A Living Review of Machine Learning for Particle Physics

Matthew Feickert\textsuperscript{a} and Benjamin Nachman\textsuperscript{b}, for the Inter-Experimental LHC Machine Learning Working Group

\textsuperscript{a}Department of Physics, University of Illinois at Urbana Champaign
\textsuperscript{b}Physics Division, Lawrence Berkeley National Laboratory

E-mail: matthew.feickert@cern.ch, bpnachman@lbl.gov

ABSTRACT: Modern machine learning techniques, including deep learning, are rapidly being applied, adapted, and developed for high energy physics. The goal of this document is to provide a nearly comprehensive list of citations for those developing and applying these approaches to experimental, phenomenological, or theoretical analyses. As a living document, it will be updated as often as possible to incorporate the latest developments. A list of proper (unchanging) reviews can be found within. Papers are grouped into a small set of topics to be as useful as possible. Suggestions are most welcome.
The purpose of this note is to collect references for modern machine learning as applied to particle physics. A minimal number of categories is chosen in order to be as useful as possible. Note that papers may be referenced in more than one category. The fact that a paper is listed in this document does not endorse or validate its content - that is for the community (and for peer-review) to decide. Furthermore, the classification here is a best attempt and may have flaws - please let us know if (a) we have missed a paper you think should be included, (b) a paper has been misclassified, or (c) a citation for a paper is not correct or if the journal information is now available. In order to be as useful as possible, this document will continue to evolve so please check back before you write your next paper. You can simply download the .bib file to get all of the latest references.

- **Reviews**

  Below are links to many (static) general and specialized reviews. The third bullet contains links to classic papers that applied shallow learning methods many decades before the deep learning revolution.
  
  - Modern reviews [1–6]
  - Specialized reviews [7–19]
  - Classical papers [20, 21]

- **Classification**

  Given a feature space \( x \in \mathbb{R}^n \), a binary classifier is a function \( f : \mathbb{R}^n \rightarrow [0, 1] \), where 0 corresponds to features that are more characteristic of the zeroth class (e.g. background) and 1 correspond to features that are more characteristic of the one class (e.g. signal). Typically, \( f \) will be a function specified by some parameters \( w \) (e.g. weights and biases of a neural network) that are determined by minimizing a loss of the form \( L[f] = \sum_i \ell(f(x_i), y_i) \), where \( y_i \in \{0, 1\} \) are labels. The function \( \ell \) is smaller when \( f(x_i) \) and \( y_i \) are closer. Two common loss functions are the mean squared error \( \ell(x, y) = (x - y)^2 \) and the binary cross entropy \( \ell(x, y) = y \log(x) + (1-y) \log(1-x) \). Exactly what ‘more characteristic of’ means depends on the loss function used to determine \( f \). It is also possible to make a multi-class classifier. A common strategy for the multi-class case is to represent each class as a different basis vector in \( \mathbb{R}^{n_{\text{classes}}} \) and then \( f(x) \in [0, 1]^{n_{\text{classes}}} \). In this case, \( f(x) \) is usually restricted to have its \( n_{\text{classes}} \) components sum to one and the loss function is typically the cross entropy \( \ell(x, y) = \sum_{i} y_i \log(x) \).

---

1See [https://github.com/iml-wg/HEPML-LivingReview](https://github.com/iml-wg/HEPML-LivingReview)
- Parameterized classifiers \[22, 23\].
A classifier that is conditioned on model parameters \( f(x|\theta) \) is called a parameterized classifier.

- Representations
There is no unique way to represent high energy physics data. It is often natural to encode \( x \) as an image or another one of the structures listed below.

* Jet images \[24–37\]
Jets are collimated sprays of particles. They have a complex radiation pattern and such, have been a prototypical example for many machine learning studies. See the next item for a specific description about images.

* Event images \[29, 38–41\]
A grayscale image is a regular grid with a scalar value at each grid point. ‘Color’ images have a fixed-length vector at each grid point. Many detectors are analogous to digital cameras and thus images are a natural representation. In other cases, images can be created by discretizing. Convolutional neural networks are natural tools for processing image data. One downside of the image representation is that high energy physics data tend to be sparse, unlike natural images.

* Sequences \[38, 42, 43\]
Data that have a variable with a particular order may be represented as a sequence. Recurrent neural networks are natural tools for processing sequence data.

* Trees \[44, 45\]
Recursive neural networks are natural tools for processing data in a tree structure.

* Graphs \[9, 46–62\]
A graph is a collection of nodes and edges. Graph neural networks are natural tools for processing data in a tree structure.

* Sets (point clouds) \[63–69\]
A point cloud is a (potentially variable-size) set of points in space. Sets are distinguished from sequences in that there is no particular order (i.e. permutation invariance). Sets can also be viewed as graphs without edges and so graph methods that can parse variable-length inputs may also be appropriate for set learning, although there are other methods as well.

* Physics-inspired basis \[70, 74\]
This is a catch-all category for learning using other representations that use some sort of manual or automated physics-preprocessing.

- Targets

* W/Z tagging [27, 31, 34, 56, 62, 75, 76]
Boosted, hadronically decaying W and Z bosons form jets that are distinguished from generic quark and gluon jets by their mass near the boson mass and their two-prong substructure.

* H → bb [29, 41, 51, 59, 75, 77–80]
Due to the fidelity of b-tagging, boosted, hadronically decaying Higgs bosons (predominantly decaying to bb) has unique challenges and opportunities compared with W/Z tagging.

* quarks and gluons [28, 32, 37, 45, 49, 62, 81–85]
Quark jets tend to be narrower and have fewer particles than gluon jets. This classification task has been a benchmark for many new machine learning models.

* top quark tagging [7, 26, 33, 34, 52, 62, 74, 81, 86–88]
Boosted top quarks form jets that have a three-prong substructure (t → Wb, W → q̅q).

* strange jets [89–91]
Strange quarks have a very similar fragmentation to generic quark and gluon jets, so this is a particularly challenging task.

* b-tagging [42, 43, 92, 93]
Due to their long (but not too long) lifetime, the B-hadron lifetime is macroscopic and b-jet tagging has been one of the earliest adapters of modern machine learning tools.

* Flavor physics [94]
This category is for studies related to exclusive particle decays, especially with bottom and charm hadrons.

* BSM particles [51, 54, 77, 95–103]
There are many proposals to train classifiers to enhance the presence of particular new physics models.

* Particle identification [50, 104, 108]
This is a generic category for direct particle identification and categorization using various detector technologies. Direct means that the particle directly interacts with the detector (in contrast with b-tagging).
Neutrino Detectors \[10, 58, 109, 122\]

Neutrino detectors are very large in order to have a sizable rate of neutrino detection. The entire neutrino interaction can be characterized to distinguish different neutrino flavors.

Direct Dark Matter Detectors \[123, 124\]

Dark matter detectors are similar to neutrino detectors, but aim to achieve ‘zero’ background.

Astro Particle and Cosmic Ray physics \[61, 125, 127\]

Machine learning is often used in astrophysics and cosmology in different ways than terrestrial particle physics experiments due to a general divide between Bayesian and Frequentist statistics. However, there are many similar tasks and a growing number of proposals designed for one domain that apply to the other.

Tracking \[47, 57, 66, 128, 133\]

Charged particle tracking is a challenging pattern recognition task. This category is for various classification tasks associated with tracking, such as seed selection.

Heavy ions \[82, 134, 135\]

Many tools in high energy nuclear physics are similar to high energy particle physics. The physics target of these studies are to understand collective properties of the strong force.

– Learning strategies

There is no unique way to train a classifier and designing an effective learning strategy is often one of the biggest challenges for achieving optimality.

Hyperparameters \[136\]

In addition to learnable weights $w$, classifiers have a number of non-differentiable parameters like the number of layers in a neural network. These parameters are called hyperparameters.

Weak supervision \[30, 85, 137, 148\]

For supervised learning, the labels $y_i$ are known. In the case that the labels are noisy or only known with some uncertainty, then the learning is called weak supervision. Semi-supervised learning is the related case where labels are known for only a fraction of the training examples.

Unsupervised \[149, 153\]

When no labels are provided, the learning is called unsupervised.

Reinforcement Learning \[154, 156\]
Instead of learning to distinguish different types of examples, the goal of reinforcement learning is to learn a strategy (policy). The prototypical example of reinforcement learning is learning a strategy to play video games using some kind of score as a feedback during the learning.

* **Quantum Machine Learning** [122, 157, 161]
  Quantum computers are based on unitary operations applied to quantum states. These states live in a vast Hilbert space which may have a usefully large information capacity for machine learning.

* **Feature ranking** [102]
  It is often useful to take a set of input features and rank them based on their usefulness.

  - **Fast inference / deployment**
    There are many practical issues that can be critical for the actual application of machine learning models.
    
    * **Software** [38, 97, 163, 166]
      Strategies for efficient inference for a given hardware architecture.
    
    * **Hardware/firmware** [55, 60, 167, 173]
      Various accelerators have been studied for fast inference that is very important for latency-limited applications like the trigger at collider experiments.
    
    * **Deployment** [174]
      This category is for the deployment of machine learning interfaces, such as in the cloud.

* **Regression**
  In contrast to classification, the goal of regression is to learn a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ for input features $x \in \mathbb{R}^n$ and target features $y \in \mathbb{R}^m$. The learning setup is very similar to classification, where the network architectures and loss functions may need to be tweaked. For example, the mean squared error is the most common loss function for regression, but the network output is no longer restricted to be between 0 and 1.

  - **Pileup** [38, 48, 154, 175]
    A given bunch crossing at the LHC will have many nearly simultaneous proton-proton collisions. Only one of those is usually interesting and the rest introduce a source of noise (pileup) that must be mitigating for precise final state reconstruction.
– Calibration 106 176 181
The goal of calibration is to remove the bias (and reduce variance if possible) from detector effects.

– Recasting 182 184
Even though an experimental analysis may provide a single model-dependent interpretation of the result, the results are likely to have important implications for a variety of other models. Recasting is the task of taking a result and interpreting it in the context of a model that was not used for the original analysis.

– Matrix elements 185 187
Regression methods can be used as surrogate models for functions that are too slow to evaluate. One important class of functions are matrix elements, which form the core component of cross section calculations in quantum field theory.

– Parameter estimation 188 190
The target features could be parameters of a model, which can be learned directly through a regression setup. Other forms of inference are described in later sections (which could also be viewed as regression).

– Parton Distribution Functions (and related) 173 191 192
Various machine learning models can provide flexible function approximators, which can be useful for modeling functions that cannot be determined easily from first principles such as parton distribution functions.

• Decorrelation methods 96 193 207
It it sometimes the case that a classification or regression model needs to be independent of a set of features (usually a mass-like variable) in order to estimate the background or otherwise reduce the uncertainty. These techniques are related to what the machine learning literature calls model ‘fairness’.

• Generative models / density estimation
The goal of generative modeling is to learn (explicitly or implicitly) a probability density $p(x)$ for the features $x \in \mathbb{R}^n$. This task is usually unsupervised (no labels).

– GANs: 106 107 208 244
Generative Adversarial Networks 247 learn $p(x)$ implicitly through the minimax optimization of two networks: one that maps noise to structure $G(z)$ and one a classifier (called the discriminator) that learns to distinguish examples generated from $G(z)$ and those generated from the target process.
When the discriminator is maximally ‘confused’, then the generator is effectively mimicking $p(x)$.

- **Autoencoders** [223 246 248]
  An autoencoder consists of two functions: one that maps $x$ into a latent space $z$ (encoder) and a second one that maps the latent space back into the original space (decoder). The encoder and decoder are simultaneously trained so that their composition is nearly the identity. When the latent space has a well-defined probability density (as in variational autoencoders), then one can sample from the autoencoder by applying the detector to a randomly chosen element of the latent space.

- **Normalizing flows** [249 258]
  Normalizing flows [259] learn $p(x)$ explicitly by starting with a simple probability density and then applying a series of bijective transformations with tractable Jacobians.

- **Physics-inspired** [244 260 262]
  A variety of methods have been proposed to use machine learning tools (e.g. neural networks) combined with physical components.

- **Mixture Models** [263]
  A mixture model is a superposition of simple probability densities. For example, a Gaussian mixture model is a sum of normal probability densities. Mixture density networks are mixture models where the coefficients in front of the constituent densities as well as the density parameters (e.g. mean and variances of Gaussians) are parameterized by neural networks.

- **Phase space generation** [252 254 264 270]
  Monte Carlo event generators integrate over a phase space that needs to be generated efficiently and this can be aided by machine learning methods.

- **Gaussian processes** [183 186 271]
  These are non-parametric tools for modeling the ‘time’-dependence of a random variable. The ‘time’ need not be actual time - for instance, one can use Gaussian processes to model the energy dependence of some probability density.

- **Anomaly detection** [139 140 145 146 151 152 247 255 272 299]
  The goal of anomaly detection is to identify abnormal events. The abnormal events could be from physics beyond the Standard Model or from faults in a detector. While nearly all searches for new physics are technically anomaly detection, this
category is for methods that are mode-independent (broadly defined). Anomalies in high energy physics tend to manifest as over-densities in phase space (often called ‘population anomalies’) in contrast to off-manifold anomalies where you can flag individual examples as anomalous.

• Simulation-based (‘likelihood-free’) Inference
Likelihood-based inference is the case where $p(x|\theta)$ is known and $\theta$ can be determined by maximizing the probability of the data. In high energy physics, $p(x|\theta)$ is often not known analytically, but it is often possible to sample from the density implicitly using simulations.

  – Parameter estimation [23, 258, 300, 309]
  
  This can also be viewed as a regression problem, but there the goal is typically to do maximum likelihood estimation in contrast to directly minimizing the mean squared error between a function and the target.

  – Unfolding [214, 227, 310, 317]
  
  This is the task of removing detector distortions. In contrast to parameter estimation, the goal is not to infer model parameters, but instead, the undistorted phase space probability density. This is often also called deconvolution.

  – Domain adaptation [23, 240, 300, 318]
  
  Morphing simulations to look like data is a form of domain adaptation.

  – BSM [283, 286, 302, 306]
  
  This category is for parameter estimation when the parameter is the signal strength of new physics.

• Uncertainty Quantification
Estimating and mitigating uncertainty is essential for the successful deployment of machine learning methods in high energy physics.

  – Interpretability [27, 36, 319, 320]
  
  Machine learning methods that are interpretable maybe more robust and thus less susceptible to various sources of uncertainty.

  – Estimation [31, 321, 322]
  
  A first step in reducing uncertainties is estimating their size.

  – Mitigation [193, 202, 323]
  
  This category is for proposals to reduce uncertainty.

  – Uncertainty-aware inference [324, 327]
  
  The usual path for inference is that a machine learning method is trained for
a nominal setup. Uncertainties are then propagated in the usual way. This is suboptimal and so there are multiple proposals for incorporating uncertainties into the learning to get as close to making the final statistical test the target of the machine learning as possible.

- **Experimental results**
  
  This section is incomplete as there are many results that directly and indirectly (e.g. via flavor tagging) use modern machine learning techniques. We will try to highlight experimental results that use deep learning in a critical way for the final analysis sensitivity.

  - Final analysis discriminate for searches [145, 328–330].
References


[29] J. Lin, M. Freytsis, I. Moult and B. Nachman, Boosting \( H \rightarrow b\bar{b} \) with Machine Learning, JHEP 10 (2018) 101. 1807.10768. 2, 3


[34] S. Macaluso and D. Shih, Pulling Out All the Tops with Computer Vision and Deep Learning, JHEP 10 (2018) 121, 1803.00107.


[82] Y.-T. Chien and R. Kunnawalkam Elayavalli, Probing heavy ion collisions using quark and gluon jet substructure, 1803.03589


Anomalies: Standardized and Future-proof Model Selection Using Self-normalizing Neural Networks, 2008.04316


[107] D. Belayneh et al., Calorimetry with Deep Learning: Particle Simulation and Reconstruction for Collider Physics, 1912.06794.


[124] LUX collaboration, D. Akerib et al., Improving sensitivity to low-mass dark matter in LUX using a novel electrode background mitigation technique, \[2011.09602\].


[127] Y.-L. S. Tsai, Y.-L. Chung, Q. Yuan and K. Cheung, Inverting cosmic ray propagation by Convolutional Neural Networks, \[2011.11930\].


[129] S. Farrell et al., Novel deep learning methods for track reconstruction, 10, 2018. \[1810.06111\]

[130] S. Amrouche et al., The Tracking Machine Learning challenge : Accuracy phase, \[1904.06778\].


[157] A. Blance and M. Spannowsky, Quantum Machine Learning for Particle Physics using a Variational Quantum Classifier, [2010.07335]


[166] D. Bourgeois, C. Fitzpatrick and S. Stahl, Using holistic event information in the trigger, [1808.00711]


[168] V. Loncar et al., Compressing deep neural networks on FPGAs to binary and ternary precision with HLS4ML, [2003.06308]


[180] CMS collaboration, A. M. Sirunyan et al., *A deep neural network for simultaneous estimation of b jet energy and resolution*, 1912.06046


[194] J. Dolen, P. Harris, S. Marzani, S. Rappoccio and N. Tran, Thinking outside the ROCs: Designing Decorrelated Taggers (DDT) for jet substructure, JHEP 05 (2016) 156


[199] ATLAS collaboration, Performance of mass-decorrelated jet substructure observables for hadronic two-body decay tagging in ATLAS, ATL-PHYS-PUB-2018-014


[243] A. Maevskiy, F. Ratnikov, A. Zinchenko and V. Riabov, *Simulating the Time*
Projection Chamber responses at the MPD detector using Generative Adversarial Networks, 2012.04595


[296] M. van Beekveld, S. Caron, L. Hendriks, P. Jackson, A. Leinweber, S. Otten et al., Combining outlier analysis algorithms to identify new physics at the LHC, 2010.07940


[307] A. Andreassen, S.-C. Hsu, B. Nachman, N. Suaysom and A. Suresh, Parameter Estimation using Neural Networks in the Presence of Detector Effects, 2010.03569


[328] ATLAS collaboration, G. Aad et al., *Search for non-resonant Higgs boson pair production in the \(b\bar{b}\ell\ell\nu\nu\) final state with the ATLAS detector in \(pp\) collisions at \(\sqrt{s} = 13\) TeV*, *Phys. Lett. B* 801 (2020) 135145, 1908.06765. 9


[330] CMS collaboration, A. M. Sirunyan et al., *Inclusive search for highly boosted Higgs bosons decaying to bottom quark-antiquark pairs in proton-proton collisions at \(\sqrt{s} = 13\) TeV*, 2006.13251. 9