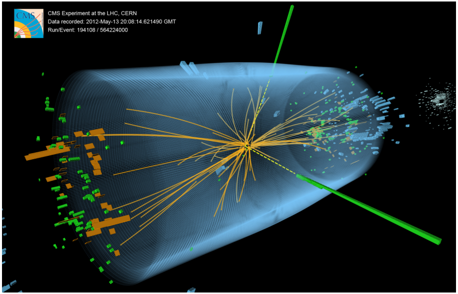
Analysis techniques in High Energy Physics

Florian Eble

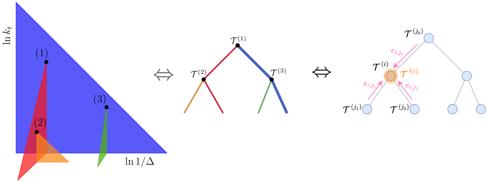
22/03/2023

IPA ML Workshop

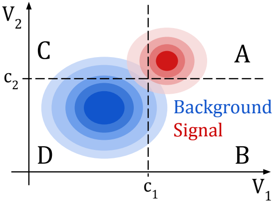
- Very wide topic!
- Will focus on some common HEP problems
- Will discuss how ML can help us solve them
- Hopefully, the problems discussed in this talk are general enough and applicable to other disciplines!



Incorporating physics inductive bias in ML models

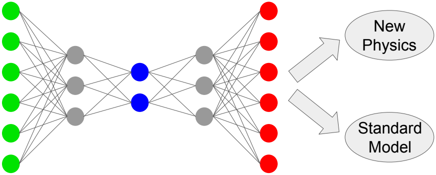


Background estimation



Training decorrelated models

Searching for new physics via anomaly detection



1. Introducing inductive bias

2. Background estimation in HEP

3. Training decorrelated models

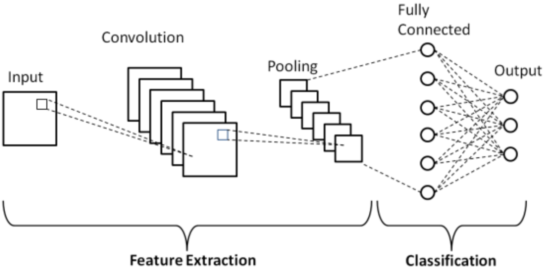
4. Searching for new physics

Inductive bias

- The **universal approximation theorem** states every continuous function can be approximated by a neural network (NN)
- However, designing architectures exploiting specificities of a problem is often a necessity for a successful learning!

  ➔ Introducing **inductive bias** in NN

- *E.g.* Convolutional Neural Network (CNN) architectures make use of translational invariance of images by implementing dedicated convolutional layer



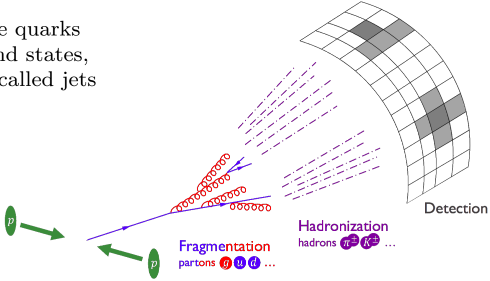
Same output!

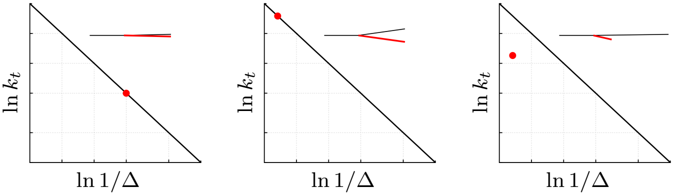
Lund plane representation of a jet

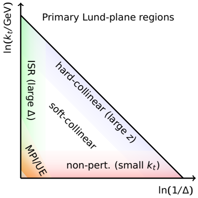
- Because of color confinement, final state quarks and gluons fragment until forming bound states, forming a collimated spray of particles called jets

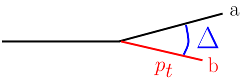
- The Lund plane[1] is 2-dimensional representation of gluon/quark emission in hadronic showers

- Natural description of the radiation pattern inside of a jet





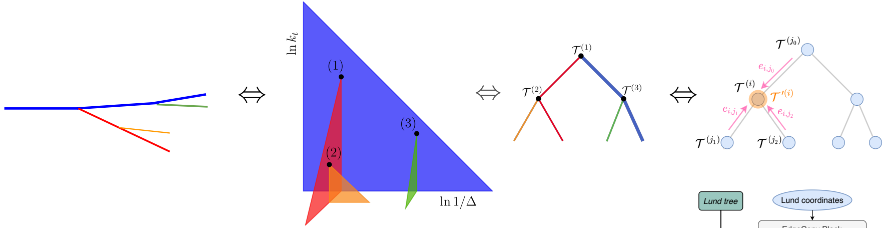




$$\Delta = \sqrt{(y_a - y_b)^2 + (\phi_a - \phi_b)^2}$$ opening angle of the splitting

$$k_t = p_t \Delta$$

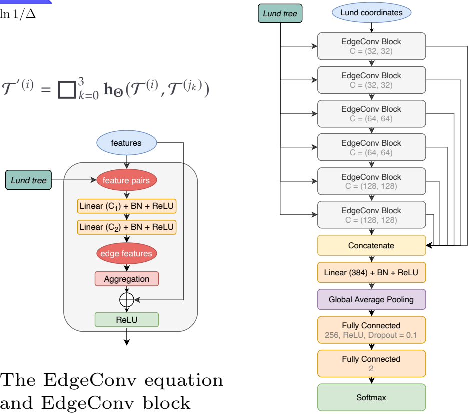
[1] See arXiv:1807.04758



- At each splitting, compute a tuple of kinematic features $\mathcal{T} = (k_t, \Delta, ...)$, used as node features in the graph
- EdgeConv operation as a fully connected NN using features of connected nodes
- Stack EdgeConv layers to build up the LundNet[1]

→ State-of-the-art jet tagger with small computational cost (fixed graph via Lund decomposition)

$$\mathcal{T}'^{(i)} = \square_{k=0}^{3} \mathbf{h}_\Theta(\mathcal{T}^{(i)}, \mathcal{T}^{(j_k)})$$



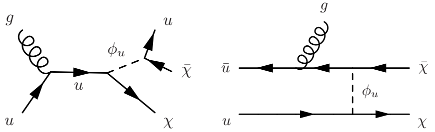
The EdgeConv equation and EdgeConv block

[1] See arXiv:2012.08526

1. Introducing inductive bias
2. Background estimation in HEP
3. Training decorrelated models
4. Searching for new physics

- General problem in HEP: estimate the expected number of background events in signal-enriched region of the phase-space
- *E.g.* searches for WIMPs by looking for an excess of event at high $\not{E}_\mathrm{T}$ (Missing Transverse Energy)
  - ➔ Need to know expected number of Standard Model events in the signal region!
- Example of arXiv:1712.02345, backgrounds estimated from simulation:

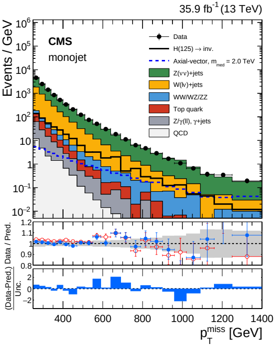
Example diagrams:



Monojet event selection:

Variable	Selection	Target background
Muon (electron) veto	$p_\mathrm{T} > 10\,\mathrm{GeV}$, $\lvert\eta\rvert < 2.4(2.5)$	$\mathrm{Z}(\ell\ell)$+jets, $\mathrm{W}(\ell\nu)$+jets
τ lepton veto	$p_\mathrm{T} > 18\,\mathrm{GeV}$, $\lvert\eta\rvert < 2.3$	$\mathrm{Z}(\ell\ell)$+jets, $\mathrm{W}(\ell\nu)$+jets
Photon veto	$p_\mathrm{T} > 15\,\mathrm{GeV}$, $\lvert\eta\rvert < 2.5$	γ+jets
Bottom jet veto	CSVv2 < 0.8484, $p_\mathrm{T} > 15\,\mathrm{GeV}$, $\lvert\eta\rvert < 2.4$	Top quark
$p_\mathrm{T}^\mathrm{miss}$	$>$250 GeV	QCD, top quark, $\mathrm{Z}(\ell\ell)$+jets
$\Delta\phi(\vec{p}_\mathrm{T}^\mathrm{\,jet}, \vec{p}_\mathrm{T}^\mathrm{\,miss})$	$>$0.5 radians	QCD
Leading AK4 jet p_T and η	$>$100 GeV and $\lvert\eta\rvert < 2.4$	All

Searching for high $\not{E}_\mathrm{T}$:

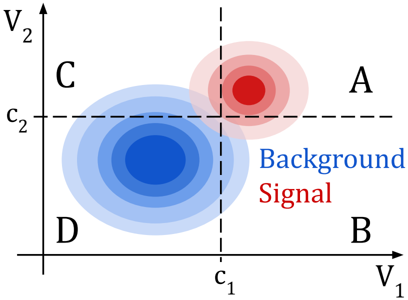


ABCD method

- Sometimes backgrounds cannot be estimated from simulation, *e.g.* cross-section of processes with large number of hadronic jets is difficult to calculate
- Need to use signal-free regions (control regions) to predict background in the signal region
- The ABCD method is one of the methods for this task

- Let V_1 and V_2 be 2 **independent** variables for the background distribution
- Signal region A: $V_1 > c_1$ and $V_2 > c_2$
- Number of events in each region: N_A, N_B, N_C, N_D
- Background estimation in signal region:

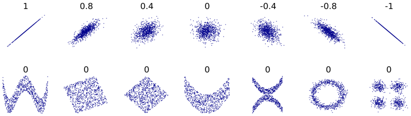
$$N_A^{\text{bkg}} = \frac{N_B}{N_D} N_C$$



- One requirement for the ABCD method is to have **2 independent variables**
- This can be checked using Distance Correlation (DisCo) [1] [2] [3]

- **Pearson correlation** only evaluates **linear correlations**:

$$\rho^2_{\text{Pearson}}(X,Y) = \frac{\mathrm{Cov}^2(X,Y)}{\mathrm{Cov}(X,X)\mathrm{Cov}(Y,Y)}$$

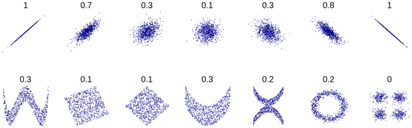


Pearson correlation coefficient

- **Distance correlation (DisCo)** defined using the probability distributions of X and Y, and their joint probability distribution

➔ Makes use of all information of the random variables!

$$\mathrm{DisCo}^2(X,Y) = \frac{\mathrm{dCov}^2(X,Y)}{\mathrm{dCov}(X,X)\mathrm{dCov}(Y,Y)}$$

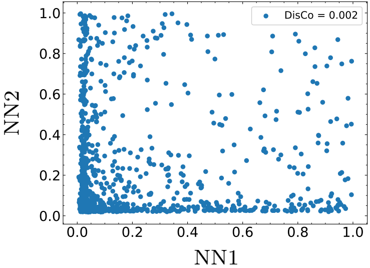


Distance correlation coefficient

Distance correlation for ABCD method

If no two clear discriminative and independent variables can be found:

- Can train one NN and use DisCo regularization to force its output to be independent of a discriminative physics observable V
  For a batch of examples X:
  $$L(X) = L_{\mathrm{NN}}(X) + \lambda \cdot \mathrm{DisCo}(\mathrm{NN(X)},\ \mathrm{V(X)}) \tag{1}$$
- Can train two NNs and use DisCo regularization to force them to be independent of each other!
  For a batch of examples X:
  $$L(X) = L_{\mathrm{NN1}}(X) + L_{\mathrm{NN2}}(X) + \lambda \cdot \mathrm{DisCo}(\mathrm{NN1(X)},\ \mathrm{NN2(X)}) \tag{2}$$
  Where L_{NN}, L_{NN1}, L_{NN2} are the NNs usual loss, *e.g.* for a supervised binary classifier, binary cross-entropy.



- And then perform ABCD background estimation in the (NN, V) plane or $(\mathrm{NN1}, \mathrm{NN2})$ plane

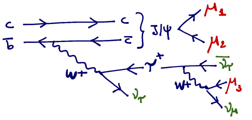
→ **ML provides a systematic way of addressing this issue!**

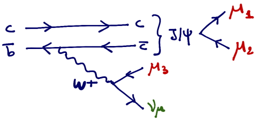
References:
arXiv:2001.05310
arXiv:2007.14400

- Search for Lepton Flavour Violation by measuring $R(J/\psi)$ branching ratio
- Define signal region requiring muon identification ("ID") and its isolation ("ISO")

$$R(J/\psi) = \frac{\mathcal{B}\left(B_c^+ \to J/\psi\tau^+ \left(\to \mu^+\bar{\nu}_\mu\nu_\tau\right)\nu_\tau\right)}{\mathcal{B}\left(B_c^+ \to J/\psi\mu^+\nu_\mu\right)}$$

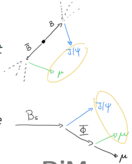
- **Main challenge:**
  estimate J/ψ + misidentified muon background!
  Very hard to simulate!





$J/\psi\mu$ background: *dominant*

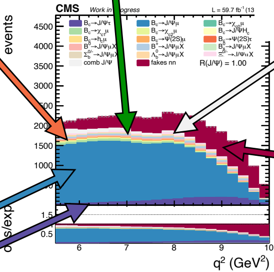
MC based; normalisation data-driven



B_c background:

MC based; normalisation data-driven

- Feeddowns: excited $c\bar{c}$ states to J/ψ
- Other J/ψ+charmed hadron, mostly $B_c \to D_s^{(*)}J/\psi$



DiMuon

Data-driven

Pairs of unrelated muons with m(μμ) close to that of the J/Ψ

Muon fakes

Data-driven anti-isolated μ sideband

J/Ψ + misidentified hadron

Signals:

Signal μ: $B_c \to J/\psi\mu\nu_\mu$

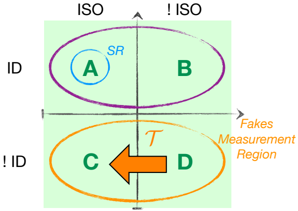
Signal τ: $B_c \to J/\psi\tau\nu_\tau$

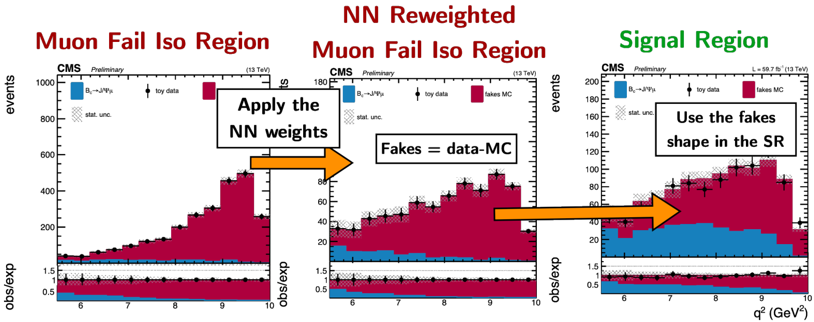
- Learn transfer factor $\mathcal{T}$ from signal-free control region D to C for N physics variables $\mathcal{V}_i$

$$\mathcal{T} = \frac{\alpha - \beta \cdot \gamma}{1 - \gamma}, \; \alpha = \frac{\mathrm{Data}_C}{\mathrm{Data}_D}, \; \beta = \frac{\mathrm{MC}_C}{\mathrm{MC}_D}, \; \gamma = \frac{\mathrm{MC}_D}{\mathrm{Data}_D}$$

- Train 3 NNs to learn the mappings α, β, γ between the N-dimensional distributions of the different "regions"
- Apply $\mathcal{T}$ from signal-free region B to signal region A:

$$\mathrm{Fake}_A = \mathcal{T}(\mathrm{Data}_B) \cdot \mathrm{Data}_B - \mathcal{T}(\mathrm{MC}_B) \cdot \mathrm{MC}_B$$





- **Distance correlation regularization** can also be used to train a classifier decorrelated from a feature $\mathcal{F}$
- Particularly interesting when this feature $\mathcal{F}$ is then used in the next step of the analysis, for instance fitted to extract signal

➔ Example: Mass decorrelated classifier for resonant search (bump hunt in a mass variable)

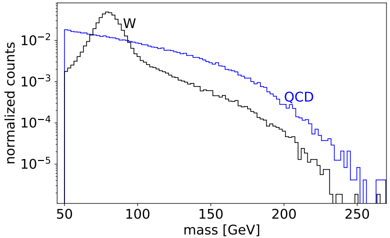


FIG. 1: Invariant mass distribution for the inclusive W and QCD samples.

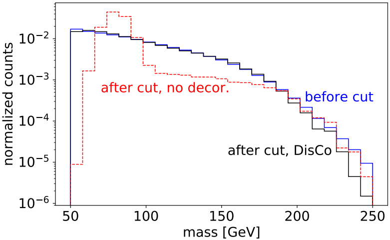


FIG. 4: QCD mass distribution before and after a cut on CNN plus DisCo (W-tagging) with signal efficiency of 50% and JSD $\sim 10^{-3}$.

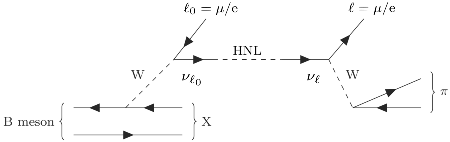
[arXiv:2001.05310]

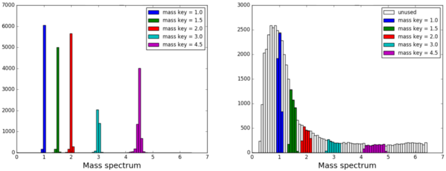
- What if the signal is unknown? *E.g.* searching for resonance at unknown mass
- Can use **parametric neural network** (pNN)
- Train a NN on a mixture of signals with an additional input feature: the true value of the signal parameter p
- For background examples:
  - If p not meaningful for background, use random value, following signal distribution
  - Else, *e.g.* p directly translates as a physics observable A, use $p = A$

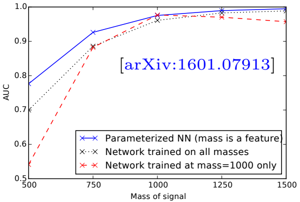
➔ pNNs achieve same performance on hypothesis p_0 as a single NN trained only on signal hypothesis p_0

➔ pNNs better interpolate between the different signal hypotheses than a plain NN trained on a mixture of signals

pNN used at ETH in search for long-lived Heavy Neutral Leptons:







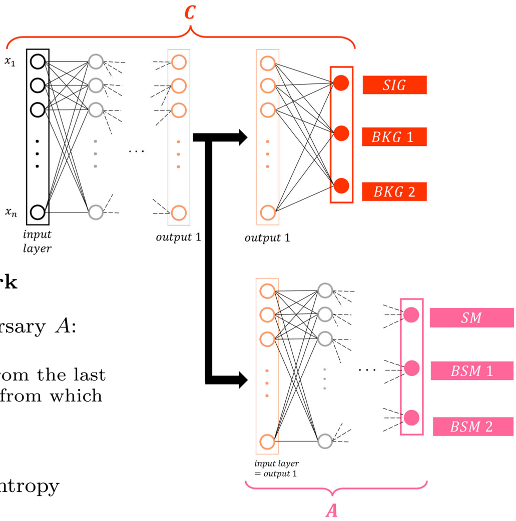
[arXiv:1601.07913]

- What if the model parameters to decorrelate against are not directly related to a physics observable?
- *E.g.* varying anomalous coupling of the Higgs boson
  - ➔ impacts many observables in a non-trivial way and signal shape for template fit!
- Can use **adversarial neural network**
- Train both a classifier C and an adversary A:
  - C classifies signal vs background
  - A takes the latent representation from the last hidden layer of C and tries to find from which process it comes:

$$L = L_C - \alpha L_A$$

where $L_{C/A}$ is the categorical cross-entropy
α is an hyper-parameter

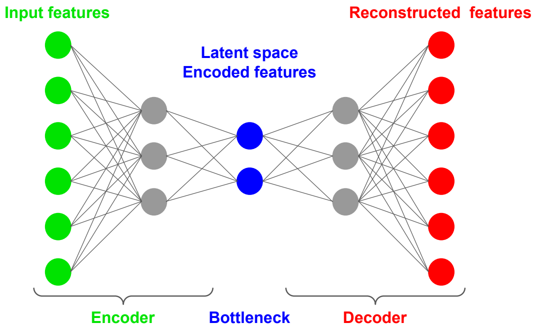
- If the adversary A cannot figure out from which process the event is, then the output of C is model-independent!



[Eur. Phys. J. C 82, 921 (2022)]

An autoencoder (AE) is composed of:

- an encoder NN f
- a "symmetric" decoder NN g



The AE network is trained to learn to reconstruct the input examples it is given.

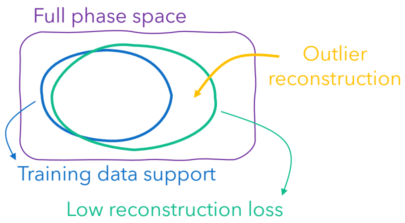
Loss for an example x:

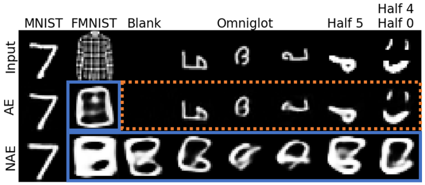
$$L(x) = ||g(f(x)) - x||$$

where $||\cdot||$ is a distance

The aim of an AE for anomaly detection is to reconstruct with low error only the examples it is trained on but not others!

Search for new physics in HEP with AE is based on learning SM physics and flag new physics as anomalous!





Outlier reconstruction example: AE and NAE trained on MNIST, other inputs are outliers.

- Outlier reconstruction happens when the network assigns low reconstruction error to out-of-distribution (OOD) examples
- OOD reconstruction not suppressed during training in plain AE
- Sometimes phrased as "OOD examples need to be more 'complex' to not be reconstructed"

→ **Normalized autoencoder[1] (NAE) features a mechanism to suppress OOD reconstruction!**

[1] NAE first introduced in arXiv:2105.05735 and used in HEP in arXiv:2206.14225

Working principle of the Normalized Autoencoder (NAE)

- **Ensure that low reconstruction error phase-space matches that of training data**
- *i.e.* OOD examples are constrained to have high reconstruction error
- The model probability p_θ is defined from the reconstruction error E_θ via the Boltzmann distribution:

$$p_\theta(x) = \frac{1}{\Omega_\theta} \exp\left(-E_\theta(x)\right)$$

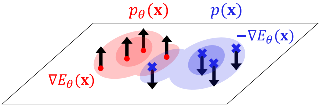


Figure 2. An illustration of the energy gradients in Eq. (7). The red and blue shades represent the model and the data density, respectively. The gradient update following Eq. (7) increases the energy of samples from $p_\theta(\mathbf{x})$ (the red dots) and decreases the energy of training data (the blue crosses).

➔ Low energy examples have high probability

- The loss is designed to learn $p_\theta = p_{\text{data}}$:

$$\mathbb{E}_{x\sim p_{\text{data}}}\left[L_\theta(x)\right] = \underbrace{\mathbb{E}_{x\sim p_{\text{data}}}\left[E_\theta(x)\right]}_{\text{positive energy } E_+} - \underbrace{\mathbb{E}_{x'\sim p_\theta}\left[E_\theta(x')\right]}_{\text{negative energy } E_-}$$

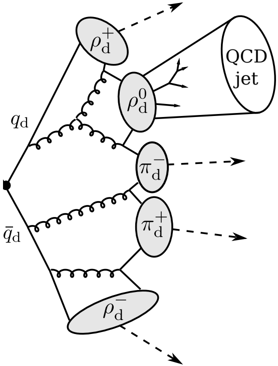
- Positive energy is the reconstruction error of the training examples
- Need to sample from the model to get the “negative samples” x' and compute E_-
  ➔ Monte Carlo Markov Chain (MCMC) employed

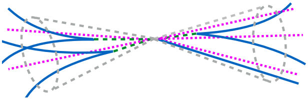
- Semi-visible jets (SVJ) are new physics signatures arising from theories where dark matter is made of dark quarks and a dark QCD force, very similar its SM counterpart
- Dark quarks hadronize to form dark hadrons, a fraction of which promptly decays to SM quarks which hadronize in the SM sector
- SVJs are jets made of visible SM hadrons with different substructure than SM QCD jets
- Currently developing NAE using substructure variables and a fully connected NN
- Loss function:

$$L = \log\left(\cosh\left(E_+ - E_-\right)\right) + \lambda_+ E_+^2$$

➔ First term to suppress OOD reconstruction

➔ Second term to learn training examples reconstruction





SM hadrons
Stable dark hadrons

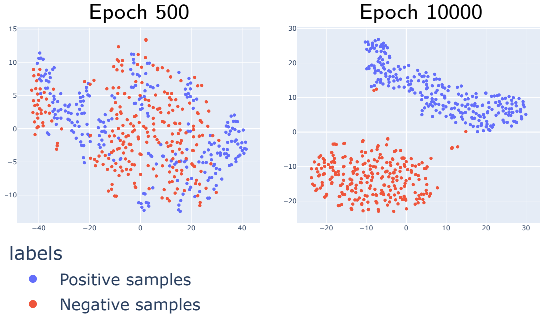
Understanding NAE dynamics

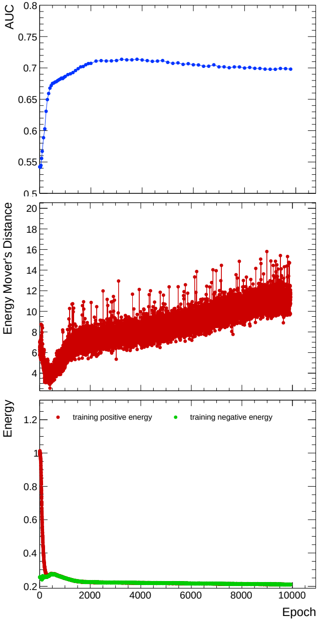
Visualizing positive and negative samples:

- Energy Mover's Distance (EMD)
- t-distributed Stochastic Neighbor Embedding (t-SNE) plots

➔ Check suppression of OOD reco, e.g. "that the reco loss is high outside the training manifold"

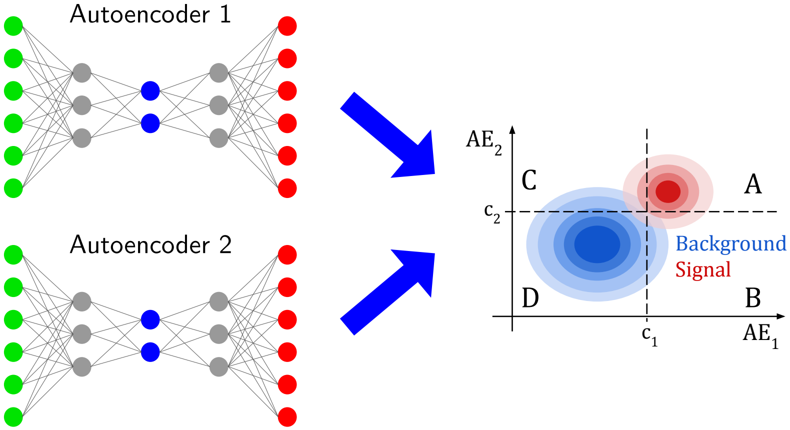
➔ Good anomaly detection: low reco error of training examples (SM physics) AND suppression of OOD (BSM physics) reco (low EMD, overlap in t-SNE plots)





The idea is[1]:

- to train 2 autoencoders, decorrelated from each other using DisCo regularization
- such that the new physics enriched region is the high loss region of the AEs
- to perform ABCD background estimation using the losses of the two AEs



[1] arXiv:2111.06417

- ML provides tools to address several HEP problems:
  - Background estimation
  - Building decorrelated classifiers with respect to signal hypotheses or a physics observable
- ... not mentioning building classifiers for jet tagging, searches or precision measurement!
- Many exiting developments to search for new physics with unsupervised learning!
- Still ongoing developments to incorporate physics knowledge into new ML models and improve interpretability

Backup

Distance correlation (DisCo)

The **Pearson correlation** only evaluates **linear correlations**:

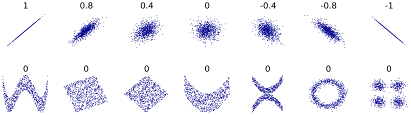
$$\rho^2_{\text{Pearson}}(X,Y) = \frac{\text{Cov}^2(X,Y)}{\text{Cov}(X,X)\text{Cov}(Y,Y)} \tag{3}$$

The **Distance correlation (DisCo)** makes use of all information of the random variables:

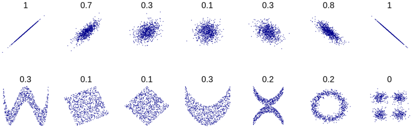
$$\text{dCov}^2(X,Y) = \int d^p s d^q t \left| f_{X,Y}(s,t) - f_X(s) f_Y(t) \right|^2 w(s,t)$$

where f_X (resp. Y) is the characteristic function of X (resp. Y), $f_{X,Y}$ is the joint characteristic function of X and Y.
$f_{X,Y} == f_X f_Y$ iff X and Y are **independent**.

$$\text{DisCo}^2(X,Y) = \frac{\text{dCov}^2(X,Y)}{\text{dCov}(X,X)\text{dCov}(Y,Y)} \tag{4}$$



Pearson correlation coefficient



Distance correlation coefficient

Energy-based models (EMBs)

- EBMs are models where the probability is defined through the Boltzmann distribution
- Let θ denote the model parameters
- The model probability p_θ is defined from the energy E_θ

$$p_\theta(x) = \frac{1}{\Omega_\theta} \exp\left(-E_\theta(x)/T\right) \tag{5}$$

where the normalization constant Ω_θ is

$$\Omega_\theta = \int \exp\left(-E_\theta(x)/T\right) dx \tag{6}$$

- The EBM loss for a training example x is the negative log-likelihood:

$$L_\theta(x) = -\log p_\theta(x) = E_\theta(x)/T + \log \Omega_\theta \tag{7}$$

- The gradient of the EBM loss is thus:

$$\nabla_\theta L_\theta(x) = \nabla_\theta E_\theta(x) - \mathbb{E}_{x' \sim p_\theta}\left[\nabla_\theta E_\theta(x')\right] \tag{8}$$

- The expectation value over the training dataset, with probability p_{data} is:

$$\mathbb{E}_{x \sim p_{\text{data}}}\left[\nabla_\theta L_\theta(x)\right] = \mathbb{E}_{x \sim p_{\text{data}}}\left[\nabla_\theta E_\theta(x)\right] - \mathbb{E}_{x' \sim p_\theta}\left[\nabla_\theta E_\theta(x')\right] \tag{9}$$

Working principle of the Normalized Autoencoder (NAE)

Loss

$$\mathbb{E}_{x\sim p_{\mathrm{data}}}\left[L_\theta(x)\right] = \underbrace{\mathbb{E}_{x\sim p_{\mathrm{data}}}\left[E_\theta(x)\right]}_{\text{positive energy}} - \underbrace{\mathbb{E}_{x'\sim p_\theta}\left[E_\theta(x')\right]}_{\text{negative energy}} = E_+ - E_-$$

Positive energy

- Simply the reconstruction error over the training dataset
- Take SM jets and compute the reconstruction error!

Negative energy

- Reconstruction error of the "negative samples" x' from the probability distribution p_θ
- Need to sample from the model to get the "negative samples"

➔ Monte Carlo Markov Chain (MCMC) employed

MCMC

- Start from an initial point x'_0
- Run n Langevin MCMC steps:

$$x'_{i+1} = x'_i - \underbrace{\lambda_i \nabla_x E_\theta(x'_i)}_{\text{drift}} + \underbrace{\sigma_i \epsilon}_{\text{diffusion}} \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{10}$$

- Repeat with several points $x_0'^{(j)}$, the negative samples are the $x_n'^{(j)}$

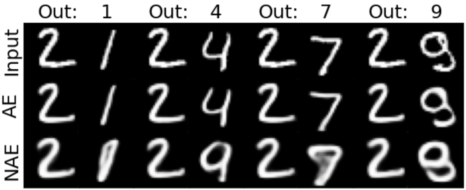
Improvements compared to plain AE

Table 1. MNIST hold-out class detection AUC scores. The values in parentheses denote the standard error of mean after 10 training runs.

HOLD-OUT:	0	1	2	3	4	5	6	7	8	9	AVG
NAE-OMI	**.989**(.002)	**.919**(.013)	**.992**(.001)	**.949**(.004)	**.949**(.005)	**.978**(.003)	**.938**(.004)	**.975**(.024)	**.929**(.004)	**.934**(.005)	**.955**
AE	.819	.131	.843	.734	.661	.755	.844	.542	.902	.537	.677

Signal	NAE	
	AUC	$\epsilon_B^{-1}(\epsilon_S = 0.2)$
top (AE)	0.875	68
top (NAE)	0.91	80
QCD (AE)	0.579	12
QCD (NAE)	0.89	350

AUC score for top tagging (2 first rows) and QCD tagging (2 last rows) for AE and NAE. The AE is a pre-training phase of the NAE.



Reconstruction examples in MNIST hold-out class detection for AE (middle row) and NAE (bottom row). Each pair of column is a different training for a different hold-out class.

- The NAE brings huge improvement compared to the plain AE on image classification task
- NAE achieves symmetric tagging, not only tagging of more complex objects!
- State-of-the-art anomaly detection on images