A Living Review of Machine Learning for Particle and Nuclear Physics

ABSTRACT: Modern machine learning techniques, including deep learning, are rapidly being applied, adapted, and developed for high energy particle and nuclear 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 and nuclear 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. Please consider citing Ref. [1] when referring to this living review.

- **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 [2–10]
  - Specialized reviews [11–26]
  - Classical papers [27, 28]
  - Datasets [29–33]

- **Classification**
  Given a feature space $x \in \mathbb{R}^n$, a binary classifier is a function $f : \mathbb{R}^n \to [0, 1]$, 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

---

1See [https://github.com/iml-wg/HEPML-LivingReview](https://github.com/iml-wg/HEPML-LivingReview)
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 \( \ell(x,y) = \sum_{i \in \text{classes}} y_i \log(x) \).

- **Parameterized classifiers** [34][36]. 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** [37][53]
    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** [12][34][60]
    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** [25][54][61][63]
    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** [65][67]
    Recursive neural networks are natural tools for processing data in a tree structure.

  * **Graphs** [13][68][101]
    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)** [51][102][112]
    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** [13][18]
  
  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** [40][44][65][80][86][119][121]
    
    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 → b̅b** [42][57][75][83][119][122][127]
    
    Due to the fidelity of b-tagging, boosted, hadronically decaying Higgs bosons (predominantly decaying to b̅b) has unique challenges and opportunities compared with W/Z tagging.

  * **quarks and gluons** [41][45][50][53][66][73][86][128][134]
    
    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** [11][39][46][47][59][76][86][117][128][135][138]
    
    Boosted top quarks form jets that have a three-prong substructure (t → Wb, W → q̅q).

  * **strange jets** [139][141]
    
    Strange quarks have a very similar fragmentation to generic quark and gluon jets, so this is a particularly challenging task.

  * **b-tagging** [61][62][64][112][142][143]
    
    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** [144]
    
    This category is for studies related to exclusive particle decays, especially with bottom and charm hadrons.

  * **BSM particles and models** [75][78][122][145][170]
    
    There are many proposals to train classifiers to enhance the presence of particular new physics models.

  * **Particle identification** [74][171][177]
    
    This is a generic category for direct particle identification and cate-
gorization using various detector technologies. Direct means that the particle directly interacts with the detector (in contrast with b-tagging).

* **Neutrino Detectors** [14, 82, 87, 90, 91, 96, 178–203]
   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** [204–211]
   Dark matter detectors are similar to neutrino detectors, but aim to achieve ‘zero’ background.

* **Cosmology, Astro Particle, and Cosmic Ray physics** [85, 212–234]
   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. See also https://github.com/georgestein/ml-in-cosmology.

* **Tracking** [63, 69, 81, 89, 92, 93, 105, 235–247]
   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 / Nuclear Physics** [52, 58, 129, 248–271]
   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** [272, 273]
   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/Semi supervision** [43, 132, 274, 286]
   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** [287, 293]
  When no labels are provided, the learning is called unsupervised.

* **Reinforcement Learning** [294, 299]
  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** [24, 193, 300–313]
  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** [118, 314]
  It is often useful to take a set of input features and rank them based on their usefulness.

* **Attention** [63]
  This is an ML tool for helping the network to focus on particularly useful features.

* **Regularization** [315]
  This is a term referring to any learning strategy that improves the robustness of a classifier to statistical fluctuations in the data and in the model initialization.

* **Optimal Transport** [288, 291, 316–318]
  Optimal transport is a set of tools for transporting one probability density into another and can be combined with other strategies for classification, regression, etc. The above citation list does not yet include papers using optimal transport distances as part of generative model training.

---

**Fast inference / deployment**

There are many practical issues that can be critical for the actual application of machine learning models.

* **Software** [54, 60, 147, 216, 247, 319, 325]
  Strategies for efficient inference for a given hardware architecture.

* **Hardware/firmware** [79, 84, 101, 326, 340]
  Various accelerators have been studied for fast inference that is very important for latency-limited applications like the trigger at collider experiments.
Deployment

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

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

The goal of calibration is to remove the bias (and reduce variance if possible) from detector (or related) effects.

- Recasting

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

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

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)

Various machine learning models can provide flexible function approxima-
tors, which can be useful for modeling functions that cannot be determined easily from first principles such as parton distribution functions.

- **Lattice Gauge Theory** [386, 396]
  Lattice methods offer a complementary approach to perturbation theory. A key challenge is to create approaches that respect the local gauge symmetry (equivariant networks).

- **Function Approximation** [397–401]
  Approximating functions that obey certain (physical) constraints.

- **Symbolic Regression** [402]
  Regression where the result is a (relatively) simple formula.

- **Decorrelation methods** [146, 403–420]
  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**: [173, 174, 311, 323, 382, 421–472]
    Generative Adversarial Networks [473] 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** [95, 100, 292, 436, 472, 474–483]
    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** [233, 386, 390, 463, 465, 483–499]
    Normalizing flows [500] learn \( p(x) \) explicitly by starting with a simple prob-
ability density and then applying a series of bijective transformations with tractable Jacobians.

- Physics-inspired [458, 501, 504]
  A variety of methods have been proposed to use machine learning tools (e.g. neural networks) combined with physical components.

- Mixture Models [177, 505, 506]
  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 [372, 486-488, 507-514]
  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 [362, 366, 515, 516]
  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 [29, 30, 32, 97, 210, 224, 276, 277, 282, 283, 289, 290, 306, 313, 316, 340, 420, 475, 478, 483, 489, 496, 517-561]
  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 [35, 36, 232, 465, 492, 502, 574]
    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** 292 427 440 497 575 587
  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** 33 454 562 588 589
  Morphing simulations to look like data is a form of domain adaptation.

- **BSM** 316 528 564 568
  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** 40 118 133 135 481 590 592
    Machine learning methods that are interpretable maybe more robust and thus less susceptible to various sources of uncertainty.

  - **Estimation** 14 593 595
    A first step in reducing uncertainties is estimating their size.

  - **Mitigation** 315 403 412 596
    This category is for proposals to reduce uncertainty.

  - **Uncertainty- and inference-aware learning** 597 602
    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.
– Searches and measurements were ML reconstruction is a core component [603, 604].

– Final analysis discriminate for searches [585, 605, 607].

– Measurements using deep learning directly (not through object reconstruction) [585].
References


[33] Y. Chen et al., *A FAIR and AI-ready Higgs Boson Decay Dataset*, 2108.02214


[59] M. Andrews et al., End-to-End Jet Classification of Boosted Top Quarks with the CMS Open Data, 2104.14659.


[64] ATLAS collaboration, Identification of Jets Containing b-Hadrons with Recurrent


[66] T. Cheng, Recursive Neural Networks in Quark/Gluon Tagging, 1711.02633


[78] E. Bernreuther, T. Finke, F. Kahlhoefer, M. Krämer and A. Mück, Casting a graph net to catch dark showers, 2006.08639


[101] A. Elabbd et al., Graph Neural Networks for Charged Particle Tracking on FPGAs, [2112.02043](https://arxiv.org/abs/2112.02043). 2


[121] T. Kim and A. Martin, *A W± polarization analyzer from Deep Neural Networks*, 2102.05124


[124] B. Tannenwald, C. Neu, A. Li, G. Buehlmann, A. Cuddeback, L. Hatfield et al.,

[126] C. Jang, S.-K. Ko, Y.-K. Noh, J. Choi, J. Lim and T. J. Kim, Learning to increase matching efficiency in identifying additional b-jets in the \(t\bar{t}b\bar{b}\) process, \[2103.09129\]

[127] C. K. Khosa and S. Marzani, Higgs tagging with the Lund jet plane, \[2105.03989\]


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


[134] F. Dreyer, G. Soyez and A. Takacs, Quarks and gluons in the Lund plane, \[2112.09140\]


[137] S. H. Lim and M. M. Nojiri, Morphology for Jet Classification, \[2010.13469\]

[138] J. A. Aguilar-Saavedra, Pulling the Higgs and Top needles from the jet stack with Feature Extended Supervised Tagging, \[2102.01667\]


[141] J. Erdmann, O. Nackenhorst and S. V. Zeißner, Maximum performance of strange-jet tagging at hadron colliders, \[2011.10736\]
[142] CMS collaboration, A. M. Sirunyan et al., Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV, 1712.07158


[149] D. Cogollo, F. Freitas, C. 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, 2008.03409


[156] A. Stakia et al., Advanced Multi-Variate Analysis Methods for New Physics Searches at the Large Hadron Collider, 2105.07530

[157] E. Arganda, A. D. Medina, A. D. Perez and A. Szynkman, Towards a method to anticipate dark matter signals with deep learning at the LHC, 2105.12018


[189] MICROBOoNE collaboration, P. Abratenko et al., A Convolutional Neural Network for Multiple Particle Identification in the MicroBooNE Liquid Argon Time Projection Chamber, [2010.08653].


[196] ARGoNeUT collaboration, R. Acciarri et al., A deep-learning based raw waveform region-of-interest finder for the liquid argon time projection chamber, [2103.06391].

[197] D. Maksimović, M. Nieslony and M. Wurm, CNNs for enhanced background discrimination in DSNB searches in large-scale water-Gd detectors, [2104.13426].


Networks for Shower Energy Prediction in Liquid Argon Time Projection Chambers, [2110.10766]

[201] MicroBooNE collaboration, P. Abratenko et al., Electromagnetic Shower Reconstruction and Energy Validation with Michel Electrons and \( \pi^0 \) Samples for the Deep-Learning-Based Analyses in MicroBooNE, [2110.11874]

[202] MicroBooNE collaboration, P. Abratenko et al., Wire-Cell 3D Pattern Recognition Techniques for Neutrino Event Reconstruction in Large LArTPCs: Algorithm Description and Quantitative Evaluation with MicroBooNE Simulation, [2110.13961]


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


[207] A. Golovatiuk, A. Ustyuzhanin, A. Alexandrov and G. De Lellis, Deep Learning for direct Dark Matter search with nuclear emulsions, [2106.11995]

[208] J. I. McDonald, Scanning the landscape of axion dark matter detectors: applying gradient descent to experimental design, [2108.13894]


[215] Pierre Auger collaboration, A. Aab et al., Deep-Learning based Reconstruction of the Shower Maximum X_max using the Water-Cherenkov Detectors of the Pierre Auger Observatory, 2101.02946

[216] DarkMachines High Dimensional Sampling Group collaboration, C. Balázs et al., A comparison of optimisation algorithms for high-dimensional particle and astrophysics applications, 2101.04525


[222] R. Arjona and S. Nesseris, Novel null tests for the spatial curvature and homogeneity of the Universe and their machine learning reconstructions, 2103.06789


[225] T. Ikeda, T. Tanimori, A. Takada, Y. Mizumura, K. Yoshikawa, M. Abe et al., Development of Convolutional Neural Networks for an Electron-Tracking Compton Camera, 2105.02512

[226] A. Aizpuru, R. Arjona and S. Nesseris, Machine Learning improved fits of the sound horizon at the baryon drag epoch, 2106.00428


in developing a hybrid deep learning algorithm for identifying and locating primary vertices, [2103.04962]


[256] Y.-L. Du, D. Pablos and K. Tywoniuk, Jet tomography in heavy ion collisions with deep learning, 2106.11271


[258] Y.-G. Huang, L.-G. Pang, X. Luo and X.-N. Wang, Probing criticality with deep learning in relativistic heavy-ion collisions, 2107.11828


[260] J. He, W.-B. He, Y.-G. Ma and S. Zhang, Machine-learning-based identification for initial clustering structure in relativistic heavy-ion collisions, 2109.06277


[267] Y. S. Lai, J. Mulligan, M. Płoskoń and F. Ringer, The information content of jet quenching and machine learning assisted observable design, 2111.14559


[270] P. Xiang, Y.-S. Zhao and X.-G. Huang, Determination of impact parameter in high-energy heavy-ion collisions via deep learning, 2112.03824

[271] S. Soma, L. Wang, S. Shi, H. Stöcker and K. Zhou, Neural network reconstruction of the dense matter equation of state from neutron star observables, 2201.01756


[278] M. Borisyak and N. Kazeev, Machine Learning on data with sPlot background subtraction, 1905.11719


[283] O. Amram and C. M. Suarez, Tag N’ Train: A Technique to Train Improved Classifiers on Unlabeled Data, 2002.12376


Energy Mover’s Distance in the Search for Rare Phenomena at Colliders,

C. Pollard and P. Windischhofer, Transport away your problems: Calibrating stochastic simulations with optimal transport.

T. Cai, J. Cheng, K. Craig and N. Craig, Which Metric on the Space of Collider Events?

G. C. Strong, On the impact of modern deep-learning techniques to the performance and time-requirements of classification models in experimental high-energy physics.

V. V. Gligorov and M. Williams, Efficient, reliable and fast high-level triggering using a bonsai boosted decision tree.


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


S. Amrouche et al., The Tracking Machine Learning challenge : Throughput phase.


J. Ngadiuba et al., Compressing deep neural networks on FPGAs to binary and ternary precision with HLS4ML.

S. Summers et al., Fast inference of Boosted Decision Trees in FPGAs for particle physics.

J. Krupa et al., GPU coprocessors as a service for deep learning inference in high energy physics.


[340] E. Govorkova et al., *Autoencoders on FPGAs for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider*, 2108.03986.


[345] S. Cheong, A. Cukierman, B. Nachman, M. Safdari and A. Schwartzman,


[360] D. F. Rentería-Estrada, R. J. Hernández-Pinto, G. F. R. Sborlini and P. Zurita, 
Reconstructing partonic kinematics at colliders with Machine Learning, 2112.05043

project: SUSY-AI–generalizing LHC limits on supersymmetry with machine learning, 
The European Physical Journal C 77 (2017) 257

Accelerating the BSM interpretation of LHC data with machine learning, 1611.02704

[363] B. Kronheim, M. Kuchera, H. Prosper and A. Karbo, Bayesian Neural Networks for 
Fast SUSY Predictions, 2007.04506

multiplicity scattering amplitudes, JHEP 06 (2020) 114, 2002.07516

Generation, 1912.11055

Xsec: the cross-section evaluation code, 2006.16273

[367] F. Bury and C. Delaere, Matrix Element Regression with Deep Neural Networks – 
breaking the CPU barrier, 2008.10949

[368] D. L. B. Sombillo, Y. Ikeda, T. Sato and A. Hosaka, Unveiling the pole structure of 
S-matrix using deep learning, 2104.14182

[369] D. L. B. Sombillo, Y. Ikeda, T. Sato and A. Hosaka, Model independent analysis of 
coupled-channel scattering: a deep learning approach, 2105.04898

production at hadron colliders using amplitude neural networks, 2106.09474

[371] D. Maitre and H. Truong, A factorisation-aware Matrix element emulator, 
2107.06625

generation – rejection sampling using neural network event-weight estimates, 
2109.11964

[373] R. Winterhalder, V. Magerya, E. Villa, S. P. Jones, M. Kerner, A. Butter et al., 
Targeting Multi-Loop Integrals with Neural Networks, 2112.09145

[374] Y.-K. Lei, C. Liu and Z. Chen, Numerical analysis of neutrino physics within a high 
scale supersymmetry model via machine learning, 2006.01495

EFT sensitivity, 2007.10356


1909.10990.


[398] W. Haddadin, Invariant polynomials and machine learning, 2104.12733

[399] I. Chahrouf and J. D. Wells, Function Approximation for High-Energy Physics: Comparing Machine Learning and Interpolation Methods, 2111.14788

[400] L. Wang, S. Shi and K. Zhou, Reconstructing spectral functions via automatic differentiation, 2111.14760


[404] 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 1603.00027


selection efficiencies from multivariate classifiers, \textit{JINST} 8 (2013) P12013 [1305.7248].


[413] S. Wunsch, S. Jörger, R. Wolf and G. Quast, Reducing the dependence of the neural network function to systematic uncertainties in the input space, 1907.11674.


  \[2109.10919\]


[527] G. M. Alessandro Casa, Nonparametric semisupervised classification for signal detection in high energy physics. [1809.02977]


[540] 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].


[556] J. A. Aguilar-Saavedra, *Anomaly detection from mass unspecific jet tagging*, 2111.02647


[604] MicroBooNE collaboration, P. Abratenko et al., *Search for an anomalous excess of
charged-current quasi-elastic $\nu_e$ interactions with the MicroBooNE experiment using Deep-Learning-based reconstruction. 2110.14080

[605] 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.
