Chapter 6

Deep Learning Applications to Particle Physics: A Review

Marco Rossi
Department of Physiscs
Università degli Studi di Milano, Italy
marco.rossi5@unimi.it

DOI: 10.54103/milanoup.282.c638

6.1 Abstract

Particle physics experiments like those at the LHC produce massive, high-dimensional datasets, making machine learning essential for efficient data processing and analysis. Traditional methods, based on engineered features and machine learning techniques, often discard valuable information. The rise of deep learning and accessible high-performance hardware has enabled models to learn directly from raw detector data, improving performance across various tasks. With support from labeled Monte Carlo simulations, deep learning has become increasingly prominent in particle physics. This article reviews key applications in collider physics, jet physics, tracking, fast simulation, anomaly detection, and recent advances in neutrino physics.

6.2 Introduction and motivation

Particle physics produces huge datasets. For example, the Large Hadron Collider (LHC) collects data from protons, organized in bunches colliding at $\sim 40\,\mathrm{MHz}$ frequency, with $\mathcal{O}10^8$ sensors. Each collision generates a large number of particles, whose properties must be measured and stored. Gathering this enormous amount of data might give physicists enough statistics to study interesting rare events.

These facts highlight that not only the quantity of collected data is immense, but also its dimensionality. Therefore, machine learning is a set of techniques of paramount importance in this scenario, providing automation in data processing and dimensionality reduction of such information.

For years, physicists in the High-Energy Physics (HEP) domain investigated machine learning techniques like neural networks, support vector machines, genetic algorithms and predominantly boosted decision trees (BDTs) implemented in the TMVA framework [1]. This approach was based on the idea of engineering high-level low-dimensional quantities from raw detector data to be fed as multiple inputs to multivariate analysis (MVA) and provided important boosts in many data analysis tasks. However, it was clear that reducing the input dimensionality consisted in discarding a large part of potentially interesting information, leading to inherently limited algorithms. As a consequence, these tools often struggled to provide competitive performance in applications where the dimensionality gap between raw data and extracted features grew large.

Starting in 2012, the computer science community achieved important results in training big neural networks [2–4], converging to models able to provide outperforming solutions against traditional approaches. These publications set the stage for further investigation of deep learning techniques in many other research fields, including particle physics. Moreover, this explosion of research activity was helped by the recent technical improvements in hardware accelerators and their spread as consumer-grade products, granting high-quality computational power at affordable prices. In HEP, this wave mostly translated into the idea that engineered features, designed at cost of time and great intellectual effort, could have been replaced by high-dimensional low-level raw information if processed by deep enough models.

Besides producing large datasets, the particle physics field is especially suited for the proliferation of deep learning applications thanks to the availability of labeled datasets from Monte Carlo event generators. These programs aim to simulate the physics world employing probabilistic laws, accurately describing particle interactions hierarchically from the sub-atomic scales, all the way up to include the macroscopic long-range effects of physics theories. [5–9] represent modern examples of Monte Carlo event generators. The role of the artificial intelligence tools in this picture is often to grasp the probability laws of nature from sets of observations (like particle momenta and charge) and estimate the corresponding Monte Carlo truths (such as the type of a particle or even an interaction between particles in the event).

The article is dedicated to an overview of the main results in particle physics obtained with deep learning models. We split the plethora of models proposed in the literature by their sector of application. Among physics at colliders, we identify four main areas: jet physics, tracking, fast simulation and anomaly detection. Conversely, in dealing with non-collider physics, we restrict our attention to the advancements in deep learning tools for neutrino physics only, given its prominent role in the present thesis work.

Regarding physics at colliders, we develop a detailed discussion on jet physics and tracking, while mentioning the AI applications to fast simulation and anomaly detection only briefly in this section. We remark that the fast simulation of detector data is mainly achieved with the implementation of Generative Adversarial Networks (GAN) [10]: the model generates the specific detector response with a fast inference pass of the GAN generator, producing physical distributions from synthetic random numbers.

The anomaly detection applications, instead, are mainly devoted to searches of new physics, which is not described by the current theories, i.e. the Standard Model (SM) [11, 12]. Physicists acknowledge that SM is not the ultimate theory of nature and contains many problems and inconsistencies [13, 14]. Therefore, the search for evidence of the existence of Beyond Standard Model (BSM) theories is currently an active field of research. In this picture, machine learning algorithms aim to identify rare events or tensions between data and theoretical predictions that signal the presence of some new physics mechanism. In this area, model-dependent searches aim to identify new kinds of particles or interactions through classification, such as in [15–18], while model independent approaches [19, 20] design ad-hoc strategies to look for new physics with a model agnostic approach.

The remainder of the article is organized as follows. First, in Section 6.3, we describe the application of machine learning and deep learning to jet physics from a historical point of view. Then, Section 6.4 illustrates the methods developed within the neutrino oscillation research field. In particular, we categorize the reviewed approaches by the specific neutrino experiment which proposed each solution. Section 6.5 is devoted to the problem of tracking at colliders. Finally, we present our conclusions in Section 6.6.

6.3 Jet physics

In this section, we deal with deep learning applications to jet physics focusing on how the different ways of encoding the available data dramatically changed the physicists' approach to the problem: new kinds of data encoding opened the possibility to implement several models to try to solve the problems of jet tagging and pileup mitigation, two central issues in this research field.

Events at HEP colliders are associated with a hierarchical picture of subsequent particle splittings called parton showering. This mechanism consists in recursive branchings, where each particle in the event involved in the main interaction undergoes multiple subsequent splittings in a $1 \rightarrow 2$ fashion, resulting in a tree structure. At this stage, a large number of particles is created and eventually, their momenta directions are mostly focused in a collimated region around the particle initiating the shower, called a jet. These concepts are key ingredients for Monte Carlo event generators, which are tools that manage to link the predictions of physics theories with the outcomes of the measuring experiments at colliders.

Machine learning applications to jet physics mainly involve classification algorithms and include flavor tagging, jet substructure tagging, quark-gluon tagging and pileup removal. All the tagging tasks are related to the identification of the shower initiating particle from the knowledge of the properties of either the final particles representing the tree leaves of the jet or the whole tree nodes itself. Flavor tagging classifies the jet among heavy (c, b, t) or light (u, d, s) quarks, gluons or W/Z/H bosons. Jet substructure tagging, instead, discriminates between W/Z/H and t jets. Finally, quark-gluon tagging is the ability to distinguish between the two kinds of particles contributing to the main source of background in HEP events, sometimes named QCD background.

Pileup is a concept that roots in the design of accelerators machines: in order to increase the probability to produce interactions, at colliders, bunches of protons tightly packed together are smashed against each other, rather than individually. The luminosity $\mathscr{L}$ is a quantity measuring such compactness: the the higher the luminosity, the more many protons are squeezed together in the bunch and increase the expected number of collisions. As a consequence, it is likely that for each beam crossing, more than one couple of protons scatters, emitting low-energy radiation at wide angles named pileup (PU), as opposed to the interesting high-energy interaction often referred as the leading vertex (LV). Pileup is extensively studied at colliders and depends on the machine operative setup.

The data collected by the two major experiments at LHC, Atlas and CMS, estimated a number of pileup collisions per event $n_{\mathrm{PU}}{\sim}20$ during the LHC run II from 2015 to 2018, while for run III (2021–2023) and the High Luminosity LHC phase (HL-LHC, from 2026), this quantity is expected to increase to $n_{\mathrm{PU}}{\sim}80$ and $n_{\mathrm{PU}}{\sim}200$, respectively.

Pileup interactions modify the shape of the quantities measured in the events, affecting jet properties like its overall momentum and mass, rather than jet multiplicity in the event. Being able to design automatic tools to mitigate those effects is expected to be one of the biggest data analysis challenges during the forthcoming LHC phases. These considerations justify the importance given to pileup mitigation strategies at colliders.

The next paragraphs will review the proposed techniques in the literature concerning the described two areas of jet tagging and pileup removal. Both of them try to present the advancements in the research activity as it evolved during the last decades. from the point of view of the different data representations of jet objects used as inputs of the several neural network architectures proposed.

6.3.1 Tagging

Jet tagging mainly concerns classification algorithms. Since the early ′90s, shallow artificial neural networks have been used to detect the type of jet-initiating particles. In these initial years, the predominant approach was to feed neural networks, comprised of just a few layers, with event features tailor-made for the specific task or, sometimes, by packing jet information into small vectors of fixed size, containing the most representative characteristics of the object.

Following this idea, [21] exploited a neural network with 3 fully connected hidden layers with 6 neurons each, to process the 4−momenta of the four leading particles within a jet, to discriminate between quarks and gluons. [22], instead, implemented a neural network to distinguish between b and c jets al LEP, with the help of the Fortran77 JETNET 3.0 library [23], which represented the de facto standard for machine learning in HEP physics during those early years.

The advent of deep learning and the improvements in hardware accelerator technologies paved the way for new strategies to solve the jet tagging problem. [24] processed for the first time entire events through neural networks: their insight was to encode calorimeter information as an image. The image consisted of a two-dimensional regular grid in the pseudorapidity η and azimuthal angle ϕ coordinates, where each pixel value described the energy deposited by the particle, or, equivalently, its transverse momentum p_T.

The pixels of the image contained raw event information that could be used to compute discriminative quantities. The authors implemented a recipe to compute the Fisher linear discriminant [25] after some physics-inspired preprocessing of the images. The algorithm was tested for W boson tagging against QCD background (gluon tagged jets), providing performance improvements against the traditional discrimination method based on the N-subjettiness (τ_2/τ_1) quantity [26, 27].

Raw inputs-based neural network tools started being investigated extensively from that point onward. Examples can be found in top quark tagging tasks [28] and jet substructure classification (namely, understanding if the considered jet is due to a showering of a low-mass single particle or a massive particle decaying into multiple fast-moving lighter objects producing overlapping jets in the calorimeter, like for the $W \to qq$ process) [29]. Another application of this framework has been presented by [30], who studied the dependency of trained models on the Monte Carlo truths labels in the training datasets. The key observation pointed out that the supervised learning algorithm might bias the model predictions following the QCD approximations employed by the specific generator used to collect the dataset, rather than focusing on learning the underlying true laws of nature. The work raised the problem of the interpretability of neural networks in the jet physics research field for the first time, finding large discrepancies when testing the models on datasets produced by different generators. The authors' final assertion underlined the need to deeply understand how the input information is exploited to extract the output and what assumptions a trained architecture relies on.

The calorimeter tower representation of [24] has then proven to be a powerful representation of jets events, mainly thanks to the success of Convolutional Neural Networks (CNNs) [3]. Indeed, [31] exploited CNNs to inspect the (η, ϕ) plane deposited energy encoding of jet events. They proposed a network to identify highly boosted W bosons against the quark-gluon QCD background. The inputs were initially cast to grayscale images (one channel only), however further developments considered also multi-channel input images. In particular, [32] proposed to build a three-channel RGB image tensor stacking information from charged and neutral particles' transverse momenta, plus the number of charged particles measured within each pixel area.

The standard calorimeter tower images were not the only image-like encoding that has been studied in the literature: an alternative strategy has been given by the Lund Jet Plane [33]. It considers kinematic variables arising while rewinding backward the Cambridge Aachen clustering algorithm [34, 35], attempting to reconstruct a de-clustering history of a jet. The output of this procedure is an ordered set of variables that characterizes a jet object and can be seen as an image tensor. According to the authors, this description should provide greater output interpretability as well as discrimination power when employed in classification tasks.

Although the image based successfully tackled the jet classification problem multiple times, CNNs rely on the assumption that pixels form a perfect grid, while it is known that actual detectors' geometry is not perfectly regular. Moreover, jet images often contain sparse features which lead to inefficient processing by convolutional kernels. Hence, different data representation strategies have been investigated. A jet object is the result of a clustering algorithm[1], which generates a list of jet constituent particles. As a consequence, it can be represented as a sequence of tracks and vertices, forming an acyclic-directed graph or, equivalently, a tree. The complication arising from adopting this encoding scheme is mostly given by the variable length size of the sequences, which cannot be handled by standard Feed Forward Neural Networks.

[37] overcame this difficulty by proposing a Recursive Neural Network architecture comprised of Long Short Term Memory (LSTMs) [38] cells, able to deal with variable-size inputs. The work takes into account other solutions involving Feed Forward Neural Networks supported by input truncation and zero padding. The authors presented a comparison of the different strategies applying them to the problem of light (u, d, s, c) versus heavy quark (b) jet flavor classification, achieving similar performance for the different models. Since then, several algorithms based on RNNs have been proposed to become part of the Atlas [39, 40] and CMS [41, 42] software stack and many more have been published to exploit variable size inputs [43, 44].

The introduction of RNNs allowed for the treatment of the jet as lists and trees of particles. However, even if some natural ordering is obtained by clustering like in the k_t−algorithm, this is just an approximation. Imposing an ordering often means

[1] A modern C++ implementation of jet definitions and clustering algorithms is given by the FASTJET 3.0 [36] library.

Table 6.1 *Summary of the proposed architectures for jet classification. The table is inspired from [51].*

	quark/gluon	W/Z	H	b/c	t
Image	[32, 52, 53]	[29, 31]	[54]		[55, 56]
Sequences	[53]			[39]	
Tree	[44]	[57, 58]			[56]
Graph			[45]		
Unordered set	[47]				
Point Cloud	[48]				[48, 56]

establishing a spatio-temporal relationship between particles to be identified as a history producing a specific final state. However, quantum mechanics principles break down the causality concepts of space and time relying on probabilistic laws. Therefore, the most natural way to represent a jet object would be to decouple from this artificial ordering and process it like an unordered set of particles described by their 4−momenta and quantum numbers. Designing an architecture with the ability to deal with unordered sets of particles would then be desirable. Graph neural networks, deep sets and point clouds networks achieve this objective.

[45] implemented a RelNet [46] to accomplish W jet tagging against QCD background: particles are regarded as graph nodes and the adjacency matrix is learned to aggregate information between nodes through a message-passing operation. [47] constrained a network architecture acting on deep sets, to build infra-red and collinear (IRC) safe observables: the information in each particle observable is then aggregated with a global permutation-invariant operation. The authors implement two different networks called EnergyFlow and ParticleFlow, which consider IRC-safe and non-IRC-safe quantities, respectively. [48] proposed to to use the EdgeConv operation [49] on the k-nearest neighbors points of each particle in a point cloud. The point cloud jet representation encodes an event as a matrix where each row represents a vector of properties associated with each particle.

Figura 6.1 shows how the neural network input representations for jet physics evolved through time. Table 6.1, instead, summarizes all the relevant applications of machine learning and deep learning to jet physics.

6.3.2 Pileup mitigation

Pileup contribution from charged particles can be removed almost completely thanks to the excellent vertex resolution at the ATLAS and CMS detectors [59–61]. These particles are identified and removed from the event with the charged-hadron subtraction (CHS) procedure [62]. The challenge comes from pileup radiation due to neutral particles, which must be taken into account with specialized algorithms. The rich literature on traditional methods can be categorized on the level of

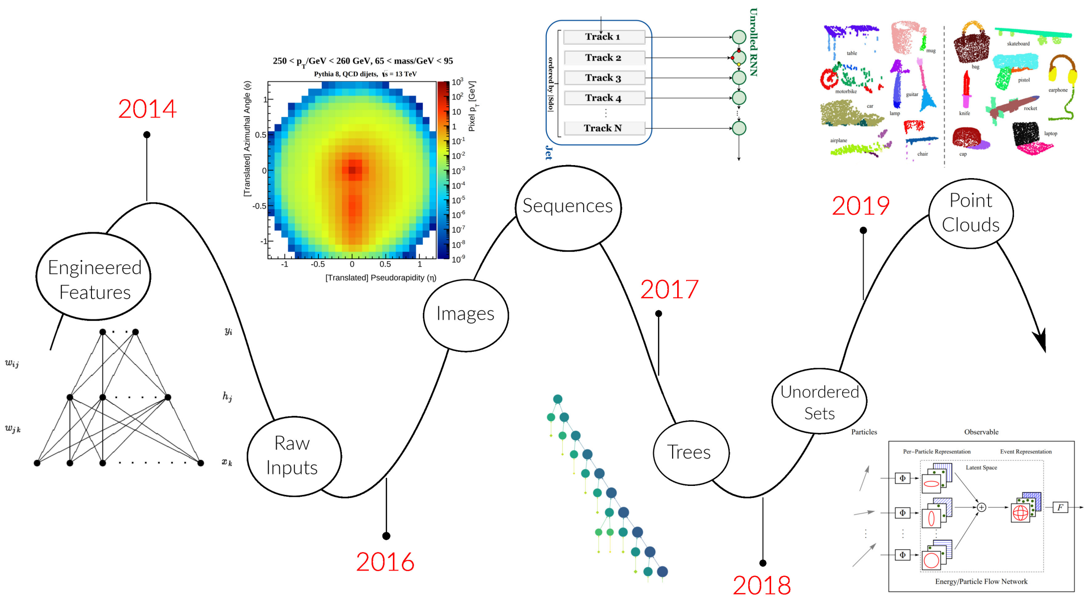


Figure 6.1 *The data structure timeline of physics jets: only engineered features were used as input to neural networks before 2014; after [24], several encoding structures have been investigated to efficiently represent jets. The descriptive pictures in the chart are taken, in order of appearance, from: [23, 31, 39, 44, 47, 50].*

detail these tools act on. A first technique, known as jet areas subtraction [63], relied on calibrating jet level information, scaling its 4−momentum by a relevant factor. However, this procedure did not manage to mitigate the pileup contribution effectively for the computation of several subjet observables.

Therefore, other algorithms have been proposed to act on the internal jet structure, namely at the subjet level. Examples of such tools are usually classified as jet constituents pre-processing, jet or event grooming, subjet corrections and constituent corrections. Grooming, in particular, progressively removes jet constituents contaminated by pileup, cutting the tree description of a jet arising from clustering algorithms through filtering [64], pruning [65,66] and trimming [67]. SoftKiller [68], instead, is a popular event-level grooming algorithm that equally divides the (η, ϕ) plane in patches of a certain area and imposes a cut-off p_T^{cut} on the transverse momentum cumulated on the patches, such that half of the patches are radiation free. This tool has been used by several works as a benchmark to test the goodness of the proposed models.

Finally, the most advanced pileup mitigating algorithms act at the deepest level, working on a particle-by-particle basis [69–71]. Among those, an excellent example is PUPPI, which evaluates a scaling factor for each particle 4−momentum in the event, by computing a local shape variable α, which collects information about each particle neighborhood. The α distribution for charged particles, for which pileup information is known thanks to the CHS method, can be exploited to extract the scaling weight for each neutral particle. The net effect is to correct jet and subjet observables of interest for physics analysis as if the pileup effects have been switched off.

Machine learning applications for pileup removal mainly act at the particle level, since, as already discussed in this chapter, they can extract useful information from the low-level description of events. The first application of this framework was PUMML [72], a Convolutional Neural Network to inspect RGB images in the (η,ϕ) plane. The three RGB channels convey information about the transverse momenta of all neutral particles, all charged pileup particles and all charged leading vertex particles, respectively. The architecture is trained in a supervised way to output the missing p_T of the neutral leading vertex particles. Performance comparisons were presented against SoftKiller and PUPPI algorithms for the reconstruction of mass and transverse momentum distributions of the LV jets.

Other models have been proposed in subsequent years, trying to take advantage of the different technologies developed in the computer vision research field: [73] introduced PUPPIML, a network working on a graph representation of the event. After subtracting the charged pileup particles, the remaining ones are arranged in a graph where all pairs of particles closer than a fixed radius R_1 in the (η,ϕ) plane (default value is $R_1 = 0.3$) are connected by an edge. This graph is processed by several Gated Recurrent Units (GRU) [74] and outputs a binary score for each particle to discriminate between leading vertex and pileup. The authors claimed performance improvements up to ~30% of PUPPIML against PUPPI on the resolution of jet-related quantities and even higher ones with respect to SoftKiller.

PUMA [75] exploits the attention mechanism [76] to tackle the pileup mitigation task in realistic detector scenarios, corresponding to extreme setups with $n_{PU}\sim 200$. The performance was tested against classical benchmarks, like CHS and PUPPI, showing large improvements in the key reconstructed jet variable distributions. The authors judged this work as an important achievement in showing the usefulness of statistically-learned algorithms during the HL-LHC phase.

Beyond the supervised algorithms presented in the paragraphs above, some alternative approaches have been proposed. [77] implemented a grooming procedure within a reinforcement learning (RL) framework: a jet is represented as a binary tree graph where each node i is described by a Lund plane derived variable $\mathcal{T}^{(i)}$, containing the state vector observed by the RL agent as well as a pointer to the the parent node and the two child ones. The algorithm concerns applying recursively a policy function π_g to all the nodes in the graph. The policy function outputs the probability to groom or not a node in the tree, which determines the action of the RL agent on the environment. The agent is trained through a smooth reward function carefully designed to optimize the resolution of kinematic variables both at the graph and node level, such as the mass of the resulting jet or the fact that a node contributes to the wide-angle soft radiation (PU) rather than to the hard-collinear emission (LV), respectively.

A semi-supervised learning approach for Graph Neural Networks, named Graph SSL, has been investigated by [78]. The main advantage introduced by this technique is the possibility to train directly on real detector data, without the need of Monte Carlo truth labels. The algorithm is based on supervised training to learn charged particles' properties, while inference is done on neutral particles, which represents the main challenge in the identification of pileup. A careful masking procedure is required to train effectively on charged particles as if they were neutral ones. This method allows for avoiding the complex issues regarding the dependence of the models on Monte Carlo datasets and the high costs in terms of simulation time to reproduce physics processes with Monte Carlo generators. The authors benchmarked Graph SSL against PUPPI and observed performance improvements both for the accuracy in the LV-PU identification at the particle level and regarding the resolution of the reconstructed jet quantities.

6.4 Non-collider physics (neutrino)

This section presents the applications of deep learning in neutrino physics. For sake of brevity, we restrict our attention to experiments focusing on neutrino oscillations only: in fact, a large number of experiments are concentrating their efforts to improve the understanding of such mechanism. In the field of neutrino physics, deep learning is mainly investigated as a tool for event classification and automated reconstruction algorithms. The former task is well established in the physics community since the end of the 20^{th} century as a robust strategy to select signal and reject background events, while the latter is still an open issue and many techniques are currently being inspected.

The first neural network application to neutrino event classification is given by [79] in the context of the SNO experiment. The network contains a modest $\mathcal{O}(700)$ number of trainable parameters employing a shallow feed-forward neural network with $\mathcal{O}(30)$ inputs engineered on detector hit patterns and count a single hidden layer with 20 neurons, to distinguish between four classes of neutrino interactions:

- Charge current (CC): $\nu_e + {}^2\mathrm{H} \to p + p + e^-$;
- Electron scattering (ES): $\nu_x + e^- \to \nu_x + e^-$;
- Chlorine neutral current (NC): $n + {}^{35}\mathrm{Cl} \to {}^{36}\mathrm{Cl} + \gamma$;
- Deuteron neutral current (ND): $n + {}^2\mathrm{H} \to {}^3\mathrm{H} + \gamma$.

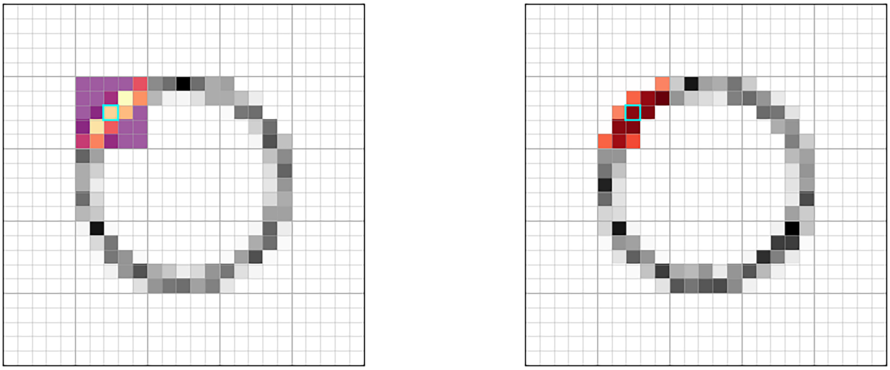
The investigation of artificial neural networks eventually spread among neutrino physicists. The main cause of this success can be found in the detector data format: most of the detectors built for detecting neutrinos produce image-like data, which can be processed with the help of modern computer vision and convolutional neural network techniques. As a consequence, two decades after the SNO paper, the NOνA collaboration [80] proposed to build a CNN to identify neutrino background interactions [81]. The network, named Convolutional Visual Network (CVN), is comprised of two GoogLeNet [82] separate branches inspecting (x, y) and (y, z) hit projections, respectively. The two resulting feature embeddings are concatenated and fed into a classifier to extract the desired multi-class score. It is interesting to notice that the two views are not concatenated along the channel axis like in RGB images: the authors recognize that each coordinate pixel in 2D projection would overlap unrelated features, as they do not refer to the same (x, y, z) spatial 3D point. The network is trained to compute the ν_e appearance and ν_μ disappearance rates. This work marked a milestone in the field since it became the first neural network-based analysis whose results were included in a physics publication [83].

The GoogLeNet architecture has also been exploited by [84] to search for neutrino-less double beta decay $0\nu\beta\beta$ [85] process at the NEXT experiment. In this application, as opposed to the NOνA one, three 2D projections of event images are concatenated like RGB images. The authors highlight an improvement against the traditional blob discrimination method described in [84].

The techniques just reviewed try to process images with CNNs, achieving better performance than baseline methods. In recent years, several other articles and many experimental collaborations showed interest in developing CNN-based classifiers, showing the performance superiority of this approach compared to the traditional methods of event classification [81, 86–90].

CNNs have also proved useful in several tasks of reconstruction. First, they have been used to tackle regression problems, namely to predict the interacting neutrino energy value [91] or its direction in the frame of reference of the detector [88, 92]. Then, they helped in identifying non-empty activity regions, drawing bounding boxes around interactions to discard uninteresting parts of the input images: this technique is called Faster-Region Convolutional Neural Network (Faster-CNN). Alternatively, [93] exploit a model inspired by the U-Net architecture [94] to precisely locate track end-points and shower vertices. Finally, [95–97] implemented CNNs aiming at segmenting the input images to assign each pixel to a type of particle drifting in the detector, detecting Region Of Interest (ROI) coordinates in raw data in 2-dimensional planes and 1-dimensional channels, respectively.

Although Convolutional Neural Network models are the de-facto standard in image processing, neutrino detectors often collect data with special features that cause these techniques to be inefficient. The majority of the events recorded by these experiments contain sparse long 1-dimensional tracks with locally dense features. The result is that large portions of such images are empty, leading to a waste of computational resources when inspected with convolutional filters: those filters, indeed, transform equally both the empty spaces and signal regions. Additionally, the huge quantity of sensors in these detectors gathers information into



(a) Normal 2-dimensional convolution: single kernel transformation.

(b) Sparse 2-dimensional convolution: single kernel transformation.

Figure 6.2 *The cyan pixel highlights the current convolution pixel. The normal convolution kernel operates on all pixels within the kernel window, while the sparse one acts on non-zero neighboring pixels only.*

high-resolution images with $\mathcal{O}(10^6)$ of pixels, that barely fit the memory constraints of modern hardware accelerating devices.

During the last few years, then, the neutrino community has dedicated a great effort to design better encodings and experiment novel techniques to analyze such data. In this picture, Sparse Convolutional Neural Networks (Sparse CNNs) [98] and Graph Neural Networks (GNNs) [99] have been investigated. Two architectures based on Sparse CNNs, acting with convolutional filters on non-zero pixels only, have been implemented by [100, 101]. The operation, depicted by Figura 6.2b, allows to store the event data in an efficient sparse format and dramatically decrease the number of operations required by each convolutional layer forward pass.

On the other hand, GNNs provided performance improvements in processing data from detectors with irregular geometry like IceCube [102] and JUNO [103]. Besides this success, even if the data graph encoding is not always a natural choice when dealing with either image or point cloud data, several works [104–106] showed good results implementing these architectures for event classification and other reconstruction tasks.

The use of GNNs is subject to two major issues. First, there is no standard choice of encoding neutrino data into a graph. Luckily, the sparsity of neutrino images allows identifying detector hits as graph nodes, resulting in graphs of manageable sizes (usually up to a few thousand nodes). Node connectivity, instead, is use-case dependent. The majority of the authors reviewed in this paragraph use similar approaches with small modifications: they rely on some distance metric, computed between each pair of nodes i and j in the graph, and a pre-defined cut-off value d_{cut} above which no edge between the corresponding nodes is drawn. Alternatively, they propose to weight each edge with a normalized version of the distance metric value itself, such that distant nodes have a suppressed information flow in the network. The second problem related to this approach is the additional overhead represented by the graph construction operation: this is often done through dedicated algorithms, like in [103, 104], and must be repeated for all events inevitably increasing the pre-processing wall-time.

In table 6.2 we collect the main deep learning applications to neutrino oscillation experiments found in the literature. The table groups the works published by several collaborations into five task categories:

- event classification, which encompasses event topology, interaction classification and background rejection;
- regression, grouping neutrino energy reconstruction and neutrino direction reconstruction;
- object detection, that collects interaction localization (vertex reconstruction, bounding box drawing around pixel activity), track end-point localization and shower starting-point localization;
- graph operations, that include background rejection (graph classification), clustering (node classification), 3D reconstruction (graph cleaning through node classification) and primary particle classification (edge classification);
- segmentation, receiving contribution from pixel-level particle identification, instance segmentation and region of interest (ROI) finding.

6.5 Tracking

Tracking is a central process of reconstruction at colliders, it consists in grouping detector hits within an event produced by a charged particle interacting in the inner detector region and moving inside a static magnetic field. The traditional approach is based on four different phases: hit clustering, track seed finding, track building and track fitting. The present discussion gives a brief overview of the traditional method employed to solve the tracking problem and it is inspired by specialized reviews on tracking strategies at LHC, [113–115].

The tracking process consists in sequentially reducing with clustering algorithms the number of data from $\mathcal{O}(10^8)$ detector readout channels, to $\mathcal{O}(10^4)$ hits containing energy depositions and finally to $\mathcal{O}(10^3)$ tracks per event. The hierarchical approach starts with hit clustering, which consists in finding the $3-$dimensional locations of hits and the corresponding deposited energies from the pixel-level raw data readouts.

After this first stage, the two most computationally expensive steps take place. First, the hits in the inner detector are processed to identify triplets, which consist of the minimum number of points to estimate two important track parameters, namely the curvature and the perigee with respect to the center of the interaction

Table 6.2 *Review of the deep learning for neutrino physics publications. The first column identifies which detector the publication focuses on. PilarNet [107] is a generalpurpose open dataset for LArTPCs data. Note: [100] was published before the PilarNet [107] dataset but deals with similar data and objectives. The citations are colorcoded based on the neural network type implemented in the relative work: FFNNs, CNNs, GNNs, Hexagonal CNNs, Sparse CNNs, Quantum CNNs.*

Detector	Event classification	Regression	Object detection	Graph	Segmentation
SNO	[79]	–	–	–	–
NEXT	[108]	–	–	–	–
Daya Bay	[86]	–	–	–	–
NOνA	[81]	[91]	–	–	–
MicroBooNE	[87, 90]	–	[87]	–	[95] [109]
KM3NeT/ORCA	[88]	[88]	–	–	–
DUNE/pDUNE	[89] [110]	[92]	–	[106]	[96]
JUNO	[111]	[103]	[103]	[103]	–
SuperFGD (T2K)	–	–	–	[105]	–
IceCube	[102]	[112]	–	[102]	–
ArgoNeuT	–	–	–	–	[97]
PilarNet	–	–	[93]	[104]	[100, 101]

region. The three hits in each triplet form a seed for the final track. Therefore, this step fixes the final track multiplicity.

Second, once the seeds are selected, the proper track construction process starts: the trajectory is sequentially extrapolated from the triplet from the inner to the outer layers of the detector. Many pattern recognition techniques have been designed to tackle this problem, ranging from global methods, such as conformal mapping and Hough transform [116], to local ones, like the track road methods. However, the most efficient algorithm in use is the Kalman filter [117–119]. A more refined version of the original algorithm, the Combinatorial Kalman Filter [120], is leveraged to build tracks from seeds, including the possibility to keep track of branching when multiple candidate points are identified within the same layer and eventually, discard the fake tracks with high efficiency.

The final stage of the tracking problem, namely track fitting, requires estimating the track parameters for each reconstructed trajectory. These include the location of the interaction vertex, the direction of the track along with its curvature and the momentum associated with the interacting particle. Moreover, tests to remove outliers that do not belong to the track are performed in this final phase to further refine the output. This technique achieves almost perfect performance, meaning that the investigation of new methods is devoted to optimizing the existing software implementation and trying to reduce CPU usage time.

However, the next generation High Luminosity LHC (HL-LHC) phase [121], starting from 2026, will see an increase in the current luminosity setup of the Large Hadron Collider by a factor of 10, putting these low-level reconstruction tools under enormous stress. It is expected, in this collider configuration, a great improvement in the hit detector occupancy and the particle tracking software should be able to manage charged particles at a rate of $\mathcal{O}(50\,\mathrm{MHz})$. The traditional approach does not scale at such regimes. A naive solution would be to limit the reconstruction to detector regions around specific calorimetry depositions compatible with rare signatures like leptons or jets with high p_T. However, this approach will completely neglect other phenomena that might hide in discarded regions, like low p_T ones. Hence, alternative methods are currently under investigation.

In this context, the HEP.TrkX project [122] aims to study deep learning solutions to the particle tracking issue. The main outcome has been a model [123] combining CNNs and Recurrent Neural Networks (RNNs), mainly employing Long Short Term Memories (LSTMs) cells, to reconstruct tracks within a simplified detector simulation. The generated data involve straight-line tracks and neglect all other kinds of physical complexities, such as track curvature, material effects and detection inefficiencies. The model is trained to solve two tasks in particular: a 2−dimensional single-track reconstruction starting from seeded hits and an end-to-end estimation of the track parameters without any seeding.

Detector data are projected onto two axes representing the detector layer and the channel within each specific layer. The tool opens for the possibility of encoding irregular layer geometries of varying size with two strategies: either zero padding the input to retrieve a regular rectangular grid, or through an autoencoder-like architecture that embeds each layer input into a fixed-size vector representation with the help of a dense network, followed at the end of the pipeline by another fully connected layer that projects back the output into the original layer dimensionality. The data encoding based on bi-dimensional images has also been exploited by [124], the key idea is again to model the recursive track-following approach of the Kalman filter through an LSTM, showing promising results in a semi-realistic detector simulation.

Other approaches based on Graph Neural Networks (GNNs) have been introduced by [125] driven by the observation that the image-like representation of the data would not be able to manage realistic use cases matching the HL-LHC conditions. Indeed, the collider and detector updates will provide high-dimensional and sparse data due to the increased number of detector layers built with irregular geometries, which would probably cause inefficiencies in the standard approaches with CNNs. The authors advocate the investigation of methods acting on the space-point representation of data, instead, involving variable amounts of hits per event and exploiting the full detector resolution.

In 2018, the TrackML competition [126] took off within the HEP community with the intent of finding the best candidate for the future particle tracking algorithm. The desired feature of such a tool would be to achieve the best performance score across several metrics reflecting the need to target high reconstruction efficiencies with the fastest algorithm in terms of inference time. Event examples from the TrackML dataset highlight a large number of tracks to be reconstructed and the complex detector design.

Following this competition, HEP.TrkX evolved into the Exa.TrkX project [127] which investigated a wide variety of models to solve the task, mainly through GNNs [128, 129]. The project finally published an article [130] summarizing the GNN pipeline on the TrackML dataset, towards a first validation on ATLAS and CMS real detector data. The potential of GNNs has also been exploited on implementations for specific hardware acceleration, mainly provided by Field Programmable Gate Arrays (FPGAs) [131].

6.6 Conclusion

In this article, we reviewed the most important contributions to the literature concerning the applications of deep learning in the high-energy and neutrino physics fields. We presented an overview of different sectors of active research, including both collider and non-collider physics. This work aims to show the huge success of deep learning models when applied to problems in the physics domain. Although the first implementations of machine learning techniques date back to some decades ago, the popularity of such methods is constantly increasing. The main challenge for the next decade is to demonstrate that such models provide a robust generalization to unseen data, so that they can become the standard approach to solve the complex problems we introduced in this work and many others. The interpretability of their predictions is going to be another crucial point in the affirmation of these solutions and will certainly be a hot topic in the research of the following years.

Bibliography

[1] H. Voss, A. Höcker, J. Stelzer, and F. Tegenfeldt, "TMVA, the Toolkit for Multivariate Data Analysis with ROOT," in *Proc.of XI Int. Workshop on Advanced Computing and Analysis Techniques in Physics Research*, vol. 050, 2009.

[2] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with

deep convolutional neural networks," in *Proc.of the 25th Int. Conf. on Neural Information Processing Systems*, 2012, pp. 1097–1105.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "deep learning," *Nature*, vol. 521, pp. 436–444, may 2015.

[4] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[5] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis *et al.*, "Geant4a simulation toolkit," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506, no. 3, pp. 250–303, 2003.

[6] T. Sjöstrand, S. Mrenna, and P. Skands, "PYTHIA 6.4 physics and manual," *Journal of High Energy Physics*, vol. 2006, no. 05, pp. 026–026, may 2006.

[7] M. Bähr, S. Gieseke, M. Gigg, D. Grellscheid, K. Hamilton, Latunde-Da., S. Plätzer, P. Richardson, M. Seymour, A. Sherstnev, and B. Webber, "Herwig++ physics and manual," *The European Physical Journal C*, vol. 58, pp. 639–707, dec 2008.

[8] T. Gleisberg, S. Höche, F. Krauss, M. Schönherr, S. Schumann, F. Siegert, and J. Winter, "Event generation with SHERPA 1.1," *Journal of High Energy Physics*, vol. 2009, no. 02, pp. 007–007, feb 2009.

[9] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.-S. Shao, T. Stelzer, P. Torrielli, and M. Zaro, "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations," *Journal of High Energy Physics*, vol. 2014, jul 2014.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014.

[11] S. F. Novaes, "Standard model: An Introduction," in *10th Jorge Andre Swieca Summer School: Particle and Fields*, 1 1999, pp. 5–102.

[12] "Standard Model." [Online]. Available: https://www.physik.uzh.ch/groups/serra/StandardModel.html

[13] J. M. Butterworth, "The standard model: how far can it go and how can we tell?" *Philosophical Trans. of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2075, 2016.

[14] E. R. Paudel, "Problems of standard model, review," *BMC Journal of Scientific Research*, vol. 4, no. 1, pp. 65–73, Dec. 2021.

[15] P. Baldi, P. Sadowski, and D. Whiteson, "Searching for Exotic Particles in High-Energy Physics with Deep Learning," *Nature Commun.*, vol. 5, 2014.

[16] O. Cerri, T. Q. Nguyen, M. Pierini, M. Spiropulu, and J. R. Vlimant, "Variational Autoencoders for New Physics Mining at the Large Hadron Collider,"

JHEP, vol. 05, p. 036, 2019.
[17] E. Bernreuther, T. Finke, F. Kahlhoefer, M. Krämer, and A. Mück, “Casting a graph net to catch dark showers,” *SciPost Phys.*, vol. 10, no. 2, p. 046, 2021.
[18] D. Cogollo, F. F. Freitas, C. A. de S. Pires, Y. M. Oviedo-Torres, and P. Vasconcelos, “Deep learning analysis of the inverse seesaw in a 3-3-1 model at the LHC,” *Phys. Lett. B*, vol. 811, p. 135931, 2020.
[19] R. T. D’Agnolo and A. Wulzer, “Learning New Physics from a Machine,” *Phys. Rev. D*, vol. 99, no. 1, p. 015014, 2019.
[20] R. T. d’Agnolo, G. Grosso, M. Pierini, A. Wulzer, and M. Zanetti, “Learning new physics from an imperfect machine,” *Eur. Phys. J. C*, vol. 82, no. 3, p. 275, 2022.
[21] L. Lönnblad, C. Peterson, and T. Rögnvaldsson, “Finding gluon jets with a neural trigger,” *Phys. Rev. Lett.*, vol. 65, pp. 1321–1324, Sep 1990.
[22] T. Behnke and D. Charlton, “Electroweak measurements using heavy quarks at lep,” *Physica Scripta*, vol. 52, no. 2, pp. 133–157, aug 1995.
[23] C. Peterson, T. Rögnvaldsson, and L. Lönnblad, “JETNET 3.0A versatile artificial neural network package,” *Computer Physics Communications*, vol. 81, no. 1, pp. 185–220, 1994.
[24] J. Cogan, M. Kagan, E. Strauss, and A. Schwarztman, “Jet-images: Computer vision inspired techniques for jet tagging,” *JHEP*, vol. 02, p. 118, 2015.
[25] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
[26] J. Thaler and K. Van Tilburg, “Identifying Boosted Objects with N-subjettiness,” *JHEP*, vol. 03, p. 015, 2011.
[27] ——, “Maximizing Boosted Top Identification by Minimizing N-subjettiness,” *JHEP*, vol. 02, p. 093, 2012.
[28] L. G. Almeida, M. Backović, M. Cliche, S. J. Lee, and M. Perelstein, “Playing Tag with ANN: Boosted Top Identification with Pattern Recognition,” *JHEP*, vol. 07, p. 086, 2015.
[29] P. Baldi, K. Bauer, C. Eng, P. Sadowski, and D. Whiteson, “Jet Substructure Classification in High-Energy Physics with Deep Neural Networks,” *Phys. Rev. D*, vol. 93, no. 9, 2016.
[30] J. Barnard, E. N. Dawe, M. J. Dolan, and N. Rajcic, “Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks,” *Phys. Rev. D*, vol. 95, no. 1, 2017.
[31] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, and A. Schwartzman, “Jet-images — deep learning edition,” *JHEP*, vol. 07, p. 069, 2016.
[32] P. T. Komiske, E. M. Metodiev, and M. D. Schwartz, “Deep learning in color: towards automated quark/gluon jet discrimination,” *JHEP*, vol. 01, p. 110, 2017.
[33] F. A. Dreyer, G. P. Salam, and G. Soyez, “The Lund Jet Plane,” *JHEP*, vol. 12, p. 064, 2018.

[34] Y. L. Dokshitzer, G. D. Leder, S. Moretti, and B. R. Webber, “Better jet clustering algorithms,” *JHEP*, vol. 08, 1997.

[35] M. Wobisch and T. Wengler, “Hadronization corrections to jet cross-sections in deep inelastic scattering,” in *Proc. of the Workshop on Monte Carlo Generators for HERA Physics*, 4 1998, pp. 270–279.

[36] M. Cacciari, G. P. Salam, and G. Soyez, “FastJet User Manual,” *Eur. Phys. J. C*, vol. 72, p. 1896, 2012.

[37] D. Guest, J. Collado, P. Baldi, S. C. Hsu, G. Urban, and D. Whiteson, “Jet flavor classification in high-energy physics with deep neural networks,” *Phys. Rev. D*, vol. 94, Dec 2016.

[38] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

[39] ATLAS Collaboration, “Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment,” 3 2017. [Online]. Available: https://inspirehep.net/literature/1795312

[40] ——, “Optimisation and performance studies of the ATLAS b-tagging algorithms for the 2017-18 LHC run,” 7 2017. [Online]. Available: https://inspirehep.net/literature/1795306

[41] CMS Collaboration, “Heavy flavor identification at CMS with deep neural networks,” 2017. [Online]. Available: https://cds.cern.ch/record/2255736

[42] ——, “Performance of heavy flavour identification algorithms in proton-proton collisions at 13 TeV at the CMS experiment,” 2017. [Online]. Available: https://cds.cern.ch/record/2263801

[43] S. Egan, W. Fedorko, A. Lister, J. Pearkes, and C. Gay, “Long short-term memory (lstm) networks with jet constituents for boosted top tagging at the lhc,” 2017. [Online]. Available: https://arxiv.org/abs/1711.09059

[44] T. Cheng, “Recursive Neural Networks in Quark/Gluon Tagging,” *Comput. Softw. Big Sci.*, vol. 2, no. 1, p. 3, 2018.

[45] I. Henrion, J. Brehmer, J. Bruna, K. Cho, K. Cranmer, G. Louppe, and G. Rochette, “Neural message passing for jet physics,” in *Workshop on Deep Learning for Physical Sciences of the 31st Annual Conf. on Neural Information Processing Systems (NIPS17)*, 2017.

[46] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, “A simple neural network module for relational reasoning,” 2017. [Online]. Available: https://arxiv.org/abs/1706.01427

[47] P. T. Komiske, E. M. Metodiev, and J. Thaler, “Energy Flow Networks: Deep Sets for Particle Jets,” *JHEP*, vol. 01, p. 121, 2019.

[48] H. Qu and L. Gouskos, “ParticleNet: Jet Tagging via Particle Clouds,” *Phys. Rev. D*, vol. 101, no. 5, 2020.

[49] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph cnn for learning on point clouds,” *ACM Trans. Graph.*, vol. 38, no. 5, oct 2019.

[50] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” 2016. [Online]. Available: https://arxiv.org/abs/1612.00593

[51] A. J. Larkoski, I. Moult, and B. Nachman, “Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning,” *Phys. Rept.*, vol. 841, pp. 1–63, 2020.

[52] ATLAS Collaboration, “Quark versus Gluon Jet Tagging Using Jet Images with the ATLAS Detector,” 7 2017. [Online]. Available: https://inspirehep.net/literature/1795319

[53] CMS Collaboration, “New Developments for Jet Substructure Reconstruction in CMS,” 2017. [Online]. Available: https://cds.cern.ch/record/2275226

[54] J. Lin, M. Freytsis, I. Moult, and B. Nachman, “Boosting $H \to b\bar{b}$ with Machine Learning,” *JHEP*, vol. 10, p. 101, 2018.

[55] G. Kasieczka, T. Plehn, M. Russell, and T. Schell, “Deep-learning Top Taggers or The End of QCD?” *JHEP*, vol. 05, p. 006, 2017.

[56] A. Butter *et al.*, “The Machine Learning landscape of top taggers,” *SciPost Phys.*, vol. 7, p. 014, 2019.

[57] G. Louppe, K. Cho, C. Becot, and K. Cranmer, “QCD-Aware Recursive Neural Networks for Jet Physics,” *JHEP*, vol. 01, p. 057, 2019.

[58] A. Andreassen, I. Feige, C. Frye, and M. D. Schwartz, “Junipr: a framework for unsupervised machine learning in particle physics,” *Eur. Phys. J. C*, vol. 79, no. 2, p. 102, 2019.

[59] S. Chatrchyan *et al.*, “Description and performance of track and primary-vertex reconstruction with the CMS tracker,” *JINST*, vol. 9, no. 10, p. P10009, 2014.

[60] ATLAS Collaboration, “Characterization of interaction-point beam parameters using the pp event-vertex distribution reconstructed in the atlas detector at the lhc,” 5 2010. [Online]. Available: https://inspirehep.net/literature/1203962

[61] ——, “Performance of primary vertex reconstruction in proton-proton collisions at $\sqrt{s}$ =7 tev in the atlas experiment,” 7 2010. [Online]. Available: https://inspirehep.net/literature/1204019

[62] CMS Collaboration, “Pileup Removal Algorithms,” 2014. [Online]. Available: https://inspirehep.net/literature/1311934

[63] M. Cacciari and G. P. Salam, “Pileup subtraction using jet areas,” *Phys. Lett. B*, vol. 659, pp. 119–126, 2008.

[64] J. M. Butterworth, A. R. Davison, M. Rubin, and G. P. Salam, “Jet substructure as a new higgs-search channel at the large hadron collider,” *Phys. Rev. Lett.*, vol. 100, p. 242001, Jun 2008.

[65] S. D. Ellis, C. K. Vermilion, and J. R. Walsh, “Techniques for improved heavy particle searches with jet substructure,” *Phys. Rev. D*, vol. 80, p. 051501, Sep

2009.
[66] ——, “Recombination algorithms and jet substructure: Pruning as a tool for heavy particle searches,” *Phys. Rev. D*, vol. 81, p. 094023, May 2010.
[67] D. Krohn, J. Thaler, and L. T. Wang, “Jet Trimming,” *JHEP*, vol. 02, p. 084, 2010.
[68] M. Cacciari, G. P. Salam, and G. Soyez, “SoftKiller, a particle-level pileup removal method,” *Eur. Phys. J. C*, vol. 75, no. 2, p. 59, 2015.
[69] D. Bertolini, P. Harris, M. Low, and N. Tran, “Pileup Per Particle Identification,” *JHEP*, vol. 10, p. 059, 2014.
[70] P. Berta, M. Spousta, D. W. Miller, and R. Leitner, “Particle-level pileup subtraction for jets and jet shapes,” *JHEP*, vol. 06, p. 092, 2014.
[71] ATLAS Collaboration, “Constituent-level pile-up mitigation techniques in ATLAS,” 8 2017. [Online]. Available: https://inspirehep.net/literature/1620091
[72] P. T. Komiske, E. M. Metodiev, B. Nachman, and M. D. Schwartz, “Pileup Mitigation with Machine Learning (PUMML),” *JHEP*, vol. 12, p. 051, 2017.
[73] J. Arjona Martínez, O. Cerri, M. Pierini, M. Spiropulu, and J. R. Vlimant, “Pileup mitigation at the Large Hadron Collider with graph neural networks,” *Eur. Phys. J. Plus*, vol. 134, no. 7, p. 333, 2019.
[74] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” 2014. [Online]. Available: https://arxiv.org/abs/1409.1259
[75] B. Maier, S. M. Narayanan, G. de Castro, M. Goncharov, C. Paus, and M. Schott, “Pile-up mitigation using attention,” *Mach. Learn. Sci. Tech.*, vol. 3, no. 2, p. 025012, 2022.
[76] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.
[77] S. Carrazza and F. A. Dreyer, “Jet Grooming through Reinforcement Learning,” *J. Phys. Conf. Ser.*, vol. 1525, p. 012111, 2020.
[78] T. Li, S. Liu, Y. Feng, G. Paspalaki, N. Tran, M. Liu, and P. Li, “Semi-supervised Graph Neural Networks for Pileup Noise Removal,” 3 2022. [Online]. Available: https://arxiv.org/abs/2203.15823
[79] S. J. Brice, “The results of a neural network statistical event class analysis,” 1996. [Online]. Available: https://sno.phy.queensu.ca/str/SNO-STR-96-001.pdf
[80] D. S. Ayres *et al.*, “The NOvA Technical Design Report,” 10 2007. [Online]. Available: https://inspirehep.net/files/1e897a237c85bae0087a7f644e9ad832
[81] A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M. D. Messier, E. Niner, G. Pa-

wloski, F. Psihas, A. Sousa, and P. Vahle, “A Convolutional Neural Network Neutrino Event Classifier,” *JINST*, vol. 11, no. 09, p. P09001, 2016.

[82] C. Szegedy, L. Wei, J. Yangqing *et al.*, “Going deeper with convolutions,” in *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[83] P. Adamson, L. Aliaga, D. Ambrose *et al.*, “Constraints on oscillation parameters from ν_e appearance and ν_μ disappearance in nova,” *Phys. Rev. Lett.*, vol. 118, p. 231801, Jun 2017.

[84] J. Renner *et al.*, “Background rejection in NEXT using deep neural networks,” *JINST*, vol. 12, no. 01, p. T01004, 2017.

[85] J. Schechter and J. W. F. Valle, “Neutrinoless Double beta Decay in SU(2) x U(1) Theories,” *Phys. Rev. D*, vol. 25, p. 2951, 1982.

[86] E. Racah, S. Ko, P. Sadowski, W. Bhimji, C. Tull, S. Oh, P. Baldi, and Prabhat, “Revealing Fundamental Physics from the Daya Bay Neutrino Experiment using Deep Neural Networks,” 1 2016. [Online]. Available: https://arxiv.org/abs/1601.07621

[87] R. Acciarri *et al.*, “Convolutional Neural Networks Applied to Neutrino Events in a Liquid Argon Time Projection Chamber,” *JINST*, vol. 12, no. 03, p. P03011, 2017.

[88] S. Aiello *et al.*, “Event reconstruction for KM3NeT/ORCA using convolutional neural networks,” *JINST*, vol. 15, no. 10, p. P10005, 2020.

[89] B. Abi *et al.*, “Neutrino interaction classification with a convolutional neural network in the DUNE far detector,” *Phys. Rev. D*, vol. 102, no. 9, p. 092003, 2020.

[90] P. Abratenko *et al.*, “Convolutional neural network for multiple particle identification in the MicroBooNE liquid argon time projection chamber,” *Phys. Rev. D*, vol. 103, no. 9, p. 092003, 2021.

[91] L. Hertel, L. Li, P. Baldi, and J. Bian, “Convolutional neural networks for electron neutrino and electron shower energy reconstruction in the nova detectors,” in *Proc. of the Workshop on Deep Learning for Physical Sciences of the 31st Annual Conf. on Neural Information Processing Systems*, 2017.

[92] J. Liu, J. Ott, J. Collado, B. Jargowsky, W. Wu, J. Bian, and P. Baldi, “Deep-Learning-Based Kinematic Reconstruction for DUNE,” 12 2020. [Online]. Available: https://arxiv.org/abs/2012.06181

[93] L. Dominé, P. C. de Soux, F. Drielsma, D. H. Koh, R. Itay, Q. Lin, K. Terao, K. V. Tsang, and T. L. Usher, “Point proposal network for reconstructing 3D particle endpoints with subpixel precision in liquid argon time projection chambers,” *Phys. Rev. D*, vol. 104, no. 3, p. 032004, 2021.

[94] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[95] C. Adams *et al.*, “Deep neural network for pixel-level electromagnetic particle identification in the MicroBooNE liquid argon time projection

chamber," *Phys. Rev. D*, vol. 99, no. 9, 2019.

[96] H. Yu *et al.*, "Augmented signal processing in Liquid Argon Time Projection Chambers with a deep neural network," *JINST*, vol. 16, no. 01, p. P01036, 2021.

[97] R. Acciarri *et al.*, "A deep-learning based raw waveform region-of-interest finder for the liquid argon time projection chamber," *JINST*, vol. 17, no. 01, p. P01018, 2022.

[98] B. Graham, M. Engelcke, and L. van der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 9224–9232.

[99] F. Scarselli, M. Gori *et al.*, "The graph neural network model," *IEEE Trans. on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[100] L. Dominé and K. Terao, "Scalable deep convolutional neural networks for sparse, locally dense liquid argon time projection chamber data," *Phys. Rev. D*, vol. 102, no. 1, p. 012005, 2020.

[101] D. H. Koh, P. Côte De Soux, L. Dominé, F. Drielsma, R. Itay, Q. Lin, K. Terao, K. V. Tsang, and T. L. Usher, "Scalable, Proposal-free Instance Segmentation Network for 3D Pixel Clustering and Particle Trajectory Reconstruction in Liquid Argon Time Projection Chambers," 7 2020. [Online]. Available: https://arxiv.org/abs/2007.03083

[102] N. Choma, F. Monti, L. Gerhardt *et al.*, "Graph neural networks for icecube signal classification," in *Proc. of the IEEE Int. Conf. on Machine Learning and Applications*, 2018, pp. 386–391.

[103] Z. Qian *et al.*, "Vertex and energy reconstruction in JUNO with machine learning methods," *Nucl. Instrum. Meth. A*, vol. 1010, p. 165527, 2021.

[104] F. Drielsma, Q. Lin, P. C. de Soux, L. Dominé, R. Itay, D. H. Koh, B. J. Nelson, K. Terao, K. V. Tsang, and T. L. Usher, "Clustering of electromagnetic showers and particle interactions with graph neural networks in liquid argon time projection chambers," *Phys. Rev. D*, vol. 104, no. 7, p. 072004, 2021.

[105] S. Alonso-Monsalve, D. Douqa, C. Jesús-Valls, T. Lux, S. Pina-Otey, F. Sánchez, D. Sgalaberna, and L. H. Whitehead, "Graph neural network for 3D classification of ambiguities and optical crosstalk in scintillator-based neutrino detectors," *Phys. Rev. D*, vol. 103, no. 3, p. 032005, 2021.

[106] J. Hewes *et al.*, "Graph Neural Network for Object Reconstruction in Liquid Argon Time Projection Chambers," *Proc. of the EPJ Web Conf.*, vol. 251, p. 03054, 2021.

[107] C. Adams, K. Terao, and T. Wongjirad, "PILArNet: Public Dataset for Particle Imaging Liquid Argon Detectors in High Energy Physics," 6 2020. [Online]. Available: https://arxiv.org/abs/2006.01993

[108] J. Martín-Albo *et al.*, "Sensitivity of NEXT-100 to Neutrinoless Double Beta Decay," *JHEP*, vol. 05, p. 159, 2016.

[109] P. Abratenko *et al.*, "Semantic segmentation with a sparse convolutional

neural network for event reconstruction in MicroBooNE," *Phys. Rev. D*, vol. 103, no. 5, p. 052012, 2021.

[110] S. Y.-C. Chen, T.-C. Wei, C. Zhang, H. Yu, and S. Yoo, "Quantum convolutional neural networks for high energy physics data analysis," *Phys. Rev. Res.*, vol. 4, no. 1, p. 013231, 2022.

[111] B. Clerbaux, M. C. Molla, P. A. Petitjean, Y. Xu, and Y. Yang, "Study of Using Machine Learning for Level 1 Trigger Decision in JUNO Experiment," *IEEE Trans. Nucl. Sci.*, vol. 68, no. 8, pp. 2187–2193, 2021.

[112] R. Abbasi *et al.*, "A Convolutional Neural Network based Cascade Reconstruction for the IceCube Neutrino Observatory," *JINST*, vol. 16, 2021.

[113] R. Frühwirth and R. K. Bock, *Data analysis techniques for high-energy physics experiments*, H. Grote, D. Notz, and M. Regler, Eds. Cambridge University Press, 2000, vol. 11.

[114] F. Ragusa and L. Rolandi, "Tracking at LHC," *New J. Phys.*, vol. 9, p. 336, 2007.

[115] R. Frühwirth and A. Strandlie, "Pattern recognition and reconstruction: Datasheet from Landolt-Börnstein - Group I elementary particles, nuclei and atoms." [Online]. Available: https://materials.springer.com/lb/docs/sm_lbs_978-3-642-03606-4_13

[116] H. Kälviäinen, P. Hirvonen, L. Xu, and E. Oja, "Probabilistic and non-probabilistic hough transforms: overview and comparisons," *Image Vis. Comput.*, vol. 13, no. 4, pp. 239–252, 1995.

[117] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 03 1960.

[118] D. E. Catlin, "The discrete Kalman filter," in *Estimation, Control, and the Discrete Kalman Filter*. New York, NY: Springer New York, 1989, pp. 133–163.

[119] R. Fruhwirth, "Application of Kalman filtering to track and vertex fitting," *Nucl. Instrum. Meth. A*, vol. 262, pp. 444–450, 1987.

[120] R. Mankel, "A Concurrent track evolution algorithm for pattern recognition in the HERA-B main tracking system," *Nucl. Instrum. Meth. A*, vol. 395, pp. 169–184, 1997.

[121] B. Schmidt, "The High-Luminosity upgrade of the LHC: Physics and Technology Challenges for the Accelerator and the Experiments," *J. Phys. Conf. Ser.*, vol. 706, no. 2, p. 022002, 2016.

[122] The HEP.TrkX Collaboration, "Hep advanced tracking algorithms with cross-cutting applications," 2016. [Online]. Available: https://heptrkx.github.io

[123] S. Farrell, D. Anderson, P. Calafiura, G. Cerati, L. Gray, J. Kowalkowski, M. Mudigonda, Prabhat, P. Spentzouris, M. Spiropoulou, A. Tsaris, J. R. Vlimant, and S. Zheng, "The hep.trkx project: deep neural networks for hl-lhc

online and offline tracking," *Proc. of the EPJ Web Conf.*, vol. 150, p. 00003, 2017.

[124] A. Tsaris *et al.*, "The hep.trkx project: Deep learning for particle tracking," *Journal of Physics: Conf. Series*, vol. 1085, no. 4, p. 042023, sep 2018.

[125] S. Farrell, P. Calafiura, M. Mudigonda, Prabhat, D. Anderson, J.-R. Vlimant, S. Zheng, J. Bendavid, M. Spiropulu, G. Cerati, L. Gray, J. Kowalkowski, P. Spentzouris, and A. Tsaris, "Novel deep learning methods for track reconstruction," in *Proc. of the Int. Workshop Connecting the Dots*, 10 2018.

[126] S. e. o. Amrouche, "The tracking machine learning challenge: Accuracy phase," in *The NeurIPS '18 Competition*, 2020, pp. 231–264.

[127] The Exa.TrkX Collaboration, "Hep advanced tracking algorithms at the exascale," 2019. [Online]. Available: https://exatrkx.github.io

[128] X. Ju *et al.*, "Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors," in *Proc. of the Annual Conf. on Neural Information Processing Systems*, 3 2020.

[129] N. Choma *et al.*, "Track Seeding and Labelling with Embedded-space Graph Neural Networks," in *Proc.of the Connecting the Dots Workshop*, 6 2020.

[130] X. Ju *et al.*, "Performance of a geometric deep learning pipeline for HL-LHC particle tracking," *Eur. Phys. J. C*, vol. 81, no. 10, p. 876, 2021.

[131] A. Elabd *et al.*, "Graph Neural Networks for Charged Particle Tracking on FPGAs," *Front. Big Data*, vol. 5, 2022.