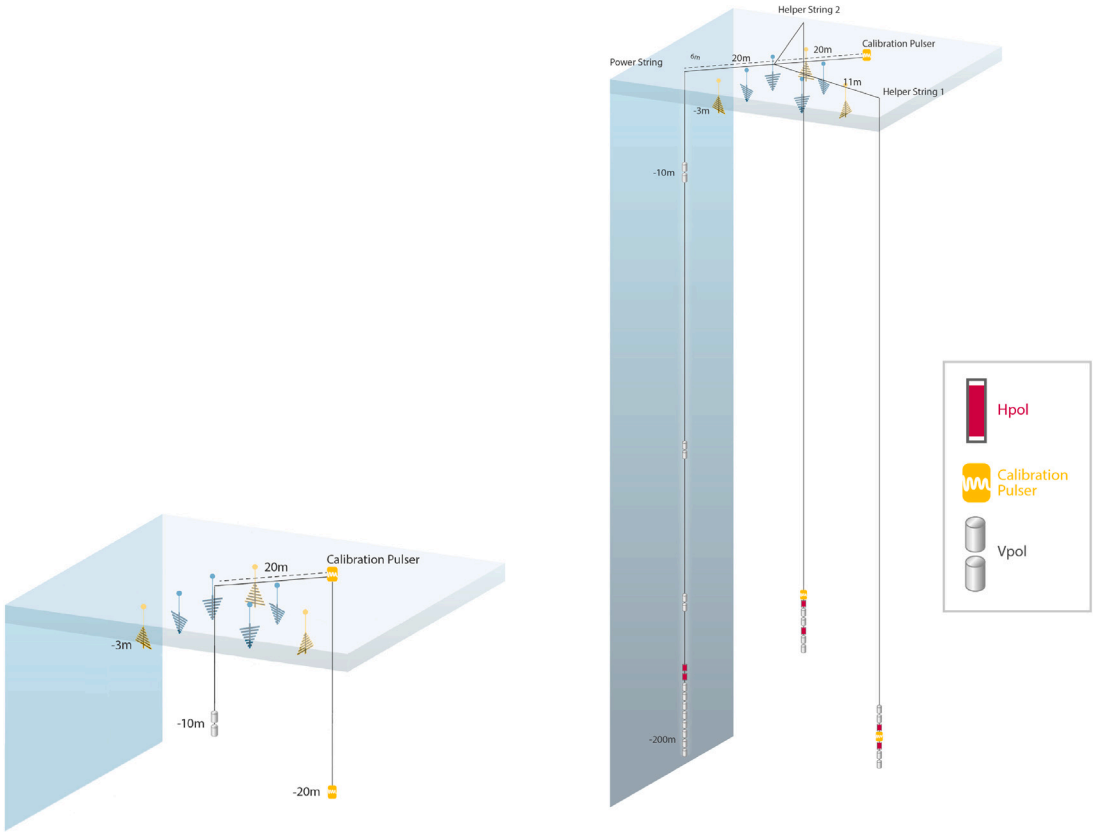


Fig. 21. Two possible radio detector station designs considered for IceCube-Gen2.
Source: Figures reproduced from Ref. [293].

To be able to measure the low flux of UHE neutrinos on Earth, a new detector technology has been developed, instrumenting polar ice sheets with radio antennas to search for neutrinos passing through the ice. A sparse array of radio-detector stations can instrument large volumes efficiently due to the large attenuation length ($\approx$1 km) of radio signals in ice. Radio antennas measure the radio emission created by a neutrino-induced particle shower in ice via the Askaryan effect [288].

The technology to build and operate an array of radio detector stations that instrument the polar ice to catch the elusive UHE neutrino interactions has matured into small pilot arrays over the last decade [289,290]. Now, for the first time, a sizeable detector is being constructed. The deployment of the Radio Neutrino Observatory in Greenland (RNO-G) started in June 2021 [291]. Detector completion is expected in 2024. 35 autonomous radio-detector stations will then instrument 300 Gigatons of ice. At the same time, an order of magnitude larger radio array is being planned as part of the IceCube-Gen2 efforts to build the next-generation neutrino observatory at the South Pole [292,293] with the start of production as early as 2025.

Figures of merit. The development of the NuRadioMC simulation code [294,295] enables a precise estimation of the figures of merits for arbitrary detector designs. The figures of merit are (1) the overall sensitivity of the detector, *i.e.,* the expected rate of detected neutrinos; (2) the achievable reconstruction resolution of the neutrino energy, direction and flavor; (3) the ability to reject rare backgrounds. These three quantities can be combined into high-level sensitivity estimates, *e.g.,* diffuse flux sensitivity or the discovery potential for point sources. At the same time, the monetary cost of the detector as well as deployment and engineering constraints need to be considered to formulate a sensible objective function; otherwise, a larger detector with more radio detector stations would always provide better performance and be preferred. Thus, the formulation of an objective function requires care.

State of the problem and current approach of optimization. A working radio detector that fulfills the main scientific objectives of the experiment can be built with a variety of different station designs. The main differences are in the number of antennas per station and their largest depth, as well as in the total number of radio stations and their separation. Two examples of station designs are shown in Fig. 21. For example, the sensitivity per station can be increased by installing the antennas deeper in ice, but at the same time the costs and deployment efforts increase. For similar monetary costs and deployment effort, a larger number of shallow detector stations (see Fig. 21, left) can be installed yielding the same overall sensitivity. Furthermore, the relative positions and orientations of antennas impact the reconstruction resolution. These two examples already show the large parameters space that needs to be considered.

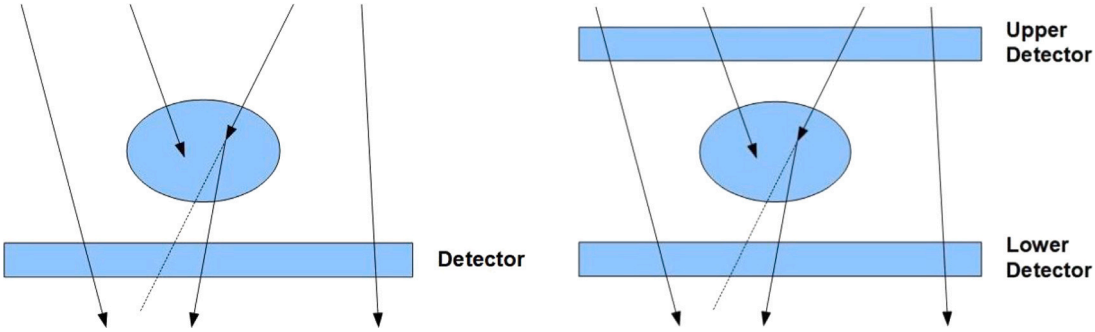


Fig. 22. Typical detector configurations in muography, with respect to the object to be imaged, with the absorption method (left), where the fraction of muons surviving energy loss is measured, and with the scattering method (right), where the observable of interest is the root-mean-square of the deflection angle.

An estimation of the global figures of merit is estimated to require a time-consuming MC production with about 1 million core hours per design. The current optimization approach relies on estimating a good station and array layout using scaling relations, and only testing one or a few options with a full MC production.

Outlook and plan towards an automated detector optimization. An automated detector optimization bears large potential to further optimize and fine-tune a radio neutrino detector. A first viable step would be the optimization of antenna positions within a single detector station with the objective of minimizing the reconstruction uncertainties of the neutrino energy and direction. This requires (1) a substantial speed up in the MC simulation and (2) an automated estimation of the reconstruction performance. Part (1) can be achieved using surrogate models (*e.g.* through a generative adversarial neural network). Part (2) can be achieved through a deep-learning-based reconstruction method. The MC data set itself is used to learn to predict the neutrino direction and energy. The performance of the network is validated on an independent test dataset and used as a proxy for the reconstruction performance. Initial work (*e.g.* Refs. [296,297]) shows that deep neural networks are capable of reconstructing the neutrino properties with good precision; however, large training data sets of >1 million events are required. The demand of training data size can be reduced by starting with a pre-trained network; *i.e.,* a large training data set only needs to be produced once, if an efficient data resampling scheme is developed.

Optimizing the speed of estimating the reconstruction performance for different antenna positions is challenging but seems doable with the steps outlined above and within the framework that the MODE collaboration has been developing.

4.3. Muography

The abundant natural flux of atmospheric muons, and their large penetration power, have been exploited for the imaging of a large variety of objects spanning in size from O(1 m) to O(1 km), with applications including archaeology, volcanology, homeland security, nuclear safety, industrial process control, and many others [298]. This technique is referred to with several names, such as Muon Radiography, Muon Tomography, Cosmic-Ray Muon Imaging, or the neologism “muography”; in the rest of this document we use the latter term. In some applications of muography, the volume of interest can be sandwiched between two trackers and one can measure the scattering of the muons that cross the target volume, which is correlated with the atomic number Z of traversed material. When the target volume is very large (*e.g.,* a mountain or an entire building), a single tracker is located downstream to measure the absorption of the muon flux through the target, from which a density map can be derived. The two methods are illustrated in Fig. 22.

By optimizing the layout of the detection elements, possible large gains in the resolution and material identification potential of a muon tracking system are achievable. This is a domain of application where detector development can have a very fast turn-around, and where one may want to quickly react to changes in the practical constraints (*e.g.* costs and budget may evolve during the lifetime of the project) or even in the goals of the project itself (*e.g.* switching from a search for cavities to material discrimination, for which the figure of merit is not the same, or changing the object to be imaged). Therefore, an automatic optimization procedure is very appealing in the context of muography, as it may have to be re-run several times along the lifetime of a single project.

The rest of this sub-section is organized as follows: Section 4.3.1 elaborates on the variety of figures of merit that depend on the specific use case; Section 4.3.2 lists the typical detector parameters to be optimized and distinguishes them by categories; Section 4.3.3 presents a new software project promoted by MODE, which has the goal to apply differentiable programming to a category of muography applications; Section 4.3.4 provides examples of industrial applications that can profit from the MODE approach; finally, Section 4.3.5 illustrates a project for a portable and modular detection set-up for muography that would maximally profit from direct end-to-end- optimization procedures.

4.3.1. Figures of merit

The figure of merit to be maximized or minimized may be very different for different use cases, leading to different outcomes for the optimal detector setup.

In many archaeological use cases the goal is to look for unknown voids, without assumptions on size, shape, and location (see, *e.g.,* Refs. [299,300]). An appropriate figure of merit might thus be the sensitivity to localized excesses in muon flux with respect to a baseline (similarly to searches for new particles or resonances in particle and nuclear physics).

In other cases, the goal is to identify the position, size, and shape of materials that are known or presumed to be present in the field of view of the detectors, *e.g.,* ice and rock in Ref. [301], or various special nuclear materials surrounded by shielding materials, in applications such as nuclear waste monitoring [302]. In these cases, an appropriate figure of merit may be the sharpness of the images obtained. Depending on the atomic number Z of the relevant materials, the size of the target volume, and other factors, the absorption or scattering methods may be more appropriate, and there are some gray areas where it is not obvious *a priori* which of the two detector configurations of Fig. 22 is the most convenient; the choice should be made after the parameters are optimized for each option.

Identification of special nuclear materials is also the goal of the "muon portals" that exploit the scattering method for the inspection of cargoes at border controls [303] (see also Section 4.3.3 below). However, the relevant figure of merit is different in this case: more than forming an image, the aim is to fire an alarm when a forbidden material is present in the cargo. The constraints set by the border authorities include the time allowed for each inspection in the portal (typically of the order of minutes), and the false positive rate, because each alarm would cause a lengthy inspection, hence a financial compensation to the cargo company for the time lost in case of a false alarm. Given these constraints, therefore, a pertinent figure of merit would be the false negative rate given a fixed time and a false positive rate.

4.3.2. Parameters of the optimization task

We distinguish here between local and global parameters. Local parameters are related to individual detector units, and are specific of the technology employed. Global parameters instead concern the geometry and spatial deployment of the overall detector setup.

The local parameters are the same as for any other tracking devices, such as the number of strips or pixels, and their pitch. Global parameters include the number of layers, their distance, and the surface area of a single layer (which can be composed of several adjacent detection units). Unless the location and orientation are entirely constrained by logistic considerations, the optimization can also take into account global parameters such as the distance from the target and the inclination of the apparatus with respect to the zenith angle (from where most of the muon flux is coming, with an approximate $\cos^2\theta$ dependence).

Multiple Coulomb scattering induces angular deflections whose root mean square depends on the inverse of momentum, implying that low-momentum muons are a nuisance to imaging as their arrival direction is less correlated with their trajectory within the target. This blurring can be reduced by either rejecting muons below a certain momentum threshold or by taking momentum into account as an input to the imaging algorithms. However, muon radiography detectors do not measure momentum directly, as that would demand the usage of large magnets, making the apparatus very expensive and cumbersome. Instead, the usage of slabs of passive material (*e.g.* lead or steel) is a popular method to either estimate momentum indirectly (through the multiple scattering induced in a known volume of known material) or simply absorb, hence remove, the low-momentum muons up to a certain threshold (at the same time removing also some important backgrounds such as charged hadrons and $e^{\pm}$ by inducing their showering). Therefore, an additional set of global parameters, which is specific of muon radiography, relates to the presence of passive material, either for momentum filtering or as reference for scattering. Passive material can either be installed all in single place in the apparatus or diffused across the layers, as illustrated respectively in the top and bottom panels of Fig. 23. In this sense the problem is similar to the deployment of the passive material in MUonE, as mentioned in Section 4.1.6. The corresponding parameters for the optimization include the material, thickness, number, and position of the passive slabs.

Once an optimal set of parameters is identified for a given use case, and a set of muon detection planes is built, changing use case demands a re-optimization of the parameters, but a new optimization may have to be limited to the global parameters, as re-building all detector units from scratch may be unfeasible due to budget or time constraints (although the costs and timescales involved in typical muon radiography projects are much smaller than for many of the other examples in this document). For example, the angular resolution of a tracker depends on both the spatial resolution and the distance between planes; while the former depends on local parameters, the latter is a global one, and it is way cheaper to adjust.

4.3.3. TomOpt: Differential Muon tomography optimization

TomOpt is the first concrete step within the Mode collaboration to investigate the practicality and scalability of the optimization pipeline proposed in Section 1. Muon tomography offers a comparatively simple optimization use case among detector systems. As discussed above, the optimization of such detectors can have a significant benefit, while allowing us to simultaneously bring up our own understanding of the requirements, feasibility, and potential limitations of physics-goal-oriented systems optimization.

Package overview. While still in the development phase, the TomOpt package is planned to be a highly-modular, python-based, PyTorch backed [52], framework providing users with a full suite for implementing and simulating all required aspects of the optimization pipeline in a flexible and extensible manner:

- Muon generation through random sampling of literature muon flux models, *e.g.* Refs. [306,307];
- Definitions of initial detector configurations;

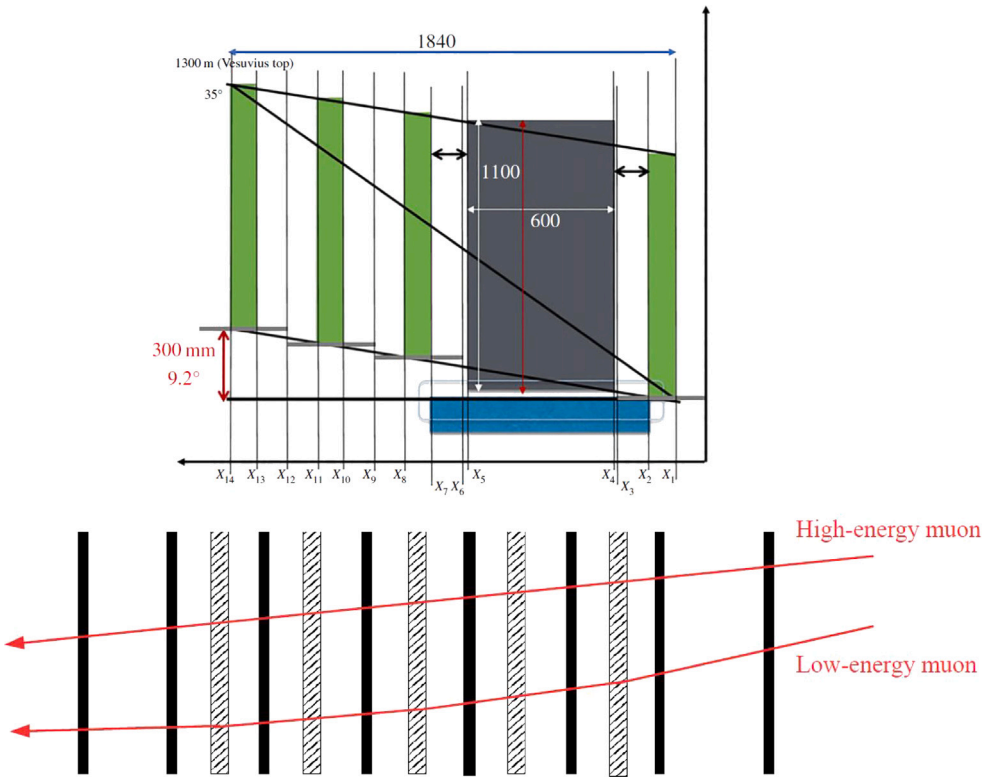


Fig. 23. Two ways to arrange passive material for momentum filtering or momentum estimation in a muography apparatus. In the MURAVES experiment [304] (top) a thick lead wall (gray rectangle in the figure) is placed before the last active layer (the object of interest, Mt. Vesuvius, is on the left). In the muon telescopes of the Sakurajima Muography Observatory [305] (bottom) relatively thin slabs of passive material (gray) are alternated with MWPC detectors (black).

- Passive volumes to be imaged (either pre-specified by the user, or randomly generated according to specified criteria), consisting of a range of materials;
- Muon propagation through volumes, and scattering within passive matter according to GEANT 4-based models;
- Inference of properties of the passive volume (based either purely on classical methods, such as Point of Closest Approach (POCA) [308], likelihood fitting, or contemporary methods based on deep learning);
- Loss functions incorporating terms based on both the predictive performance of the detector system, and its cost (whether this be fiscal, heat generation, power requirement, exposure time, etc.);
- optimization of the detector system based on the loss, using a variety of standard gradient-based optimizers (SGD [72], Adam [309], etc.), over a training cycle backed by a stateful callback system.

Given TOMOPT's emphasis on modularity, PYTORCH was chosen as the back-end graph-computation library due to the `torch.nn.Module` class, which handles computation in the forwards pass and exposes parameters to the optimizer for the backwards pass. With its relatively abstract definition, but nonetheless generally useful functionality, it provides a convenient parent class for developing the various necessary modules in TOMOPT.

In its current state, the full pipeline is in place, and development has now moved to focus on improvements to generalization, realism, and inference performance. Our intention is to publish demonstrations of optimization for a range of benchmark scenarios in 2023, along with the first public release of the package, at which point it will move in to the phase of open and continual, community-driven development under a free open-source licence.

Muon propagation through volumes. Initial muon kinematics are sampled from literature models of muon flux, *e.g.* Refs. [306,307], and may undergo transport away from sea-level, as required. Muons then pass through the passive volume and detector systems, while undergoing multiple scattering. The passive volumes are modeled in terms of discrete voxels of varying material. As the muons pass through the passive volumes, they undergo multiple scattering according to the distance traveled in, and the radiation length (X_0) in the material of, each voxel. A parameterized scattering model is used, based on GEANT 4 simulation.

Muon hits are recorded above and below the passive volume. To allow for a reasonably flexible detector system, detector panels are allowed to float within specified regions of space. Each panel has a fixed spatial-resolution for muon hit recording, with a fixed efficiency. The parameters to be optimized are their (x, y, z) position and transverse spans (x, y). Currently, the number of panels is also fixed, but a simple extension will allow their variation at optimization stage. During optimization, the hit positions are made

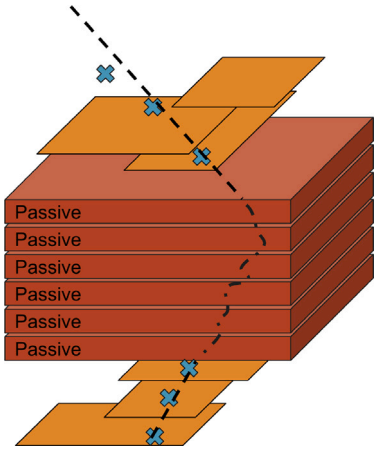


Fig. 24. Example schematic for muon propagation (black dashed line) through a volume. The passive volume (red layers) consists of voxels of varying materials, and the detectors, indicated as orange panels, record hits (crosses). Note that although the muon does not pass through the top-most detector panel, a low-resolution hit is still recorded due to resolution and efficiency being currently modeled as a distribution which extends outside the panel. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

differential w.r.t. the panel parameters by replacing the resolution and efficiencies of the panels with parameters distributions, allowing hits to be recorded outside the panels, but with a much lower precision than if the panel were centered on the muon path; this ensures differentiability of the relevant optimization metric. Fig. 24 shows a diagrammatic example of a muon propagating through both the detector and the passive volume.

Inference. Inference is based on the POCA method, by fitting incoming and outgoing trajectories to the muon hits and extrapolating the trajectories inside the passive volume. Through inversion of the scattering model, an X_0 may be predicted based on the differences between the trajectories. Since such a method effectively assigns the whole of the scattering to a single voxel, it is inherently biased. Additionally, the POCA carries an uncertainty due to the finite resolution of the detectors, which is computable via autograd-based uncertainty propagation.

This uncertainty may be accounted for by replacing the POCA with a 3D Gaussian PDF which scales according to the POCA uncertainties. A weight per muon per voxel can be computed by integrating the PDF over each voxel. This weight can then be augmented by the uncertainty in the X_0 per muon (again computable via autograd), and their hit efficiency. The final predictions for each voxel in the passive volume is then a weighted average of the predictions of every muon.

While this method accounts for the uncertainty on the POCA location, it does not completely address the inherent bias in the predictions. An alternate approach is to use a frozen graph neural network (GNN) [310] that has been pre-trained for the specific type of passive volumes expected. Such a network can take in features for each muon (including the POCA predictions) and output predictions for the volume (whether this be some overall prediction for the volume, or predictions per voxel). Since the GNN acts as a flexible and differentiable map from muon-level predictions to target space, it can learn to avoid biases in a way that the fixed-computation model-inversion approach cannot.

Loss definitions and optimization. In general, the detector should aim to be as performant as possible, while being within budget. As previously discussed, the budget may contain multiple factors, not just pecuniary ones; such as size, imaging time, and power requirements. These will depend on the exact scenario for which the detector is being optimized. Similarly, the definition of "performance" will also vary according to task. Provided these aspects can be expressed as analytic and differentiable functions of the detector predictions and parameters, then the detector can be optimized with respect to them via gradient descent.

An example loss function might be: the mean squared-error of X_0 predictions over every voxel in each volume, for the performance component; and the total cost of the detector, which scales according to the xy span of each panel. The cost component, however, does not include any notion of a budget. Instead, the functional form can be adjusted to slowly turn on a cost penalization as the actual cost approaches a pre-specified budget, beyond which the penalization rapidly increases, such as the form shown in Fig. 25. The performance and cost components of the loss will also need to be scaled appropriately to provide a good optimization, but the scaling component can be approximated by first capturing the initial performance of the detector in a frozen state over a few warm-up epochs at the beginning of the optimization process.

While the loss can be computed for the voxels in a passive volume, to ensure complete generalization the loss should be averaged over a sufficient number of representative examples of passive volumes. This can be achieved by specifying the type of passive volumes expected, and generating examples on demand. Comparing to traditional DNN training over mini-batches of data, optimization here then happens over batches of passive volumes. Once a loss has been computed, it may be back-propagated to each of the detector parameters, which may then be updated using gradient descent.

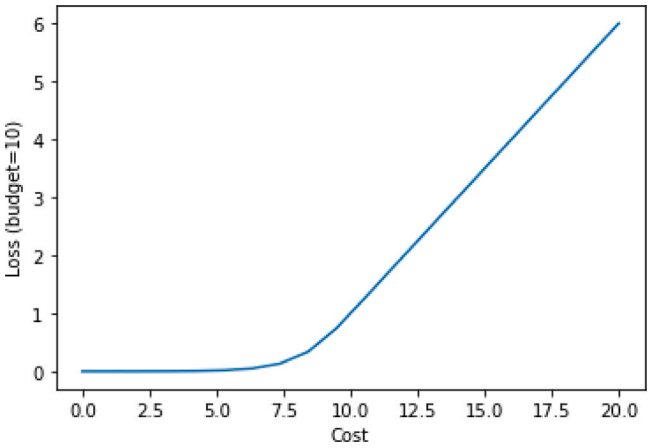


Fig. 25. Cost component of the loss function as a function of detector cost, for a target budget of 10 A.U. A sigmoid component below the budget provides a slowly increasing gradient as the cost approaches the target budget, beyond which the loss increases linearly. The function is fully differentiable.

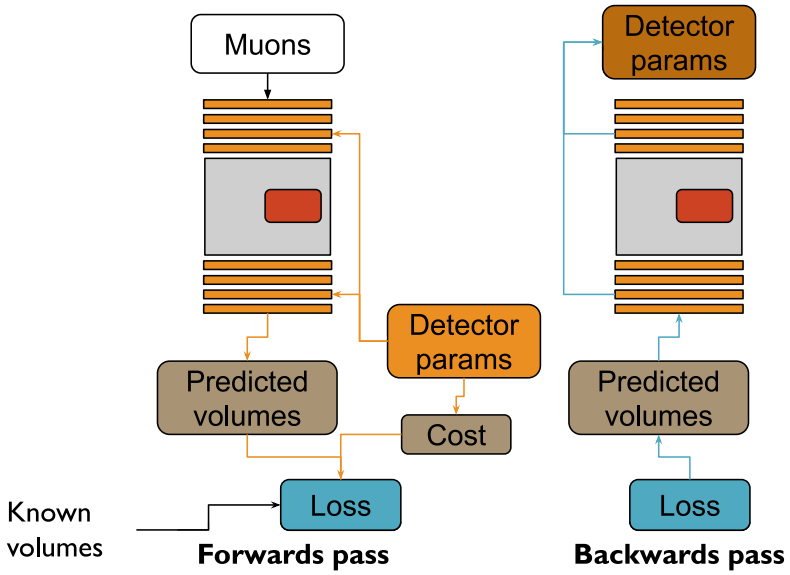


Fig. 26. Example schematic for the parameter update loop in TomOpt. In the forwards pass, predicted volumes (or target features of the volumes) are computed based on inference of muons reconstructed by the parameterized detector. In the backwards pass, the derivative of loss is passed through the inference algorithm back to the detector parameters.

Update cycle. Fig. 26 illustrates the forwards–backwards update cycle in TomOpt. In the forwards pass, batches of muons are used to scan known (and possibly randomly generated) passive volumes. A fully differentiable inference algorithm is used to predict target values for the volumes, based on the detector readouts. Finally, the loss of these predictions is combined with the cost of the current detector system. In the backwards pass, the gradients of the loss with respect to the detector parameters is computed, and used to adjust detector parameters.

4.3.4. Industrial applications

Muography can offer solutions to many problems in the industrial sector working as a Non-Destructive Testing (NDT) technique to perform preventive maintenance of equipment, quality control of the production process, and risk assessment and evaluation. There is a large variety of problems in the industry although most of them share the following characteristics: relatively large and dense objects, impossibility to have access to the object while the equipment is in production, and presence of a harsh environment in terms of dust, temperature, and space or time restrictions. Muography emerges as an interesting NDT due to its large power of penetration, the fact that its application does not require any physical contact with the target objects allowing inspection while the factory is in production, and the possibility to perform a continuous monitoring.

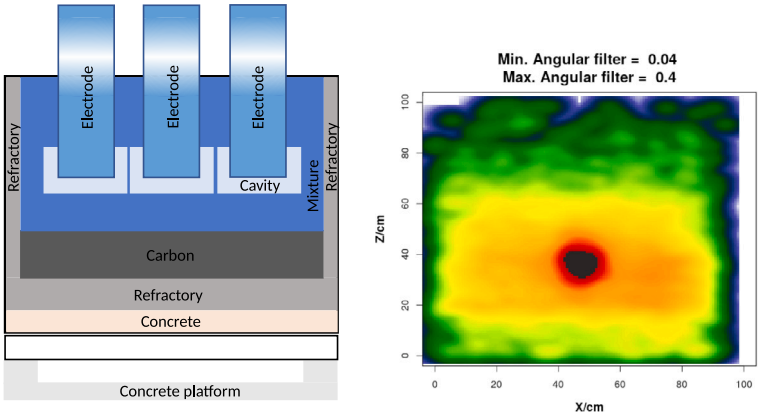


Fig. 27. Schematic view of the structure of an electric furnace including three electrodes used to heat the mixture (left). POCA-based image of a steel pipe used to feed a convolutional neural network working in regression mode to infer the inner radius (right).

Most of the industrial applications have another distinguishing feature with respect to other common applications of muography: the nominal geometries and densities of the objects are known with high precision and can be extracted from the engineering drawings. This implies that muon imaging algorithms are not required to reconstruct an unknown object but rather to find variations on top of a well known, predefined design. A consequence of this is the dramatic reduction of complexity of the problem from an algorithmic point of view. Geometry variations can be encoded in a relatively small set of parameters that can be estimated using methods such as Deep Neural Networks working in regression mode or likelihood-based estimation methods. The following lines show a few examples of this kind of applications.

Electric arc furnaces are used in the processing of several raw minerals such as silicon or manganese. These furnaces are equipped with three large electrodes inserted vertically in the mineral mixture to produce electric discharges and heat the material. A schematic view of a generic electric furnace can be seen in Fig. 27(left). The knowledge of the exact position of the electrodes is an interesting parameter that could help to understand the efficiency variations observed in this kind of equipment. Scattering muography can be applied to this problem by simply considering the angular and spatial deviation distributions characterized using their n-quantiles. This information can be used to train a Deep Neural Network using simulated datasets with different electrode locations and regressing over the height of the electrode with respect to the bottom of the furnace.

Pipes are used in almost every industrial plant to transport all kind of liquids and gases. The wear of the inner wall of the pipes is a general problem that forces the companies to perform inspections regularly in order to keep the integrity and safety of the facility. In cases where the pipes are thermally insulated, the application of other NDTs is complicated due to the properties of the low-density insulation layer. Scattering muography can be applied in order to measure the inner radius of the pipe. A first attempt uses a likelihood-based algorithm in which the distributions of the angular and spatial deviations are estimated for every muon by simulating its propagation through a pipe with a given radius. The distributions are interpreted as probability density functions and are combined in a likelihood function for all the muons. This function is then minimized against the radius in order to find the best match. A second approach was used by estimating the POCA observables of the muons and using the corresponding images to train a convolutional neural network (CNN) performing a regression to the inner radius. Fig. 27(right) shows an example of a real POCA image of a steel pipe used as input for the CNN algorithm.

The industrial sector offers a large and heterogeneous class of problems where muography could provide cost-effective solutions. The estimation and imaging algorithms used in this context have been application-specific so far, complicating the exportation of the solutions to different problems. The MODE collaboration is aiming to device a general purpose procedure to solve the inverse problem associated to muography. This approach will allow unifying muography applications in the industry from the algorithmic point of view, allowing to quickly apply the technique in different cases and problems.

4.3.5. Portable modular detectors for flexible muography

One of the ongoing trends in muography, especially for applications in geosciences, archaeology, and civil engineering, is the development of portable and inexpensive muon detectors, such as Refs. [311,312]. A recent project [313–316] aims at the development of compact, autonomous, portable, and modular muon radiography setups based on position-sensitive layers based on small-area resistive plate chambers (RPC), a technology chosen because of its good trade-off between position and time resolution, efficiency, cost, and ease of construction [317]. In this project, the goal is to allow a high degree of modularity of the geometry of the complete setup: ideally, the already mounted individual detector planes would be produced in large numbers and deployed *in situ* in the arrangement that most fits the specific use case while respecting local constraints (*e.g.,* the optimal location may be in a narrow tunnel), as illustrated in Fig. 28. The same detection layers may be arranged to form two or one tracker depending on the relative importance of scattering or absorption on the final discrimination power. In either case, it is not always obvious *a priori* whether it is more convenient to have few layers with large area to collect more statistics (arranging several detector units

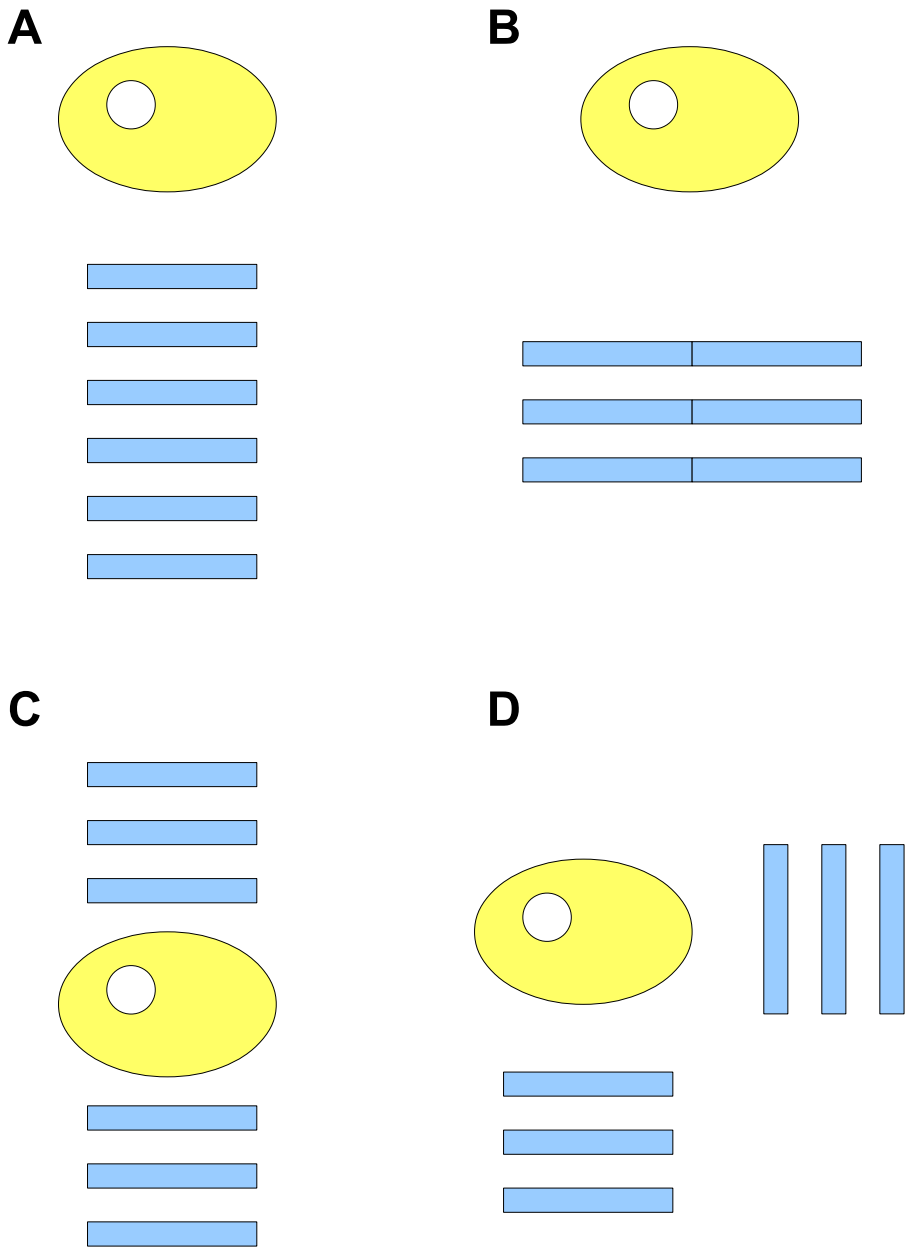


Fig. 28. Sketch of various detector configurations among which one could switch with fast turn-around in the context of a modular muography detector project such as the one described in Ref. [315]. The same six detector planes may be deployed in order to maximize angular resolution and trigger redundancy (A) or to maximize the acceptance and therefore the statistics (B); to perform a 3D tomography instead of a simple 2D projection, where appropriate the same planes can be used to "sandwich" the volume of interest and exploit the scattering method (C) or to take 2D projections from orthogonal points of view (D).
Source: Reproduced from Ref. [316].

side by side) or to maximize the number of layers crossed by the muons to improve resolution and redundancy of the setup. An automatic optimization algorithm based on machine learning, therefore, would allow for quick redesign of the geometry at every new measurement of a different target.

4.4. Proton computed tomography

Radiation therapy using energetic protons instead of X-rays for the treatment of cancer is becoming more and more available on a world-wide scale. As the energy deposition of protons is concentrated at the *Bragg peak* at the end of their trajectory, it promises a

better ratio between the doses deposited inside and outside the tumor [318,319]. Despite the differences in how protons and photons interact with matter, the planning of proton treatment is still based on computed tomography images, *i.e.* the spatial distribution of X-ray attenuation inside the patient (*phantom*), obtained from X-ray measurements from various angles. A direct measurement of the *relative stopping power* (RSP, the ratio between the stopping power of a certain material and that of water) of protons inside the phantom can inform treatment planning in a better way.

To this end, many prototypical *proton computed tomography* (pCT) scanner designs have been reported in the literature [320,321]. The processing of the raw measurements can be logically split into two parts:

First, a hardware-specific sub-procedure determines the positions and directions of protons both before entering and after exiting the phantom, and the energy loss in terms of a *water-equivalent path length* (WEPL).

Second, a reconstruction algorithm is applied to find the three-dimensional RSP image. In this respect, applying a *model-based iterative reconstruction* (MBIR) algorithm means that a linear system

$$A \cdot \mathrm{RSP} = \mathrm{WEPL} \tag{28}$$

is approximately solved in a least-squares sense, possibly in addition to image quality objectives. The entries of the matrix A reflect how much the voxels of the RSP image overlap with the proton paths. The paths must be estimated according to their entry and exit positions and directions [322], possibly taking the uncertainties in the measurements into account [323].

These two sub-procedures are visualized by the central and right arrow in Fig. 29. For optimization and other computational purposes, the raw measurement data can be manufactured by a randomized particle physics simulation, based on a model of the hardware and the RSP distribution inside the phantom. This is the leftmost sub-procedure listed in Fig. 29.

Figures of merit. The accuracy achievable with the utilized hardware and software design can be assessed by the following comparisons [65]:

- Comparing the reconstructed proton paths with the proton paths found in the MC simulation.
- Comparing the estimated WEPL with the true energy loss found in the MC simulation.
- Comparing the reconstructed RSP solution of (28) with the original RSP that the MC simulation was run on.
- Finally, when the reconstructed RSP image informs a proton therapy treatment plan (via a fourth sub-procedure on the right in Fig. 29), this plan can be assessed *e.g.* in terms of doses inside and outside the tumor.

Parameters for optimization. On the hardware side, design parameters

- of the proton beam, such as its energy, intensity, spot size and divergence, and
- the geometrical setup, including the relative positions, material budgets, and granularity of detector layers,

can be optimized. The software also involves constants that could be tuned:

- Sophisticated path estimation via the (extended) most likely path [322–324] involves coefficients of a certain polynomial fit to simulation data [325] or the assumed uncertainties of the measured positions and directions, among others.
- If a neural network is used somewhere in the software pipeline, its weights can be considered as parameters of the software pipeline.

Quantification of uncertainties. Apart from gradient-based optimization, linearization can also be used to estimate the probability distribution of output variables given the probability distribution of input variables, and to identify sub-procedures that amplify uncertainty.

The digital tracking calorimeter of the bergen pCT collaboration. In the remainder of this section, we summarize a recent survey [97] on the applicability of differentiable programming to the software pipeline of the Bergen pCT collaboration [326], displayed in Fig. 29.

The digital tracking calorimeter (DTC), which is designed and currently being built by the collaboration, consists of 43 parallel layers of 108 ALPIDE chips each and is to be placed behind the phantom. Protons leaving the phantom activate several pixels in each layer until they are stopped, as visualized in Fig. 30. The open-source package GATE [327], based on the GEANT4 toolkit [102–104], is used to simulate the passage of protons through both a model phantom with known RSP, and the DTC. Gate outputs the positions where protons hit ALPIDE chips, and the corresponding energy depositions, as floating-point numbers. These data are further processed into the discrete detector response, that only tells which ALPIDE pixels have been activated in each read-out cycle. A library of sample activation clusters depending on the deposited energy can be used here.

A numerical study showed that GATE as a part of the MC simulation sub-procedure is only piece-wise differentiable and has many discontinuities. It is as of yet unclear whether this is just a fixable implementation problem. In addition, the subsequent procedures in the pipeline could make it smooth by combining the MC results of many protons. As applying differentiable programming to GATE and its dependencies is a massive technical effort, either way a surrogate model should be used here first.

In the *proton history reconstruction* sub-procedure, the discrete detector output is mapped back to floating-point hit positions and energy depositions. This procedure involves many "discrete" operations such as:

- grouping neighboring pixels activated by the same proton into a *cluster*;
- assigning an energy deposition to this cluster according to its size, which is an integer;

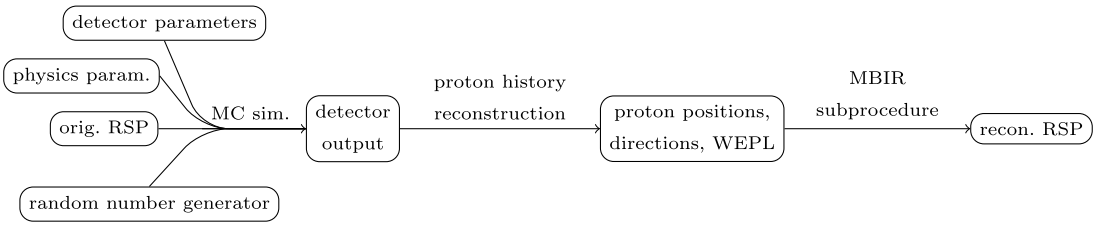


Fig. 29. Overview of the software pipeline of the Bergen pCT collaboration.
Source: Reproduced from Ref. [97].

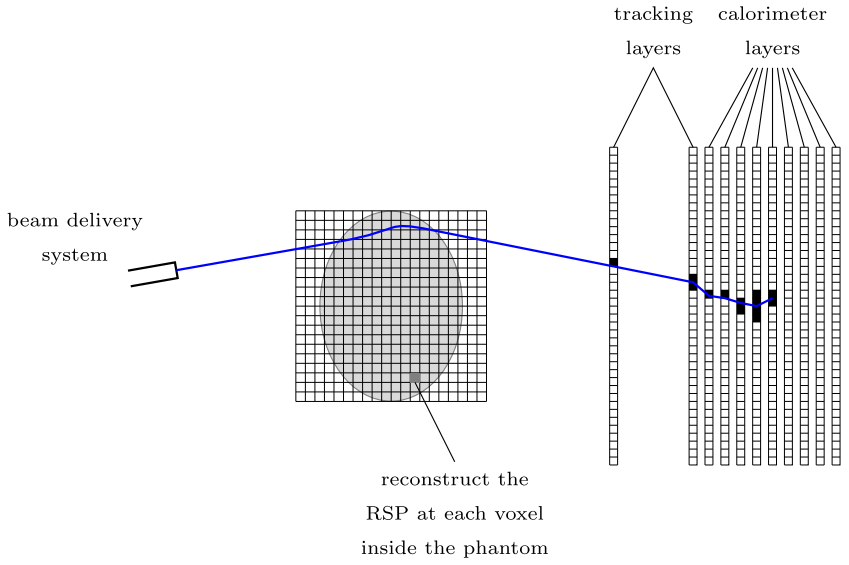


Fig. 30. Schematic figure of the scanning process with the digital tracking calorimeter of the Bergen pCT collaboration [326].
Source: Reproduced from Ref. [97].

- matching clusters likely related to the same proton into a *track*.

As the derivative of such discrete functions is either 0 or non-existent, we cannot benefit from differentiable programming here and need a surrogate model. The residual WEPL of the proton is estimated by a fit of the Bragg–Kleeman equation [328] to the estimated energy loss in each layer.

In the *model-based iterative reconstruction* sub-procedure, the matrix A in (28) is generated, and the linear system is solved iteratively. For typical setups, A is too large to be stored in memory [329], so its elements are regenerated whenever required.

Differentiating through the iterative solver in a "black-box" fashion is infeasible in reverse mode because of memory limits, and might suffer from a bad accuracy. This issue can probably be solved by exploiting the simple mathematical structure that allows to compute analytical derivatives.

4.5. Low-energy particle physics

Low-energy particle physics provides many unique research opportunities to search for exotic particle candidates or beyond Standard Model physics. They range from decay [330,331] to electric dipole moment (EDM) [332] and other measurements [333]. Specifically, tritium decay [334], neutron decay [335–337], neutron lifetime [338], and neutron EDM [339] produce key results. These experiments are high-precision measurements designed for specific purposes, leading to complex designs. These experiments have many tunable parameters, and we must design and operate them optimally to maintain continuous improvements of experimental results. This quality requirement highlights the importance of advanced methods based on differentiable programming for the field. Parameters for the optimization of such experiments are case-specific. However, we aim to set global and local parameters to improve measurement precision by reducing uncertainties. We may choose whether to do end-to-end optimization for local or global experimental parameters, depending on the complexity and effective dimensionality of the experiment. In some cases, differentiable programming is not ideal, as other methods achieve better convergence, such as Bayesian optimization [340] as mentioned in Section 4.1.1 and [203,206,341].

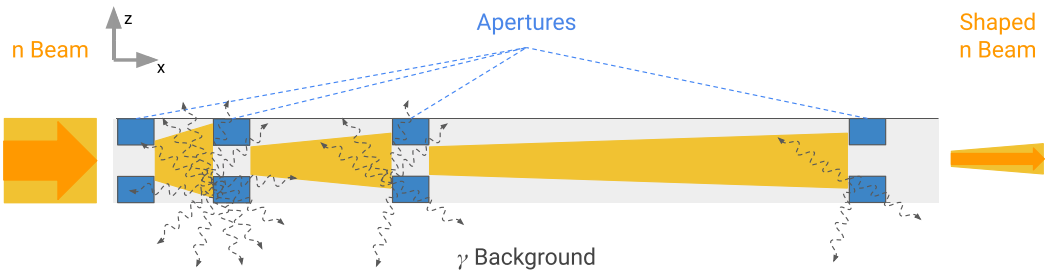


Fig. 31. Schematic of the neutron beam-line optimization for n absorbing quadratic apertures with position and width. Beam divergence and background are only drawn for illustration and not physically accurate.

To illustrate the capabilities of differentiable programming for low-energy particle physics, we choose the example of optimizing a neutron beam-line as used in [335,337] by tuning aperture placements and their width for desired beam characteristics. We highlight how we can customize and adapt the optimization pipeline of Section 3 to the problem. Consider a source of cold neutrons reaching an experiment through neutron guides and a velocity selector. The neutrons have a known wavelength, transverse momentum, and position distribution depending on the neutron guides and the velocity selector. We position a set of n quadratic neutron apertures between the velocity selector and the experiment to shape the resulting beam distribution, as shown in Fig. 31. The beam distribution can be calculated analytically by trigonometry and a set of integrals, requiring no surrogate model. Optimizing the beam distribution is essential to reduce systematic effects, maintain experiment confinements, or other constraints like costs. The two most significant systematic effects are beam homogeneity and created background signals by the beam-line through neutron absorption. We may encode the desired beam homogeneity or shape in a target distribution $P(x)$ with x being the distance perpendicular to the beam center. Therefore, we set the optimization objective as KL divergence or *relative entropy* D_{KL} [121] of $P(x)$ and the resulting beam distribution of the current setup $Q(x)$ for a fixed detector position on the beam axis.

$$D_{\mathrm{KL}}(Q \parallel P) = \sum_{x \in \mathcal{X}} Q(x) \log\left(\frac{Q(x)}{P(x)}\right).$$

Furthermore, we can expand the objective value with additional terms addressing different systematic effects. It is beneficial to place apertures further away from the experiment to minimize beam-line-induced background. We add the distance p_i of aperture i to the beam-line start as the first objective function adaption as

$$\mathcal{L}_1 = \frac{\alpha_1}{n} \sum_{i=1}^{n} p_n^2.$$

We also add aperture width w_i of aperture i as regularization

$$\mathcal{L}_2 = \frac{\alpha_2}{n} \sum_{i=1}^{n} w_n^2.$$

We use the weighting parameters α_1 and α_2 to tune the importance of each term. Therefore, the total optimization objective $\mathcal{L}$ for the differential programming pipeline is

$$\mathcal{L} = D_{\mathrm{KL}}(Q \parallel P) + \mathcal{L}_1 + \mathcal{L}_2 \,. \tag{29}$$

We keep the model general, so that it can be adapted to specific experiments. We present example results in Fig. 32 with different regularization results for a Gaussian target distribution. The *Julia* code and pipeline is available on *GitHub*[8] and uses the *ForwardDiff* package [342]. Beyond beam-line optimization, we can also optimize the geometry of scintillation energy detectors for optimal light transport, leading to better energy resolution and detector uniformity. Such detectors are often used for low-energy particle physics, and we propose using a differentiable programming optimization pipeline as described in Section 3 with a surrogate model from simulations to achieve optimal detector performance.

5. System architecture and requirements

A complete optimization of a model of any one of the example use cases discussed in Section 4, properly implemented in terms of a pipeline connecting all elements of the problem, would require a substantial investment in computing resources. However, such investment would render dedicated resources idle or obsolete once the optimization is complete. A good way to face such a time-peaking computing demand is to rely on a cloud infrastructure providing scalable and manageable computational resources. Before we discuss the details of such infrastructure, let us outline the main principles we consider relevant for the optimization system to run successfully.

[8] https://github.com/maxlampe/NobleAD.

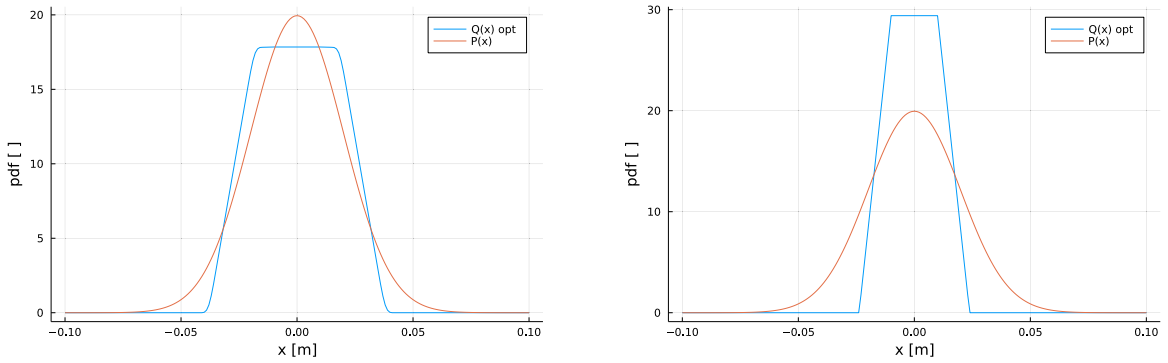


Fig. 32. Two different optimized beam-line distributions $Q(x)$ for $n = 3$ apertures. (a) uses only D_{KL} and $\mathcal{L}_2$, enforcing a good distribution approximation of $P(x)$ with equally sized apertures; (b) utilizes $\mathcal{L}$ as in Eq. (29), sacrificing function approximation quality for lower background levels.

5.1. Guiding principles

First of all, the developed pipeline should support a variety of simulation packages that are commonly used in various physics branches: *e.g.* GEANT4 [102], Pythia [343–345], Genie [346], to mention just a few. Each of the cited packages relies on other software components that should be pre-installed on the system to allow for its execution. Also, to support the execution of the main differential engine, the system would require different sets of software components depending on the task at hand. Secondly, the system should support different scales of devices under optimization: as we have seen *supra* these range from a coarse, high-level comparison of design options for a simple problem (*e.g.* the one of the MUonE detector, see *infra*, Section 4.1.6) to a detailed estimation of the performance of an LHC-scale detector entailing the simultaneous consideration of several hundred design parameters; in other words, given the breadth of the possible applications and their very different demands and scale, the usage of computing resources is going to be relatively small in some cases, while it can be enormous in others. Yet the scale of the selected optimization task should not be an issue implying specific attention to an end user.

An integrated end-to-end optimization system should run on many connected computing nodes in an automated fashion. Also, it is important to stress that running on regular cloud nodes might be quite expensive; the main commercial cloud providers offer so-called *spot* nodes, which may be interrupted at a very short notice at any time. We suggest our system should be capable of running on such nodes as well. Then, in order to meet computing power peak demands the system should be able to exploit "opportunistic" resources and volunteer computing if matching hardware requirements. Opportunistic resources are resources not dedicated to the project that can be temporarily made available to it, for example from a site of the collaboration. Volunteer computing indicates a type of distributed computing in which computer owners can make their computing resources available to research projects (see, for example, https://boinc.berkeley.edu/).

Finally, the system should support handling many user tasks simultaneously, by providing users with independent computational resources while allowing them to share their results. In a realistic scenario, a large optimization problem would be studied by several researchers who would test different parts of the developed pipeline independently, by freezing some optimization parameters in turn. The capability to run in parallel several of these partial tasks would enable a quicker convergence of the global task.

Below we provide a synthetic outline of the described principles:

- flexibility;
- distributed execution and scalability;
- interruption-friendly execution;
- user task independence & result sharing.

Of course, the above outline constitutes the final goal of a long development task, which must be approached in an incremental fashion.

5.2. System architecture outline

Following the guiding principles defined *supra*, the system should be merged into a cloud infrastructure ideally supporting all the major cloud providers (AWS, GCP, Azure, etc.). Today's state-of-the-art virtual nodes management system is Kubernetes [347], which allows flexible setup, execution, and monitoring of different tasks on various virtual nodes comprising an elastic cluster. Such a cluster may be populated by a few nodes (*e.g.,* a single node). Depending on the computing demands, it extends to a maximum number of nodes configured by the administrator. Also, it enables the creation of software environments (containers) that include all the packages required for specific software to run. In order to support both independent task execution and spot instances handling,

the system should have storage, computing and code decoupled at the design time. Storage should incorporate both persistent structured data like a relational database for different run results and unstructured data volumes for handling temporary files and intermediate results. Computing resources will be provided with the help of Kubernetes and nodes of the cluster. The code for the computations of individual tasks should be aware of the storage capacities available to it as well as of the computational resources it can use. Part of the system responsible for the overall execution should ensure the availability of storage components, individual task tracking, and result sharing components.

5.3. Infrastructure requirements

The optimization system relies on a few functionalities from the cloud provider. We list them below.

- API for Kubernetes cluster access/management/container execution;
- Ability for Kubernetes cluster to run on spot instances;
- Kubernetes cluster monitoring;
- Support for structured DB setup and configuration;
- Support for unstructured persistent volumes setup and management;
- Billing API for monitoring and controlling of computational budget.

5.4. Hardware requirements

Depending on the optimization task and its software computations packages different hardware resources should be provided. Thus, the system should run on relatively capable virtual nodes (24 CPUs, 128 GB of RAM). However, some of the computational tasks might require additional hardware resources like GPU/TPU cards or extremely large RAM volumes. For such dedicated tasks the system should be able to instantiate a separate computational Kubernetes clusters.

5.5. Main software components

The optimization system should include the following main software components:

- structural storage (database) management;
- volume storage management;
- cloud compute management;
- simulation package connection and software environment configuration;
- user task management;
- optimization monitoring/benchmarking interface;
- black-box optimization runtime (Python, Julia, etc.);
- differentiable optimization runtime (PyTorch [52], Tensorflow [53], JAX [94]).

The above is a non-exhaustive list, and is only meant to offer a view of the typical deployment needs of the system we imagine.

5.6. Integration requirements

Finally, we believe that the system should also provide interfaces to ease the integration of the following:

- cloud provider-specific Kubernetes interfaces, storage resources and computing resources configuration;
- new/custom physics simulators;
- external benchmarking services like Weights-and-biases or Comet.ml.

6. Conclusions

The history of particle detection is over 100 years old, and it is full of breakthroughs and paradigm-changing inventions which often ingeniously exploited technological advancements conceived for other applications to improve the performance of apparatus. In this document we argue that the latest advancements in differentiable programming open the field to breakthrough advancements in particle detector development is the rise of differentiable programming. Coupled to the large computing power available today, the automatic calculation of derivatives of complex functions and computer code offers the possibility of scanning the high-dimensional space of design solutions in a systematic search of the global maximum of a carefully defined utility function. The potential gains—in terms of performance, decreased spending, or others—of a global model of the system under design are very large, and they come from the possibility of the optimization task to re-align the design goal to the specific choices that a designer is faced with when picking a value of the many free parameters of the system, once all external constraints (*e.g.,* total cost, time, space) are established.

In this white paper we have discussed the core idea for the AI-assisted design of scientific instruments for fundamental research powered by gradient descent techniques implemented using the differentiable programming paradigma. We outline a series of fields

and concrete applications which, by considering and seeking a solution to a wide range of use cases of complexity varying from easy to very hard, may endow our community with the technology needed to attack detector design problems still harder and of larger scale. While automated optimization systems cannot at the present time have the ambition to substitute the hand and the intuition of the expert, they cannot any longer be ignored as assistants to the design task. Ultimately, we believe that their careful specification and systematic use may allow us to discover entirely new, groundbreaking solutions to our century-old problems, furthering the happy partnership of progress in technology and pure research we have witnessed until today.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Pietro Vischia reports financial support was provided by IRIS-HEP. Pietro Vischia reports financial support was provided by JENAA. Tommaso Dorigo reports financial support was provided by European Commission. Pietro Vischia, Andrea Giammanco reports financial support was provided by Fund for Scientfic Research. Christan Krause reports financial support was provided by US Department of Energy. Max Aehle, Nicolas Gauger, Ralf Keidel reports financial support was provided by German federal state of Rhineland-Palatnate. Lukas Heinrich reports financial support was provided by Deutsche Forschungsgemeinschaft. Alberto Ramos reports financial support was provided by Government of Valencia. Alberto Ramos reports financial support was provided by Spain Ministry of Science and Innovation. Haitham Zaraket reports travel was provided by Erasmus Plus. Andrea Giammanco, Lukas Layer, Nathan Simpson reports financial support was provided by European Commission. The first author (Tommaso Dorigo) is also an editor of Reviews in Physics. As such he asks this paper to be handled by one of the other editors.

Acknowledgments

We wish to thank all the participants of the first MODE workshop on differentiable programming that took place in Louvain-la-Neuve (Belgium) from 6 to 8 September 2021, for the fruitful discussions that influenced the content of this document.

We gratefully acknowledge support by IRIS-HEP (Institute for Research and Innovation in Software for High Energy Physics, National Science Foundation grant OAC-1836650, https://iris-hep.org/) and JENAA (Joint ECFA-NuPECC-APPEC Activities, http://www.nupecc.org/jenaa/). A. Giammanco's work was partially supported by the EU Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie Grant Agreement No. 822185 ("INTENSE") and the Research and Innovation Action for Security Grant Agreement No. 101021812 ("SilentBorder"), and by the Fonds de la Recherche Scientifique - FNRS under Grants No. T.0099.19 and J.0070.21. P. Vischia's work was supported by the FNRS under the Grant No. 40000963 and by the Ramón *y* Cajal program under the Project No. RYC2021-033305-I. C. Krause's work is supported by DOE grant DOE-SC0010008. M. Aehle, N. Gauger, and R. Keidel gratefully acknowledge the funding of the research training group SIVERT by the German federal state of Rhineland-Palatinate. The work of T. Dorigo, L. Layer and N. Simpson is supported by a Marie Sklodowska-Curie Innovative Training Network Fellowship of the European Commissions Horizon 2020 Programme under Contract Number 765710 INSIGHTS. L. Heinrich and M. Lamparth are supported by the Excellence Cluster ORIGINS, which is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC-2094-390783311. Alberto Ramos acknowledges financial support from the Generalitat Valenciana (genT program CIDEGENT/2019/040) and the Ministerio de Ciencia e Innovacion (PID2020-113644GB-I00). H. Zaraket would like to thank the Erasmus Plus mobility program. Fig. 23 (bottom) has been adapted from Ref. [305] thanks to the courtesy of László Oláh. We would also like to thank the International Muon Collider Collaboration for providing us with the data to produce Figs. 14 and 15.

References

[1] O. Sigmund, K. Maute, Topology optimization approaches, Struct. Multidiscip. Optim. 48 (2013) 1031–1055, http://dx.doi.org/10.1007/s00158-013-0978-6.

[2] P. Vischia, Reporting results in high energy physics publications: A manifesto, Rev. Phys. 5 (2020) 100046, http://dx.doi.org/10.1016/j.revip.2020.100046, URL https://www.sciencedirect.com/science/article/pii/S2405428320300095.
[3] CMS Collaboration, CMS Physics: Technical Design Report Volume 1: Detector Performance and Software, Technical design report, CMS, CERN, Geneva, 2006, URL https://cds.cern.ch/record/922757.
[4] K. Cranmer, J. Brehmer, G. Louppe, The frontier of simulation-based inference, Proc. Natl. Acad. Sci. 117 (48) (2020) 30055–30062, http://dx.doi.org/10.1073/pnas.1912789117.
[5] A. Baydin, K. Cranmer, P. de Castro Manzano, C. Delaere, D. Derkach, J. Donini, T. Dorigo, A. Giammanco, J. Kieseler, L. Layer, G. Louppe, F. Ratnikov, G. Strong, M. Tosi, A. Ustyuzhanin, P. Vischia, H. Yarar, Toward machine learning optimization of experimental design, Nucl. Phys. News 31 (1) (2021) 25–28, http://dx.doi.org/10.1080/10619127.2021.1881364.
[6] Y. Mishnayot, M. Layani, I. Cooperstein, S. Magdassi, G. Ron, Three-dimensional printing of scintillating materials, Rev. Sci. Instrum. 85 (2014) 085102, http://dx.doi.org/10.1063/1.4891703, arXiv:1406.4817.
[7] G. Giacomini, W. Chen, G. D'Amen, A. Tricoli, Fabrication and performance of AC-coupled LGADs, JINST 14 (09) (2019) P09004, http://dx.doi.org/10.1088/1748-0221/14/09/p09004, arXiv:1906.11542.
[8] L. Evans, P. Bryant, LHC machine, J. Instrum. 3 (08) (2008) S08001, http://dx.doi.org/10.1088/1748-0221/3/08/s08001.
[9] S. Farrell, et al., Particle track reconstruction with deep learning, in: Proceedings of the Deep Learning for Physical Sciences Workshop at NIPS (2017), 2017.
[10] S. Farrell, et al., Novel deep learning methods for track reconstruction, in: 4th International Workshop Connecting the Dots 2018, 2018, arXiv:1810.06111.
[11] S. Amrouche, et al., The tracking machine learning challenge: Accuracy phase, in: The NeurIPS '18 Competition, Springer International Publishing, 2020, p. 231, http://dx.doi.org/10.1007/978-3-030-29135-8_9.
[12] X. Ju, et al., Graph neural networks for particle reconstruction in high energy physics detectors, in: 33rd Annual Conference on Neural Information Processing Systems, 2020, arXiv:2003.11603.
[13] S. Akar, et al., An updated hybrid deep learning algorithm for identifying and locating primary vertices, 2020, arXiv:2007.01023.
[14] J. Shlomi, et al., Secondary vertex finding in jets with neural networks, Eur. Phys. J. C 81 (2021) 540, http://dx.doi.org/10.1140/epjc/s10052-021-09342-y.
[15] N. Choma, et al., Track seeding and labelling with embedded-space graph neural networks, 2020, arXiv:2007.00149.
[16] F. Siviero, et al., First application of machine learning algorithms to the position reconstruction in resistive silicon detectors, J. Instrum. 16 (2021) P03019, http://dx.doi.org/10.1088/1748-0221/16/03/p03019.
[17] P. Fox, S. Huang, J. Isaacson, X. Ju, B. Nachman, Beyond 4D tracking: Using cluster shapes for track seeding, JINST 16 (05) (2021) P05001, http://dx.doi.org/10.1088/1748-0221/16/05/P05001, arXiv:2012.04533.
[18] S. Amrouche, M. Kiehn, T. Golling, A. Salzburger, Hashing and metric learning for charged particle tracking, in: 33rd Annual Conference on Neural Information Processing Systems, 2021, arXiv:2101.06428.
[19] G. Kiichi, et al., Development of a vertex finding algorithm using recurrent neural network, 2021, arXiv:2101.11906.
[20] C. Biscarat, S. Caillou, C. Rougier, J. Stark, J. Zahreddine, Towards a realistic track reconstruction algorithm based on graph neural networks for the HL-LHC, in: 25th International Conference on Computing in High-Energy and Nuclear Physics, 2021, arXiv:2103.00916.
[21] S. Akar, et al., Progress in developing a hybrid deep learning algorithm for identifying and locating primary vertices, EPJ Web Conf. 251 (2021) 04012, http://dx.doi.org/10.1051/epjconf/202125104012.
[22] S. Thais, G. DeZoort, Instance segmentation GNNs for one-shot conformal tracking at the LHC, 2021, arXiv:2103.06509.
[23] X. Ju, et al., Performance of a geometric deep learning pipeline for HL-LHC particle tracking, Eur. Phys. J. C 81 (2021) 876.
[24] G. Dezoort, et al., Charged particle tracking via edge-classifying interaction networks, 2021, arXiv:2103.16701.
[25] A. Edmonds, D. Brown, L. Vinas, S. Pagan, Using machine learning to select high-quality measurements, J. Instrum. 16 (2021) T08010, http://dx.doi.org/10.1088/1748-0221/16/08/t08010.
[26] E. Lavrik, M. Shiroya, H. Schmidt, A. Toia, J. Heuser, Optical inspection of the silicon micro-strip sensors for the CBM experiment employing artificial intelligence, Nucl. Instrum. Methods A 1021 (2022) 165932, http://dx.doi.org/10.1016/j.nima.2021.165932.
[27] B. Huth, A. Salzburger, T. Wettig, Machine learning for surface prediction in ACTS, in: 25th International Conference on Computing in High-Energy and Nuclear Physics, 2021, arXiv:2108.03068.
[28] P. Goncharov, et al., Ariadne: PyTorch library for particle track reconstruction using deep learning, in: 24th International Scientific Conference of Young Scientists and Specialists, 2021, arXiv:2109.08982.
[29] A. Lazar, et al., Accelerating the inference of the Exa.TrkX pipeline, 2022, arXiv:2202.06929.
[30] M. Benedikt, A. Blondel, P. Janot, M. Mangano, F. Zimmermann, Future circular colliders succeeding the LHC, Nat. Phys. 16 (4) (2020) 402–407, http://dx.doi.org/10.1038/s41567-020-0856-2.
[31] D. Hernandez, T. Brown, Measuring the algorithmic efficiency of neural networks, 2020, arXiv e-prints arXiv:2005.04305.
[32] J. Kieseler, Object condensation: one-stage grid-free multi-object reconstruction in physics detectors, graph and image data, Eur. Phys. J. C 80 (9) (2020) 886, http://dx.doi.org/10.1140/epjc/s10052-020-08461-2, arXiv:2002.03605.
[33] R. Hicks, E. Murman, G. Vanderplaats, An Assessment of Airfoil Design by Numerical Optimization, Tech. rep., NASA Ames Research Center, 1974.
[34] R. Hicks, P. Henne, Wing design by numerical optimization, J. Aircr. 15 (7) (1978) 407–412, http://dx.doi.org/10.2514/3.58379.
[35] O. Pironneau, On optimum design in fluid mechanics, J. Fluid Mech. 64 (1) (1974) 97–110, http://dx.doi.org/10.1017/S0022112074002023.
[36] O. Pironneau, On optimum profiles in stokes flow, J. Fluid Mech. 59 (1) (1973) 117–128, http://dx.doi.org/10.1017/S002211207300145X.
[37] A. Jameson, Aerodynamic design via control theory, J. Sci. Comput. 3 (3) (1988) 233–260, http://dx.doi.org/10.1007/BF01061285.
[38] M. Towara, U. Naumann, A discrete adjoint model for OpenFOAM, Procedia Comput. Sci. 18 (2013) 429–438, http://dx.doi.org/10.1016/j.procs.2013.05.206.
[39] T. Albring, M. Sagebaum, N. Gauger, Efficient Aerodynamic Design using the Discrete Adjoint Method in SU2, AIAA 2016-3518, 2016.
[40] M. Luers, et al., Adjoint-based volumetric shape optimization of turbine blades, in: 2018 Multidisciplinary Analysis and Optimization Conference, American Institute of Aeronautics and Astronautics, 2018, http://dx.doi.org/10.2514/6.2018-3638.
[41] A. Nemili, E. Özkaya, N. Gauger, F. Kramer, F. Thiele, Accurate discrete adjoint approach for optimal active separation control, AIAA J. 55 (9) (2017) 3016–3026, http://dx.doi.org/10.2514/1.J055009.
[42] B. Zhou, S. Ryong Koh, N. Gauger, M. Meinke, W. Schöder, A discrete adjoint framework for trailing-edge noise minimization via porous material, Comput. & Fluids 172 (2018) 97–108, http://dx.doi.org/10.1016/j.compfluid.2018.06.017.
[43] R. Bombardieri, R. Cavallaro, R. Sanchez, N. Gauger, Aerostructural wing shape optimization assisted by algorithmic differentiation, Struct. Multidiscip. Optim. 64 (2) (2021) 739–760, http://dx.doi.org/10.1007/s00158-021-02884-5.
[44] M. Morlighem, D. Goldberg, T. Dias dos Santos, J. Lee, M. Sagebaum, Mapping the sensitivity of the Amundsen sea embayment to changes in external forcings using automatic differentiation, Geophys. Res. Lett. 48 (23) (2021) http://dx.doi.org/10.1029/2021GL095440.
[45] J. Andersson, J. Åkesson, M. Diehl, Casadi: A symbolic package for automatic differentiation and optimal control, in: Recent Advances in Algorithmic Differentiation, Springer, 2012, pp. 297–307.
[46] Y. Achdou, O. Pironneau, Computational Methods for Option Pricing, Society for Industrial and Applied Mathematics, 2005, http://dx.doi.org/10.1137/1.9780898717495.

[47] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.
[48] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[49] J. Janai, et al., Computer vision for autonomous vehicles: Problems, datasets and state of the art, Found. Trends Comput. Graph. Vis. 12 (1–3) (2020) 1–308.
[50] Y. Goldberg, Neural network methods for natural language processing, Synth. Lect. Hum. Lang. Technol. 10 (1) (2017) 1–309.
[51] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.
[52] A. Paszke, et al., Pytorch: An imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035.
[53] M. Abadi, et al., {TensorFlow}: A system for {Large-Scale} machine learning, in: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 2016, pp. 265–283.
[54] A. Baydin, B. Pearlmutter, A. Radul, J. Siskind, Automatic differentiation in machine learning: A survey, J. Mach. Learn. Res. 18 (1) (2017) 5595–5637.
[55] C. Olah, Neural networks, types, and functional programming, 2015, retrieved on 17-03-2022. URL http://colah.github.io/posts/2015-09-NN-Types-FP/.
[56] D. Dalrymple, Differentiable programming as the 2016 most important recent scientific news, 2016, retrieved on 17-03-2022. URL https://www.edge.org/response-detail/26794.
[57] Y. LeCun, Facebook post on differentiable programming, 2018, retrieved on 17-03-2022. URL https://www.facebook.com/yann.lecun/posts/10155003011462143.
[58] S. Shirobokov, V. Belavin, M. Kagan, A. Ustyuzhanin, A. Baydin, Black-box optimization with local generative surrogates, 2020, arXiv:2002.04632.
[59] T. Dorigo, Geometry optimization of a muon-electron scattering detector, Phys. Open 4 (2020) 100022.
[60] F. Ratnikov, Using machine learning to speed up and improve calorimeter R & D, J. Instrum. 15 (05) (2020) C05032.
[61] E. Cisbani, et al., AI-optimized detector design for the future Electron-Ion Collider: the dual-radiator RICH case, J. Instrum. 15 (05) (2020) P05009.
[62] A. Edelen, et al., Machine learning for orders of magnitude speedup in multiobjective optimization of particle accelerator systems, Phys. Rev. Accel. Beams 23 (4) (2020) 044601, http://dx.doi.org/10.1103/PhysRevAccelBeams.23.044601, publisher: American Physical Society.
[63] D. Koser, et al., Input beam matching and beam dynamics design optimization of the IsoDAR RFQ using statistical and machine learning techniques, Front. Phys. (2021) arXiv:2112.02579 [physics]. submitted for publication. arXiv:2112.02579.
[64] F. Van Der Veken, et al., Machine learning in accelerator physics: applications at the CERN Large Hadron Collider, in: Proceedings of Artificial Intelligence for Science, Industry and Society PoS(AISIS2019), Vol. 372, SISSA Medialab, 2020, p. 044.
[65] S. Meyer, et al., Optimization and performance study of a proton CT system for pre-clinical small animal imaging, Phys. Med. Biol. 65 (15) (2020) 155008, http://dx.doi.org/10.1088/1361-6560/ab8.
[66] D. Shanno, Conditioning of quasi-newton methods for function minimization, Math. Comp. 24 (111) (1970) 647–656, http://dx.doi.org/10.1090/S0025-5718-1970-0274029-X.
[67] D. Goldfarb, A family of variable-metric methods derived by variational means, Math. Comp. 24 (109) (1970) 23–26, http://dx.doi.org/10.1090/S0025-5718-1970-0258249-6.
[68] R. Fletcher, A new approach to variable metric algorithms, Comput. J. 13 (3) (1970) 317–322, http://dx.doi.org/10.1093/comjnl/13.3.317.
[69] R. Byrd, P. Lu, J. Nocedal, C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM J. Sci. Comput. 16 (5) (1995) 1190–1208, http://dx.doi.org/10.1137/0916069.
[70] C. Zhu, R. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-BFGS-b: Fortran subroutines for large-scale bound-constrained optimization, ACM Trans. Math. Softw. 23 (4) (1997) 550–560, http://dx.doi.org/10.1145/279232.279236.
[71] M. Belkin, Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation, 2021, arXiv:2105.14368.
[72] J. Hadamard, Mémoire sur le Problème d'Analyse Relatif à l'Équilibre des Plaques Élastiques Encastrées, Mémoires Présentés par Divers Savants à l'Académie des Sciences de l'Institut de France: Éxtrait, Imprimerie nationale, 1908.
[73] R. Vidal, J. Bruna, R. Giryes, S. Soatto, Mathematics of deep learning, 2017, arXiv:1712.04741.
[74] P. Virtanen, et al., SciPy 1.0: Fundamental algorithms for scientific computing in Python, Nature Methods 17 (2020) 261–272, http://dx.doi.org/10.1038/s41592-019-0686-2.
[75] R. Burden, J. Faires, A. Burden, Numerical Analysis, Cengage Learning, 2015.
[76] A. Griewank, A. Walther, Evaluating Derivatives, Other Titles in Applied Mathematics, Society for Industrial and Applied Mathematics, 2008, http://dx.doi.org/10.1137/1.9780898717761.
[77] R. Wengert, A simple automatic derivative evaluation program, Commun. ACM 7 (8) (1964) 463–464.
[78] S. Linnainmaa, The Representation of the Cumulative Rounding Error of an Algorithm as a Taylor Expansion of the Local Rounding Errors (Master's Thesis), Univ. Helsinki, 1970, (in Finnish).
[79] B. Speelpenning, Compiling Fast Partial Derivatives of Functions Given by Algorithms, University of Illinois at Urbana-Champaign, 1980.
[80] D. Rumelhart, G. Hinton, R. Williams, Learning Internal Representations by Error Propagation, Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
[81] A. Walther, A. Griewank, Getting started with ADOL-C, in: U. Naumann, O. Schenk (Eds.), Combinatorial Scientific Computing, in: Chapman-Hall CRC Computational Science, 2012, pp. 181–202, Ch. 7.
[82] R. Hogan, Fast reverse-mode automatic differentiation using expression templates in C++, ACM Trans. Math. Softw. 40 (4) (2014) 26:1–26:24, http://dx.doi.org/10.1145/2560359.
[83] J. Lotz, Hybrid Approaches to Adjoint Code Generation with DCO/C++, Department of Computer Science, RWTH Aachen University, 2016, URL http://publications.rwth-aachen.de/record/667318.
[84] M. Sagebaum, T. Albring, N. Gauger, High-performance derivative computations using codipack, ACM Trans. Math. Softw. 45 (4) (2019) http://dx.doi.org/10.1145/3356900.
[85] V. Vassilev, M. Vassilev, A. Penev, L. Moneta, V. Ilieva, Clad — automatic differentiation using clang and LLVM, J. Phys. Conf. Ser. 608 (2015) 012055, http://dx.doi.org/10.1088/1742-6596/608/1/012055.
[86] W. Moses, V. Churavy, Instead of rewriting foreign code for machine learning, automatically synthesize fast gradients, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 12472–12485.
[87] C. Bischof, L. Roh, A. Mauer, ADIC — An extensible automatic differentiation tool for ANSI-C, Softw.–Pract. Exp. 27 (12) (1997) 1427–1456, http://dx.doi.org/10.1002/(SICI)1097-024X(199712)27:12<1427::AID-SPE138>3.0.CO;2-Q.
[88] C. Bischof, A. Carle, P. Khademi, A. Mauer, ADIFOR 2.0: Automatic differentiation of Fortran 77 programs, IEEE Comput. Sci. Eng. 3 (3) (1996) 18–32.
[89] J. Utke, et al., Openad/f: A modular open-source tool for automatic differentiation of fortran codes, ACM Trans. Math. Software 34 (4) (2008) http://dx.doi.org/10.1145/1377596.1377598.
[90] L. Hascoet, V. Pascual, The tapenade automatic differentiation tool: Principles, model, and specification, ACM Trans. Math. Softw. 39 (3) (2013) 1–43, http://dx.doi.org/10.1145/2450153.2450158.
[91] M. Bücker, F. f. Schiller, Community portal for automatic differentiation, 2000, retrieved on 17-03-2022. URL http://www.autodiff.org.
[92] L. Heinrich, M. Kagan, Differentiable matrix elements with MadJax, in: 20th Intern. Workshop on Adv. Computing and Analysis Techniques in Phys. Res., 2022, arXiv:2203.00057.

[93] J. Alwall, et al., The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations, J. High Energy Phys. 07 (2014) 079, http://dx.doi.org/10.1007/JHEP07(2014)079, arXiv:1405.0301.

[94] J. Bradbury, et al., JAX: composable transformations of Python+NumPy programs, 2018, http://github.com/google/jax, version 0.2.5.

[95] B. Dauvergne, L. Hascoët, The data-flow equations of checkpointing in reverse automatic differentiation, in: V. Alexandrov, G. van Albada, P. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Vol. 3994, Springer Berlin Heidelberg, 2006, pp. 566–573, http://dx.doi.org/10.1007/11758549_78.

[96] J. Blühdorn, M. Sagebaum, N. Gauger, Event-based automatic differentiation of OpenMP with OpDiLib, 2021, arXiv:2102.11572 [cs]. arXiv:2102.11572.

[97] M. Aehle, et al., Derivatives in proton CT, 2022, arXiv:2202.05551.

[98] A. Adelmann, et al., New directions for surrogate models and differentiable programming for High Energy Physics detector simulation, in: 2022 Snowmass Summer Study, 2022, arXiv:2203.08806.

[99] M. Kasim, D. Watson-Parris, L. Deaconu, S. Oliver, P. Hatfield, D. Froula, G. Gregori, M. Jarvis, S. Khatiwala, J. Korenaga, et al., Up to two billion times acceleration of scientific simulations with deep neural architecture search, in: APS Division of Plasma Physics Meeting Abstracts, Vol. 2020, 2020, pp. BO05–001.

[100] J. Conway, Incorporating nuisance parameters in likelihoods for multisource spectra, 2011, pp. 115–120, http://dx.doi.org/10.5170/CERN-2011-006.115, arXiv:1103.0354. URL http://cds.cern.ch/record/1333496.

[101] T. Dorigo, P. de Castro Manzano, Dealing with nuisance parameters using machine learning in high energy physics: a review, in: Artificial Intelligence for High Energy Physics, World Scientific, 2022.

[102] S. Agostinelli, et al., GEANT4 — a simulation toolkit, Nucl. Instrum. Methods A 506 (2003) 250, http://dx.doi.org/10.1016/S0168-9002(03)01368-8.

[103] J. Allison, et al., Geant4 developments and applications, IEEE Trans. Nucl. Sci. 53 (1) (2006) 270–278, http://dx.doi.org/10.1109/TNS.2006.869826.

[104] J. Allison, et al., Recent developments in Geant4, Nucl. Instrum. Methods A 835 (2016) 186–225, http://dx.doi.org/10.1016/j.nima.2016.06.125.

[105] T. Böhlen, et al., The FLUKA code: developments and challenges for high energy and medical applications, Nucl. Data Sheets 120 (2014) 211, http://dx.doi.org/10.1016/j.nds.2014.07.049.

[106] J. de Favereau, et al., DELPHES 3, A modular framework for fast simulation of a generic collider experiment, J. High Energy Phys. 02 (2014) 057, http://dx.doi.org/10.1007/JHEP02(2014)057, arXiv:1307.6346.

[107] A. Sinha, et al., SUPA: A lightweight diagnostic simulator for machine learning in particle physics, 2022, arXiv:2202.05012.

[108] D. Belayneh, et al., Calorimetry with deep learning: particle simulation and reconstruction for collider physics, Eur. Phys. J. C 80 (7) (2020) 688, http://dx.doi.org/10.1140/epjc/s10052-020-8251-9, arXiv:1912.06794.

[109] E. Buhmann, et al., Getting high: High fidelity simulation of high granularity calorimeters with high speed, Comput. Softw. Big Sci. 5 (1) (2021) 13, http://dx.doi.org/10.1007/s41781-021-00056-0, arXiv:2005.05334.

[110] N. Zheng, P. Mazumder, Learning in Energy-Efficient Neuromorphic Computing: Algorithm and Architecture Co-Design, Wiley, 2019.

[111] M. Schuld, F. Petruccione, Machine Learning with Quantum Computers, Springer, 2021.

[112] P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, D. Whiteson, Parameterized neural networks for high-energy physics, Eur. Phys. J. C 76 (5) (2016) 235, http://dx.doi.org/10.1140/epjc/s10052-016-4099-4, arXiv:1601.07913.

[113] D. Kingma, M. Welling, Auto-encoding variational Bayes, 2013, arXiv e-prints arXiv:1312.6114.

[114] D. Jimenez Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, 2014, arXiv e-prints arXiv:1401.4082.

[115] D. Kingma, M. Welling, An introduction to variational autoencoders, 2019, arXiv e-prints arXiv:1906.02691.

[116] I. Goodfellow, et al., Generative adversarial networks, 2014, arXiv e-prints arXiv:1406.2661.

[117] J. Gui, Z. Sun, Y. Wen, D. Tao, J. Ye, A review on generative adversarial networks: Algorithms, theory, and applications, 2020, arXiv e-prints arXiv:2001.06937.

[118] D. Jimenez Rezende, S. Mohamed, Variational inference with normalizing flows, 2015, arXiv e-prints arXiv:1505.05770.

[119] I. Kobyzev, S. Prince, M. Brubaker, Normalizing flows: An introduction and review of current methods, 2019, arXiv e-prints arXiv:1908.09257.

[120] G. Papamakarios, E. Nalisnick, D. Jimenez Rezende, S. Mohamed, B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference, 2019, arXiv e-prints arXiv:1912.02762.

[121] S. Kullback, R. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86.

[122] ATLAS Collaboration, Deep Generative Models for Fast Shower Simulation in ATLAS, Tech. Rep. ATL-SOFT-PUB-2018-001, CERN, 2018, URL http://cds.cern.ch/record/2630433.

[123] K. Deja, J. Dubiński, P. Nowak, S. Wenzel, T. Trzciński, End-to-end sinkhorn autoencoder with noise generator, 2020, arXiv e-prints arXiv:2006.06704.

[124] E. Buhmann, et al., Decoding photons: Physics in the latent space of a BIB-AE generative network, EPJ Web Conf. 251 (2021) 03003, http://dx.doi.org/10.1051/epjconf/202125103003, arXiv:2102.12491.

[125] J. Howard, S. Mandt, D. Whiteson, Y. Yang, Foundations of a fast, data-driven, machine-learned simulator, 2021, arXiv:2101.08944.

[126] E. Buhmann, et al., Hadrons, better, faster, stronger, 2021, arXiv:2112.09709.

[127] A. Hariri, D. Dyachkova, S. Gleyzer, Graph generative models for fast detector simulations in high energy physics, 2021, arXiv:2104.01725.

[128] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, 2017, arXiv e-prints arXiv:1701.07875.

[129] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of wasserstein GANs, 2017, arXiv e-prints arXiv:1704.00028.

[130] A. Borji, Pros and cons of GAN evaluation measures, 2018, arXiv e-prints arXiv:1802.03446.

[131] L. de Oliveira, M. Paganini, B. Nachman, Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis, Comput. Softw. Big Sci. 1 (1) (2017) 4, http://dx.doi.org/10.1007/s41781-017-0004-6, arXiv:1701.05927.

[132] M. Paganini, L. de Oliveira, B. Nachman, Accelerating science with generative adversarial networks: An application to 3D particle showers in multilayer calorimeters, Phys. Rev. Lett. 120 (4) (2018) 042003, http://dx.doi.org/10.1103/PhysRevLett.120.042003, arXiv:1705.02355.

[133] M. Paganini, L. de Oliveira, B. Nachman, CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks, Phys. Rev. D 97 (1) (2018) 014021, http://dx.doi.org/10.1103/PhysRevD.97.014021, arXiv:1712.10321.

[134] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn, R. Winterhalder, How to GAN away detector effects, SciPost Phys. 8 (4) (2020) 070, http://dx.doi.org/10.21468/SciPostPhys.8.4.070, arXiv:1912.00477.

[135] S. Vallecorsa, F. Carminati, G. Khattak, 3D convolutional GAN for fast simulation, EPJ Web Conf. 214 (2019) 02010, http://dx.doi.org/10.1051/epjconf/201921402010.

[136] C. Ahdida, et al., Fast simulation of muons produced at the SHiP experiment using Generative Adversarial Networks, JINST 14 (2019) P11028, http://dx.doi.org/10.1088/1748-0221/14/11/P11028, arXiv:1909.04451.

[137] V. Chekalina, et al., Generative models for fast calorimeter simulation. LHCb case, in: CHEP 2018, 2018, http://dx.doi.org/10.1051/epjconf/201921402034, arXiv:1812.01319.

[138] F. Carminati, A. Gheata, G. Khattak, M.L.P.S. Sharan, S. Vallecorsa, Three dimensional generative adversarial networks for fast simulation, J. Phys. Conf. Ser. 1085 (3) (2018) 032016, http://dx.doi.org/10.1088/1742-6596/1085/3/032016.

[139] S. Vallecorsa, Generative models for fast simulation, J. Phys. Conf. Ser. 1085 (2) (2018) 022005, http://dx.doi.org/10.1088/1742-6596/1085/2/022005.

[140] P. Musella, F. Pandolfi, Fast and accurate simulation of particle detectors using generative adversarial networks, Comput. Softw. Big Sci. 2 (1) (2018) 8, http://dx.doi.org/10.1007/s41781-018-0015-y, arXiv:1805.00850.

[141] M. Erdmann, L. Geiger, J. Glombitza, D. Schmidt, Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks, Comput. Softw. Big Sci. 2 (1) (2018) 4, http://dx.doi.org/10.1007/s41781-018-0008-x, arXiv:1802.03325.

[142] K. Deja, T. Trzcinski, L. Graczykowski, Generative models for fast cluster simulations in the TPC for the ALICE experiment, EPJ Web Conf. 214 (2019) 06003, http://dx.doi.org/10.1051/epjconf/201921406003.

[143] D. Derkach, N. Kazeev, F. Ratnikov, A. Ustyuzhanin, A. Volokhova, Cherenkov detectors fast simulation using neural networks, Nucl. Instrum. Methods A 952 (2020) 161804, http://dx.doi.org/10.1016/j.nima.2019.01.031, arXiv:1903.11788.

[144] M. Erdmann, J. Glombitza, T. Quast, Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network, Comput. Softw. Big Sci. 3 (1) (2019) 4, http://dx.doi.org/10.1007/s41781-018-0019-7, arXiv:1807.01954.

[145] L. de Oliveira, M. Paganini, B. Nachman, Tips and tricks for training GANs with physics constraints, in: Proceedings of the Deep Learning for Physical Sciences Workshop at NIPS (2017), 2017.

[146] L. de Oliveira, M. Paganini, B. Nachman, Controlling physical attributes in GAN-accelerated simulation of electromagnetic calorimeters, J. Phys. Conf. Ser. 1085 (4) (2018) 042017, http://dx.doi.org/10.1088/1742-6596/1085/4/042017, arXiv:1711.08813.

[147] B. Hooberman, et al., Calorimetry with deep learning: Particle classification, energy regression, and simulation for high-energy physics, in: Proceedings of the Deep Learning for Physical Sciences Workshop at NIPS (2017), 2017.

[148] S. Diefenbacher, et al., DCTRGAN: Improving the precision of generative models with reweighting, J. Instrum. 15 (2020) P11004, http://dx.doi.org/10.1088/1748-0221/15/11/p11004, arXiv:2009.03796.

[149] A. Maevskiy, F. Ratnikov, A. Zinchenko, V. Riabov, Simulating the time projection chamber responses at the MPD detector using generative adversarial networks, Eur. Phys. J. C 81 (7) (2021) 599, http://dx.doi.org/10.1140/epjc/s10052-021-09366-4, arXiv:2012.04595.

[150] F. Rehm, S. Vallecorsa, K. Borras, D. Krücker, Validation of deep convolutional generative adversarial networks for high energy physics calorimeter simulations, in: AAAI-MLPS 2021 Spring Symposium at Stanford University, 2021, arXiv:2103.13698.

[151] F. Rehm, S. Vallecorsa, K. Borras, D. Krücker, Physics validation of novel convolutional 2D architectures for speeding up high energy physics simulations, EPJ Web Conf. 251 (2021) 03042, http://dx.doi.org/10.1051/epjconf/202125103042, arXiv:2105.08960.

[152] R. Kansal, et al., Particle cloud generation with message passing generative adversarial networks, in: Thirty-Fifth Conference on Neural Information Processing Systems, 2021, arXiv:2106.11535.

[153] G. Khattak, S. Vallecorsa, F. Carminati, G. Khan, Fast simulation of a high granularity calorimeter by generative adversarial networks, 2021, arXiv:2109.07388.

[154] L. Anderlini, Machine learning for the LHCb simulation, in: Proceedings of the 2021 Artificial Intelligence for the Electron Ion Collider (Experimental Applications) Workshop, 2021, arXiv:2110.07925.

[155] C. Durkan, A. Bekasov, I. Murray, G. Papamakarios, Neural spline flows, 2019, arXiv e-prints arXiv:1906.04032.

[156] L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using real NVP, 2016, arXiv e-prints arXiv:1605.08803.

[157] G. Papamakarios, T. Pavlakou, I. Murray, Masked autoregressive flow for density estimation, 2017, arXiv e-prints arXiv:1705.07057.

[158] D. Kingma, et al., Improving variational inference with inverse autoregressive flow, 2016, arXiv e-prints arXiv:1606.04934.

[159] C. Krause, D. Shih, CaloFlow: Fast and accurate generation of calorimeter showers with normalizing flows, 2021, arXiv:2106.05285.

[160] C. Krause, D. Shih, CaloFlow II: Even faster and still accurate generation of calorimeter showers with normalizing flows, 2021, arXiv:2110.11377.

[161] M. Faucci Gianelli, et al., Fast calorimeter simulation challenge 2022, 2022, URL https://calochallenge.github.io/homepage/.

[162] M. Ruan, H. Videau, Arbor, a new approach of the Particle Flow Algorithm, in: Proceedings, International Conference on Calorimetry for the High Energy Frontier (CHEF 2013): Paris, France, April 22-25, 2013, 2013, pp. 316–324, arXiv:1403.4784.

[163] M. Thomson, Particle flow calorimetry and the pandorapfa algorithm, Nucl. Instrum. Methods A 611 (1) (2009) 25–40, http://dx.doi.org/10.1016/j.nima.2009.09.009.

[164] J. Marshall, A. Münnich, M. Thomson, Performance of particle flow calorimetry at clic, Nucl. Instrum. Methods A 700 (2013) 153–162, http://dx.doi.org/10.1016/j.nima.2012.10.038.

[165] J.S. Marshall, M.A. Thomson, Pandora particle flow algorithm, in: Proceedings, International Conference on Calorimetry for the High Energy Frontier (CHEF 2013): Paris, France, April 22-25, 2013, 2013, pp. 305–315, arXiv:1308.4537.

[166] J.S. Marshall, M.A. Thomson, The pandora software development kit for pattern recognition, Eur. Phys. J. C 75 (9) (2015) http://dx.doi.org/10.1140/epjc/s10052-015-3659-3.

[167] F. Sefkow, A. White, K. Kawagoe, R. Pöschl, J. Repond, Experimental tests of particle flow calorimetry, Rev. Modern Phys. 88 (1) (2016) http://dx.doi.org/10.1103/revmodphys.88.015003.

[168] H. Tran, et al., Software compensation in particle flow reconstruction, Eur. Phys. J. C 77 (10) (2017) http://dx.doi.org/10.1140/epjc/s10052-017-5298-3.

[169] CMS Collaboration, Particle-flow reconstruction and global event description with the cms detector, J. Instrum. 12 (10) (2017) P10003, http://dx.doi.org/10.1088/1748-0221/12/10/p10003.

[170] ATLAS Collaboration, Jet reconstruction and performance using particle flow with the ATLAS Detector, Eur. Phys. J. C 77 (7) (2017) http://dx.doi.org/10.1140/epjc/s10052-017-5031-2, arXiv:1703.10485.

[171] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (1998) 2278, http://dx.doi.org/10.1109/5.726791.

[172] E. Bols, J. Kieseler, M. Verzetti, M. Stoye, A. Stakia, Jet flavour classification using DeepJet, JINST 15 (12) (2020) P12012, http://dx.doi.org/10.1088/1748-0221/15/12/P12012, arXiv:2008.10519.

[173] A. Butter, K. Cranmer, D. Debnath, B. Dillon, et al., The machine learning landscape of top taggers, SciPost Phys. 7 (2019) 014, http://dx.doi.org/10.21468/SciPostPhys.7.1.014, arXiv:1902.09914.

[174] A. Abada, et al., FCC-hh: The hadron collider, Eur. Phys. J. Spec. Top. 228 (4) (2019) 755–1107, http://dx.doi.org/10.1140/epjst/e2019-900087-0.

[175] M. Aleksa, et al., Calorimeters for the FCC-Hh, cERN-FCC-PHYS-2019-0003, 2019, arXiv:1912.09962.

[176] F. Scarselli, et al., The graph neural network model, IEEE Trans. Neural Netw. 20 (1) (2009).

[177] S. Qasim, J. Kieseler, Y. Iiyama, M. Pierini, Learning representations of irregular particle-detector geometry with distance-weighted graph networks, Eur. Phys. J. C 79 (7) (2019) 608, http://dx.doi.org/10.1140/epjc/s10052-019-7113-9, arXiv:1902.07987.

[178] E. Moreno, et al., JEDI-net: a jet identification algorithm based on interaction networks, Eur. Phys. J. C 80 (1) (2020) 58, http://dx.doi.org/10.1140/epjc/s10052-020-7608-4, arXiv:1908.05318.

[179] H. Qu, L. Gouskos, ParticleNet: Jet tagging via particle clouds, Phys. Rev. D 101 (5) (2020) 056019, http://dx.doi.org/10.1103/PhysRevD.101.056019, arXiv:1902.08570.

[180] S. Farrel, et al., The hep.trkx project: deep neural networks for hl-lhc online and offline tracking, EPJ Web Conf. 150 (2017) 00003, http://dx.doi.org/10.1051/epjconf/201715000003.

[181] S. Qasim, K. Long, J. Kieseler, M. Pierini, R. Nawaz, Multi-particle reconstruction in the High Granularity Calorimeter using object condensation and graph neural networks, EPJ Web Conf. 251 (2021) 03072, http://dx.doi.org/10.1051/epjconf/202125103072, arXiv:2106.01832.

[182] J. Shlomi, P. Battaglia, J. Vlimant, Graph neural networks in particle physics, Mach. Learn.: Sci. Technol. 2 (2) (2020) http://dx.doi.org/10.1088/2632-2153/abbf9a, arXiv:2007.13681.

[183] S. Bhattacharya, N. Chernyavskaya, S. Ghosh, L. Gray, J. Kieseler, T. Klijnsma, K. Long, R. Nawaz, K. Pedro, M. Pierini, G. Pradhan, S.R. Qasim, O. Viazlo, P. Zehetner, GNN-based end-to-end reconstruction in the CMS phase 2 high-granularity calorimeter, J. Phys. Conf. Ser. 2438 (1) (2023) 012090, http://dx.doi.org/10.1088/1742-6596/2438/1/012090.

[184] S.R. Qasim, N. Chernyavskaya, J. Kieseler, K. Long, O. Viazlo, M. Pierini, R. Nawaz, End-to-end multi-particle reconstruction in high occupancy imaging calorimeters with graph neural networks, Eur. Phys. J. C 82 (8) (2022) http://dx.doi.org/10.1140/epjc/s10052-022-10665-7.

[185] CMS Collaboration, The Phase-2 Upgrade of the CMS Endcap Calorimeter, Tech. Rep. CERN-LHCC-2017-023. CMS-TDR-019, CERN, 2017, URL https://cds.cern.ch/record/2293646.

[186] J. Pata, J. Duarte, J. Vlimant, M. Pierini, M. Spiropulu, MLPF: Efficient machine-learned particle-flow reconstruction using graph neural networks, Eur. Phys. J. C 81 (5) (2021) 381, http://dx.doi.org/10.1140/epjc/s10052-021-09158-w, arXiv:2101.08578.

[187] P. de Castro, T. Dorigo, INFERNO: Inference-aware neural optimisation, Comput. Phys. Comm. 244 (2019) 170–179, http://dx.doi.org/10.1016/j.cpc.2019.06.007, arXiv:1806.04743.

[188] G. Strong, pytorch_inferno, 2021, http://dx.doi.org/10.5281/zenodo.4597140, Please check https://github.com/GilesStrong/pytorch_inferno/graphs/contributorsforthefulllistofcontributors.

[189] L. Layer, Inference-Aware Neural Optimization for Top Pair Cross-Section Measurements with CMS Open Data (Ph.D. thesis), University of Naples Federico II, 2022.

[190] N. Simpson, L. Heinrich, neos: End-to-end-optimised summary statistics for high energy physics, 2022, http://dx.doi.org/10.48550/arXiv.2203.05570, arXiv:arXiv:2203.05570.

[191] N. Simpson, L. Heinrich, Neos: version 0.2.0, 2021, http://dx.doi.org/10.5281/zenodo.6351423, URL https://github.com/gradhep/neos.

[192] G. Cowan, K. Cranmer, E. Gross, O. Vitells, Asymptotic formulae for likelihood-based tests of new physics, Eur. Phys. J. C 71 (2) (2011) http://dx.doi.org/10.1140/epjc/s10052-011-1554-0.

[193] L. Heinrich, M. Feickert, G. Stark, pyhf, https://github.com/scikit-hep/pyhf/releases/tag/v0.6.3, version 0.6.3. http://dx.doi.org/10.5281/zenodo.1169739.

[194] L. Heinrich, M. Feickert, G. Stark, K. Cranmer, Pyhf: pure-python implementation of histfactory statistical models, J. Open Source Softw. 6 (58) (2021) 2823, http://dx.doi.org/10.21105/joss.02823.

[195] N. Simpson, Relaxed: version 0.1.3, 2022, http://dx.doi.org/10.5281/zenodo.6330891, URL https://github.com/gradhep/relaxed.

[196] M. Thomson, Modern Particle Physics, Cambridge University Press, 2013, http://dx.doi.org/10.1017/CBO9781139525367.

[197] P. Zyla, et al., Review of particle physics, PTEP 2020 (8) (2020) 083C01, http://dx.doi.org/10.1093/ptep/ptaa104.

[198] D. Sagan, Bmad: A relativistic charged particle simulation library, Nucl. Instrum. Methods A 558 (1) (2006) 356–359, http://dx.doi.org/10.1016/j.nima.2005.11.001.

[199] A. Marinelli, et al., High-intensity double-pulse x-ray free-electron laser, Nature Commun. 6 (2015).

[200] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.

[201] J. Qiang, Start to end beam dynamics optimization of x-ray FEL light source accelerators, in: NAPAC16 WEA3IO, 2016.

[202] S. Biedron, et al., Snowmass21 accelerator modeling community white paper, in: 2022 Snowmass Summer Study, 2022, arXiv:2203.08335.

[203] R. Roussel, A. Hanuka, A. Edelen, Multiobjective bayesian optimization for online accelerator tuning, Phys. Rev. Accel. Beams 24 (2021) 062801, http://dx.doi.org/10.1103/PhysRevAccelBeams.24.062801.

[204] R. Roussel, et al., Turn-key constrained parameter space exploration for particle accelerators using bayesian active learning, Nature Commun. 12 (2021) http://dx.doi.org/10.1038/s41467-021-25757-3.

[205] J. Kirschner, M. Mutný, N. Hiller, R. Ischebeck, A. Krause, Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces, 2019, CoRR abs/1902.03229. arXiv:1902.03229.

[206] J. Duris, et al., Bayesian optimization of a free-electron laser, Phys. Rev. Lett. 124 (12) (2020) 124801.

[207] A. Hanuka, et al., Physics model-informed gaussian process for online optimization of particle accelerators, Phys. Rev. Accel. Beams 24 (7) (2021) 072802.

[208] A. Edelen, et al., Neural networks for modeling and control of particle accelerators, IEEE Trans. Nucl. Sci. 63 (2) (2016) 878–897, http://dx.doi.org/10.1109/TNS.2016.2543203.

[209] A. Scheinker, A. Edelen, D. Bohler, C. Emma, A. Lutman, Demonstration of model-independent control of the longitudinal phase space of electron beams in the linac-coherent light source with femtosecond resolution, Phys. Rev. Lett. 121 (4) (2018) 044801, http://dx.doi.org/10.1103/PhysRevLett.121.044801.

[210] A. Edelen, S. Biedron, S. Milton, J. Edelen, First steps toward incorporating image based diagnostics into particle accelerator control systems using convolutional neural networks, 2016, arXiv preprint arXiv:1612.05662.

[211] J. Ögren, C. Gohil, D. Schulte, Surrogate modeling of the CLIC final-focus system using artificial neural networks, J. Instrum. 16 (05) (2021) P05012, http://dx.doi.org/10.1088/1748-0221/16/05/p05012.

[212] C. Emma, et al., Machine learning-based longitudinal phase space prediction of particle accelerators, Phys. Rev. Accel. Beams 21 (2018) 112802, http://dx.doi.org/10.1103/PhysRevAccelBeams.21.112802.

[213] A. Edelen, Neural Networks for Modeling and Control of Particle Accelerators (dissertation), Colorado State University, Available at: https://www.leelinska.com/wp-content/uploads/2021/06/Auralee_Edelen_Dissertation.pdf.

[214] L. Gupta, et al., Improving surrogate model accuracy for the LCLS-II injector frontend using convolutional neural networks and transfer learning, Mach. Learn.: Sci. Technol. 2 (4) (2021) 045025, http://dx.doi.org/10.1088/2632-2153/ac27ff.

[215] A. Scheinker, Adaptive machine learning for time-varying systems: low dimensional latent space tuning, J. Instrum. 16 (10) (2021) P10008, http://dx.doi.org/10.1088/1748-0221/16/10/P10008.

[216] A. Mishra, A. Edelen, A. Hanuka, C. Mayes, Uncertainty quantification for deep learning in particle accelerator applications, Phys. Rev. Accel. Beams 24 (11) (2021) 114601.

[217] O. Convery, L. Smith, Y. Gal, A. Hanuka, Uncertainty quantification for virtual diagnostic of particle accelerators, Phys. Rev. Accel. Beams 24 (2021) 074602, http://dx.doi.org/10.1103/PhysRevAccelBeams.24.074602.

[218] A. Edelen, C. Mayes, Neural network solver for coherent synchrotron radiation wakefield calculations in accelerator-based charged particle beams, 2022, arXiv:2203.07542.

[219] M. Berz, Differential algebraic description of beam dynamics to very high orders, Part. Accel. 24 (1989) 109–124.

[220] A. Scheinker, F. Cropp, S. Paiagua, D. Filippetto, An adaptive approach to machine learning for compact particle accelerators, Sci. Rep. 11 (1) (2021) 1–11.

[221] A. Edelen, J. Edelen, S. Milton, S. Biedron, P. Van der Slot, Using neural network control policies for rapid switching between beam parameters in a free electron laser, in: NeurIPS 2017, Long Beach, CA, 2017.

[222] R. Roussel, et al., Differentiable preisach modeling for characterization and optimization of accelerator systems with hysteresis, 2022, arXiv preprint arXiv:2202.07747.

[223] A. Ivanov, I. Agapov, Physics-based deep neural networks for beam dynamics in charged particle accelerators, Phys. Rev. Accel. Beams 23 (7) (2020) 074601, http://dx.doi.org/10.1103/PhysRevAccelBeams.23.074601.

[224] C. Mayes, et al., Lightsource unified modeling environment (lume), a start-to-end simulation ecosystem, in: Proc. of IPAC, 2021, p. THPAB217.

[225] C. Mayes, R. Roussel, H. Slepicka, Xopt v0.5.0, 2021, http://dx.doi.org/10.5281/zenodo.5559141.

[226] A. Huebl, F. Poeschel, F. Koller, J. Gu, openPMD-api: C++ & Python API for Scientific I/O with openPMD, 2018, http://dx.doi.org/10.14278/rodare.27, URL https://github.com/openPMD/openPMD-api.
[227] Curated catalogue of projects supporting openPMD. URL https://github.com/openPMD/openPMD-projects.
[228] A. Huebl, et al., openpmd: A meta data standard for particle and mesh based data, 2015, http://dx.doi.org/10.5281/zenodo.591699.
[229] R. Aaij, et al., Physics Case for an LHCb Upgrade II - Opportunities in Flavour Physics, and beyond, in the HL-LHC Era, CERN LHCC, 2018, arXiv:1808.08865.
[230] LHCb Collaboration, LHCb Calorimeters: Technical Design Report, 2000.
[231] A. Alves, et al., The LHCb detector at the LHC, JINST 3 (2008) 08005, http://dx.doi.org/10.1088/1748-0221/3/08/S08005.
[232] G. Apollinari, et al., High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report V. 0.1, in: CERN Yellow Reports: Monographs, CERN, Geneva, 2017, http://dx.doi.org/10.23731/CYRM-2017-004, URL https://cds.cern.ch/record/2284929.
[233] S. Barsuk, et al., Design and Construction of Electromagnetic Calorimeter for LHCb Experiment, Tech. rep., CERN, Geneva, 2000, URL https://cds.cern.ch/record/691508.
[234] P. Jenni, P. Sonderegger, H.P. Paar, R. Wigmans, The High Resolution Spaghetti Hadron Calorimeter: Proposal, Tech. rep., NIKHEF, 1987.
[235] M. Lucchini, et al., Test beam results with LuAG fibers for next-generation calorimeters, JINST 8 (2013) P10017, http://dx.doi.org/10.1088/1748-0221/8/10/P10017.
[236] C. Fabjan, F. Gianotti, Calorimetry for particle physics, Rev. Modern Phys. 75 (4) (2003) 1243.
[237] F. Ratnikov, Using machine learning to speed up and improve calorimeter r & d, J. Instrum. 15 (05) (2020) C05032, http://dx.doi.org/10.1088/1748-0221/15/05/c05032.
[238] A. Boldyrev, D. Derkach, F. Ratnikov, A. Shevelev, ML-assisted versatile approach to Calorimeter R & D, JINST 15 (09) (2020) C09030, http://dx.doi.org/10.1088/1748-0221/15/09/C09030, arXiv:2005.07700.
[239] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, ACM, New York, NY, USA, 2016, pp. 785–794, http://dx.doi.org/10.1145/2939672.2939785.
[240] J. Kieseler, G.C. Strong, F. Chiandotto, T. Dorigo, L. Layer, Calorimetric measurement of multi-TeV muons via deep regression, Eur. Phys. J. C 82 (1) (2022) 79, http://dx.doi.org/10.1140/epjc/s10052-022-09993-5, arXiv:2107.02119. URL https://cds.cern.ch/record/2776531.
[241] D. Lopez-Paz, M. Oquab, Revisiting classifier two-sample tests, 2016, arXiv e-prints arXiv:1610.06545.
[242] A. van den Oord, et al., Parallel WaveNet: Fast high-fidelity speech synthesis, 2017, arXiv e-prints arXiv:1711.10433.
[243] K. Long, et al., Muon colliders to expand frontiers of particle physics, Nat. Phys. 17 (3) (2021) 289–292, http://dx.doi.org/10.1038/s41567-020-01130-x, arXiv:2007.15684.
[244] N. Bartosik, et al., Detector and physics performance at a muon collider, J. Instrum. 15 (05) (2020) P05001, http://dx.doi.org/10.1088/1748-0221/15/05/p05001.
[245] A. Cemmi, et al., Radiation study of lead fluoride crystals, 2021, arXiv:2107.12307.
[246] N. Mokhov, C. James, The Mars Code System User's Guide Version 15(2016), FERMILAB-FN-1058-APC, 2017, http://dx.doi.org/10.2172/1462233.
[247] G. Abbiendi, et al., Letter of Intent: The MUonE Project, Tech. Rep. CERN-SPSC-2019-026, SPSC-I-252, CERN, Geneva, 2019.
[248] A. Prinz, et al., Search for millicharged particles at SLAC, Phys. Rev. Lett. 81 (6) (1998) 1175–1178, http://dx.doi.org/10.1103/physrevlett.81.1175.
[249] R. Acciarri, et al., Improved limits on millicharged particles using the ArgoNeuT experiment at Fermilab, Phys. Rev. Lett. 124 (13) (2020) 131801, http://dx.doi.org/10.1103/PhysRevLett.124.131801, arXiv:1911.07996.
[250] C. Argüelles, K. Kelly, V. Muñoz, Millicharged particles from the heavens: single- and multiple-scattering signatures, J. High Energy Phys. 2021 (11) (2021) http://dx.doi.org/10.1007/jhep11(2021)099.
[251] A. Ball, et al., Search for millicharged particles in proton–proton collisions at s=13 tev, Phys. Rev. D 102 (3) (2020) http://dx.doi.org/10.1103/physrevd.102.032002.
[252] A. Ball, et al., Sensitivity to millicharged particles in future proton–proton collisions at the lhc with the milliqan detector, Phys. Rev. D 104 (3) (2021) http://dx.doi.org/10.1103/physrevd.104.032002.
[253] Y. Aoki, et al., FLAG Review 2021, cERN-TH-2021-191, JLAB-THY-21-3528, FERMILAB-PUB-21-620-SCD-T, 2021, arXiv:2111.09849.
[254] A. Ramos, Automatic differentiation for error analysis of Monte Carlo data, Comput. Phys. Comm. 238 (2019) 19–35, http://dx.doi.org/10.1016/j.cpc.2018.12.020, arXiv:1809.01289.
[255] N. Madras, A. Sokal, The Pivot algorithm: a highly efficient Monte Carlo method for selfavoiding walk, J. Stat. Phys. 50 (1988) 109–186, http://dx.doi.org/10.1007/BF01022990.
[256] U. Wolff, Monte Carlo errors with less errors, Comput. Phys. Comm. 156 (2004) 143–153, http://dx.doi.org/10.1016/S0010-4655(03)00467-3, arXiv:hep-lat/0306017. Erratum; Comput. Phys. Commun. 176 (2007) 383.
[257] B. Abbott, et al., Multi-messenger observations of a binary neutron star merger, Astrophys. J. 848 (2) (2017) L12, http://dx.doi.org/10.3847/2041-8213/aa91c9.
[258] A. Albert, et al., Search for multimessenger sources of gravitational waves and high-energy neutrinos with advanced ligo during its first observing run, antares, and icecube, Astrophys. J. 870 (2) (2019) 134.
[259] S. Barthelmy, Gcn and voevent: A status report, Astron. Nachr.: Astron. Notes 329 (3) (2008) 340–342.
[260] Cherenkov Telescope Array Consortium, B. Acharya, et al., Science with the Cherenkov Telescope Array, World Scientific, 2019, http://dx.doi.org/10.1142/10986.
[261] C. Skole, Search for Extremely Short Transient Gamma-Ray Sources with the VERITAS Observatory, Humboldt University of Berlin, Germany, 2016.
[262] O. Gueta, The Cherenkov Telescope Array: layout, design and performance, 2021, arXiv e-prints arXiv:2108.04512.
[263] B. Willke, et al., The geo 600 gravitational wave detector, Classical Quantum Gravity 19 (7) (2002) 1377–1387, http://dx.doi.org/10.1088/0264-9381/19/7/321.
[264] F. Acernese, et al., Advanced Virgo: A second-generation interferometric gravitational wave detector, Classical Quantum Gravity 32 (2) (2015) http://dx.doi.org/10.1088/0264-9381/32/2/024001.
[265] B. Abbott, et al., Observation of gravitational waves from a binary black hole merger, Phys. Rev. Lett. 116 (2016) 061102, http://dx.doi.org/10.1103/PhysRevLett.116.061102.
[266] P. Amaro-Seoane, et al., Laser interferometer space antenna, 2017, arXiv preprint arXiv:1702.00786.
[267] M. Maggiore, et al., Science case for the einstein telescope, J. Cosmol. Astropart. Phys. 2020 (03) (2020) 050.
[268] M. Armano, et al., Beyond the required LISA free-fall performance: New LISA pathfinder results down to 20 μHz, Phys. Rev. Lett. 120 (2018) 061101, http://dx.doi.org/10.1103/PhysRevLett.120.061101.
[269] R. Abbott, et al., Gwtc-3: Compact binary coalescences observed by ligo and virgo during the second part of the third observing run, 2021, arXiv preprint arXiv:2111.03606.
[270] L.S. Collaboration, V. Collaboration, K.S. Collaboration, et al., The population of merging compact binaries inferred using gravitational waves through gwtc-3, 2021, arXiv preprint arXiv:2111.03634.
[271] B. Abbott, et al., Prospects for observing and localizing gravitational-wave transients with advanced ligo, advanced virgo and kagra, Living Rev. Relativ. 23 (1) (2020) 1–69.

[272] B. Abbott, et al., First low-frequency einstein@ home all-sky search for continuous gravitational waves in advanced ligo data, Phys. Rev. D 96 (12) (2017) 122004.
[273] B. Abbott, et al., Search for subsolar-mass ultracompact binaries in advanced ligo's first observing run, Phys. Rev. Lett. 121 (23) (2018) 231103.
[274] F. Acernese, et al., Status of the advanced virgo gravitational wave detector, Internat. J. Modern Phys. A 32 (28n29) (2017) 1744003.
[275] F. Acernese, et al., Status of advanced virgo, in: EPJ Web of Conferences, Vol. 182, EDP Sciences, 2018, p. 02003.
[276] D. Bersanetti, et al., Advanced virgo: Status of the detector, latest results and future prospects, Universe 7 (9) (2021) 322.
[277] P. Amico, et al., Monolithic fused silica suspension for the Virgo gravitational waves detector, Rev. Sci. Instrum. 73 (9) (2002) 3318, http://dx.doi.org/10.1063/1.1499540.
[278] S. Braccini, et al., Seismic vibrations mechanical filters for the gravitational waves detector virgo, Rev. Sci. Instrum. 67 (8) (1996) 2899–2902.
[279] R. Bonnand, A. Masserot, B. Mours, L. Rolland, E. Pacaud, M. Was, D. Passuello, The Algorithms for Control and Locking (Acl)Server Documentation, Technical Report VIR-00XX-16, 2019.
[280] J. Casanueva, I. team, ISC Tools: VPM, Acl and Data Display, Technical Report VIR-0129A-18, 2018.
[281] LIGO Scientific Collaboration and Virgo Collaboration, Gwinc, version 0.4.1, 2022, URL https://git.ligo.org/gwinc/pygwinc.
[282] J. Degallaix, OSCAR: A MATLAB based package to simulate realistic optical cavities, SoftwareX 12 (2020) 100587.
[283] A. Freise, D. Brown, C. Bond, Finesse, frequency domain INterferomEter simulation softwarE, 2013, arXiv preprint arXiv:1306.2973.
[284] D. Brown, et al., Pykat: Python package for modelling precision optical interferometers, SoftwareX 12 (2020) 100613.
[285] C. Nguyen, et al., Automated source of squeezed vacuum states driven by finite state machine based software, Rev. Sci. Instrum. 92 (5) (2021) 054504.
[286] L. Giacoppo, et al., Towards ponderomotive squeezing with sips experiment, Phys. Scr. 96 (11) (2021) 114007.
[287] D. Aisa, et al., The Advanced Virgo monolithic fused silica suspension, Nucl. Instrum. Methods A 824 (2016) 644–645, http://dx.doi.org/10.1016/j.nima.2015.09.037.
[288] G.A. Askar'yan, Excess negative charge of an electron-photon shower and its coherent radio emission, Zh. Eksp. Teor. Fiz. 41 (1961) 616–618.
[289] A. Anker, et al., Targeting ultra-high energy neutrinos with the ARIANNA experiment, Adv. Space Res. (2019) http://dx.doi.org/10.1016/j.asr.2019.06.016, (in press). arXiv:1903.01609.
[290] P. Allison, et al., Constraints on the diffuse flux of ultrahigh energy neutrinos from four years of Askaryan Radio Array data in two stations, Phys. Rev. D 102 (4) (2020) 043021, http://dx.doi.org/10.1103/PhysRevD.102.043021, arXiv:1912.00987.
[291] J.A. Aguilar, et al., Design and sensitivity of the Radio Neutrino Observatory in Greenland (RNO-G), JINST 16 (03) (2021) P03025, http://dx.doi.org/10.1088/1748-0221/16/03/P03025, arXiv:2010.12279.
[292] The IceCube-Gen2 Collaboration et al., IceCube-Gen2: The window to the extreme universe, J. Phys. G: Nucl. Part. Phys. 48 (6) (2021) 060501, http://dx.doi.org/10.1088/1361-6471/abd48, arXiv:2008.04323v1.
[293] S. Hallmann, B. Clark, C. Glaser, D. Smith, for the IceCube-Gen2 collaboration, Sensitivity studies for the icecube-gen2 radio array, in: PoS(ICRC2021)1183, 2021, http://dx.doi.org/10.22323/1.395.1183.
[294] C. Glaser, et al., Nuradioreco: A reconstruction framework for radio neutrino detectors, Eur. Phys. J. C 79 (6) (2019) http://dx.doi.org/10.1140/epjc/s10052-019-6971-5, arXiv:1903.07023.
[295] C. Glaser, et al., NuRadioMC: simulating the radio emission of neutrinos from interaction to detector, Eur. Phys. J. C 80 (77) (2020) http://dx.doi.org/10.1140/epjc/s10052-020-7612-8, arXiv:1906.01670.
[296] C. Glaser, S. McAleer, P. Baldi, S. Barwick, Deep learning reconstruction of the neutrino energy with a shallow Askaryan detector, in: PoS(ICRC2021)1051, 2021, http://dx.doi.org/10.22323/1.395.1051.
[297] S. Stjärnholm, O. Ericsson, C. Glaser, Neutrino direction and flavor reconstruction from radio detector data using deep convolutional neural networks, in: PoS(ICRC2021)1055, 2021, http://dx.doi.org/10.22323/1.395.1055.
[298] L. Bonechi, R. D'Alessandro, A. Giammanco, Atmospheric muons as an imaging tool, Rev. Phys. 5 (2020) 100038, http://dx.doi.org/10.1016/j.revip.2020.100038, arXiv:1906.03934.
[299] K. Morishima, et al., Discovery of a big void in Khufu's Pyramid by observation of cosmic-ray muons, Nature 552 (7685) (2017) 386, http://dx.doi.org/10.1038/nature24647, arXiv:1711.01576.
[300] G. Saracino, et al., Imaging of underground cavities with cosmic-ray muons from observations at Mt. Echia (Naples), Sci. Rep. 7 (1) (2017) 1181, http://dx.doi.org/10.1038/s41598-017-01277-3.
[301] R. Nishiyama, et al., First measurement of ice-bedrock interface of alpine glaciers by cosmic muon radiography, Geophys. Res. Lett. 44 (12) (2017) 6244, http://dx.doi.org/10.1002/2017GL073599.
[302] D. Mahon, et al., First-of-a-kind muography for nuclear waste characterization, Phil. Trans. R. Soc. A 377 (2018) 0048, http://dx.doi.org/10.1098/rsta.2018.0048.
[303] F. Riggi, et al., The Muon Portal Project: Commissioning of the full detector and first results, in: Proceedings, 8th International Conference on New Developments in Photodetection (NDIP17), Tours, France, July 3-7, 2017, Vol. 912, 2018, p. 16, http://dx.doi.org/10.1016/j.nima.2017.10.006.
[304] M. D'Errico, et al., Muon radiography applied to volcanoes imaging: the MURAVES experiment at Mt. Vesuvius, JINST 15 (03) (2020) C03014, http://dx.doi.org/10.1088/1748-0221/15/03/c03014.
[305] L. Oláh, et al., High-definition and low-noise muography of the Sakurajima volcano with gaseous tracking detectors, Sci. Rep. 8 (1) (2018) 3207, http://dx.doi.org/10.1038/s41598-018-21423-9.
[306] G. Mengyun, C. Ming-Chung, C. Jun, L. Kam-Biu, Y. Changgen, A parametrization of the cosmic-ray muon flux at sea-level, 2015, arXiv:1509.06176.
[307] P. Shukla, S. Sankrith, Energy and angular distributions of atmospheric muons at the Earth, Internat. J. Modern Phys. A 33 (30) (2018) 1850175, http://dx.doi.org/10.1142/S0217751X18501750, arXiv:1606.06907.
[308] L. Schultz, et al., Image reconstruction and material z discrimination via cosmic ray muon radiography, Nucl. Instrum. Methods A 519 (3) (2004) 687–694, http://dx.doi.org/10.1016/j.nima.2003.11.035.
[309] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.
[310] F. Scarselli, M. Gori, A. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans. Neural Netw. 20 (1) (2009) 61–80, http://dx.doi.org/10.1109/TNN.2008.2005605.
[311] G. Baccani, et al., The MIMA project. Design, construction and performances of a compact hodoscope for muon radiography applications in the context of Archaeology and geophysical prospections, JINST 13 (11) (2018) P11001, http://dx.doi.org/10.1088/1748-0221/13/11/P11001, arXiv:1806.11398.
[312] K. Chaiwongkhot, et al., Development of a portable muography detector for infrastructure degradation investigation, IEEE Trans. Nucl. Sci. 65 (2018) 2316, http://dx.doi.org/10.1109/TNS.2018.2855737.
[313] S. Wuyckens, A. Giammanco, P. Demin, E. Cortina Gil, A portable muon telescope based on small and gas-tight Resistive Plate Chambers, Phil. Trans. R. Soc. A 377 (2018) 0139, http://dx.doi.org/10.1098/rsta.2018.0139, arXiv:1806.06602.
[314] S. Basnet, et al., Towards portable muography with small-area, gas-tight glass Resistive Plate Chambers, JINST 15 (10) (2020) C10032, http://dx.doi.org/10.1088/1748-0221/15/10/C10032, arXiv:2005.09589.
[315] R. Gamage, et al., A portable muon telescope for multidisciplinary applications, JINST 17 (01) (2022) C01051, http://dx.doi.org/10.1088/1748-0221/17/01/C01051, arXiv:2109.14489.

[316] M. Moussawi, et al., A portable muon telescope for exploration geophysics in confined environments, in: First International Meeting for Applied Geoscience & Energy, 26 September - 1 2021, Denver (USA), SEG Technical Program ExpandEd Abstracts, First International Meeting for Applied Geoscience & Energy Expanded Abstracts, 2021, pp. 3034–3038, http://dx.doi.org/10.1190/segam2021-3581267.1.

[317] A. Giammanco, E. Cortina Gil, S. Andringa, M. Tytgat, Resistive plate chambers in muography, in: L. Olah, H. Tanaka, D. Varga (Eds.), Muography: Exploring Earth's Subsurface with Elementary Particles, in: Geophysical Monograph Series, AGU - Wiley, 2022, http://dx.doi.org/10.1002/9781119722748.ch18, (ISSN: 0065-8448). Ch. 18.

[318] R. Wilson, Radiological use of fast protons, Radiology 47 (5) (1946) 487–491, http://dx.doi.org/10.1148/47.5.487.

[319] A. Cormack, Representation of a function by its line integrals, with some radiological applications, J. Appl. Phys. 34 (9) (1963) 2722–2727, http://dx.doi.org/10.1063/1.1729798.

[320] G. Poludniowski, N. Allinson, P. Evans, Proton radiography and tomography with application to proton therapy, Br. J. Radiol. 88 (1053) (2015) 20150134, http://dx.doi.org/10.1259/bjr.20150134.

[321] R. Johnson, Review of medical radiography and tomography with proton beams, Rep. Progr. Phys. 81 (1) (2018) 016701, http://dx.doi.org/10.1088/1361-6633/aa8b1d.

[322] R. Schulte, S. Penfold, J. Tafas, K. Schubert, A maximum likelihood proton path formalism for application in proton computed tomography: Maximum likelihood path formalism for proton CT, Med. Phys. 35 (11) (2008) 4849–4856, http://dx.doi.org/10.1118/1.2986139.

[323] N. Krah, F. Khellaf, J. Létang, S. Rit, I. Rinaldi, A comprehensive theoretical comparison of proton imaging set-ups in terms of spatial resolution, Phys. Med. Biol. 63 (13) (2018) 135013, http://dx.doi.org/10.1088/1361-6560/aaca1f.

[324] C. Collins-Fekete, L. Volz, S. Portillo, L. Beaulieu, J. Seco, A theoretical framework to predict the most likely ion path in particle imaging, Phys. Med. Biol. 62 (5) (2017) 1777–1790, http://dx.doi.org/10.1088/1361-6560/aa58ce.

[325] D. Williams, The most likely path of an energetic charged particle through a uniform medium, Phys. Med. Biol. 49 (13) (2004) 2899–2911, http://dx.doi.org/10.1088/0031-9155/49/13/010.

[326] J. Alme, et al., A high-granularity digital tracking calorimeter optimized for proton CT, Front. Phys. 8 (2020) 460, http://dx.doi.org/10.3389/fphy.2020.568243.

[327] S. Jan, et al., GATE - Geant4 Application for Tomographic Emission: a simulation toolkit for PET and SPECT, Phys. Med. Biol. 49 (19) (2004) 4543–4561.

[328] H. Pettersen, et al., Design optimization of a pixel-based range telescope for proton computed tomography, Phys. Medica 63 (2019) 87–97, http://dx.doi.org/10.1016/j.ejmp.2019.05.026.

[329] A. Biguri, M. Dosanjh, S. Hancock, M. Soleimani, TIGRE: a MATLAB-GPU toolbox for CBCT image reconstruction, Biomed. Phys. Eng. Express 2 (5) (2016) 055010, http://dx.doi.org/10.1088/2057-1976/2/5/055010.

[330] M. González-Alonso, O. Naviliat-Cuncic, N. Severijns, New physics searches in nuclear and neutron β decay, Prog. Part. Nucl. Phys. 104 (2019) 165–223, http://dx.doi.org/10.1016/j.ppnp.2018.08.002.

[331] D. Dubbers, B. Märkisch, Precise measurements of the decay of free neutrons, Annu. Rev. Nucl. Part. Sci. 71 (2021) 139–163.

[332] T. Chupp, P. Fierlinger, M. Ramsey-Musolf, J. Singh, Electric dipole moments of atoms, molecules, nuclei, and particles, Rev. Modern Phys. 91 (1) (2019) 015001.

[333] J. Jaeckel, A. Ringwald, The low-energy frontier of particle physics, Annu. Rev. Nucl. Part. Sci. 60 (2010) 405–437.

[334] M. Aker, et al., Improved upper limit on the neutrino mass from a direct kinematic method by katrin, Phys. Rev. Lett. 123 (22) (2019) 221802.

[335] H. Saul, et al., Limit on the fierz interference term b from a measurement of the beta asymmetry in neutron decay, Phys. Rev. Lett. 125 (11) (2020) 112501.

[336] X. Sun, et al., Improved limits on Fierz interference using asymmetry measurements from the ultracold neutron asymmetry (ucna) experiment, Phys. Rev. C 101 (2020) 035503, http://dx.doi.org/10.1103/PhysRevC.101.035503.

[337] X. Wang, et al., Design of the magnet system of the neutron decay facility perc, in: EPJ Web of Conferences, Vol. 219, EDP Sciences, 2019, p. 04007.

[338] F. Gonzalez, et al., Improved neutron lifetime measurement with ucn τ, Phys. Rev. Lett. 127 (16) (2021) 162501.

[339] C. Abel, et al., Measurement of the permanent electric dipole moment of the neutron, Phys. Rev. Lett. 124 (8) (2020) 081803.

[340] B. Shahriari, K. Swersky, Z. Wang, R. Adams, N. de Freitas, Taking the human out of the loop: A review of bayesian optimization, Proc. IEEE 104 (1) (2016) 148–175, http://dx.doi.org/10.1109/JPROC.2015.2494218.

[341] M. Lamparth, M. Bestehorn, B. Märkisch, Gaussian processes and bayesian optimization for high precision experiments, 2022, arXiv preprint arXiv:2205.07625.

[342] J. Revels, M. Lubin, T. Papamarkou, Forward-mode automatic differentiation in Julia, 2016, arXiv:1607.07892 [cs.MS].

[343] T. Sjöstrand, S. Mrenna, P. Skands, PYTHIA 6.4 physics and manual, J. High Energy Phys. 05 (2006) 026, http://dx.doi.org/10.1088/1126-6708/2006/05/026, arXiv:hep-ph/0603175.

[344] T. Sjöstrand, S. Mrenna, P. Skands, A brief introduction to PYTHIA 8.1, Comput. Phys. Comm. 178 (2008) 852–867, http://dx.doi.org/10.1016/j.cpc.2008.01.036, arXiv:0710.3820.

[345] T. Sjöstrand, et al., An introduction to PYTHIA 8.2, Comput. Phys. Comm. 191 (2015) 159, http://dx.doi.org/10.1016/j.cpc.2015.01.024, arXiv:1410.3012.

[346] A. Castillo, et al., Genie: an interactive real-time simulation for teaching genetic drift, Evol. Educ. Outreach 15 (2022) 3, http://dx.doi.org/10.1186/s12052-022-00161-7, arXiv:10.1101/268672v3.full.

[347] T.K. Community, Kubernetes: Production-grade container orchestration, 2014, retrieved on 19-03-2022. URL https://kubernetes.io/.