Unsupervised tagging of semivisible jets with normalized autoencoders in CMS

Florian Eble, on behalf of the CMS collaboration

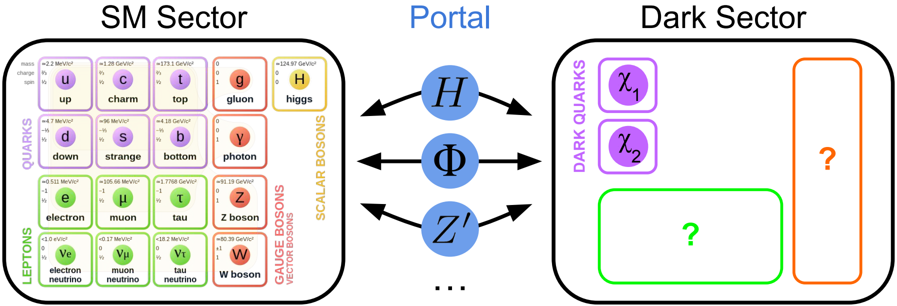
ETH zürich

09/11/2023

ML4Jets 2023, Hamburg

DM as a strongly coupled dark sector

- Hidden Valley [arXiv:hep-ph/0604261] with new particles and forces form the dark sector
- Strongly coupled dark sector
  - ➔ New confining $SU(N)$ force, dark QCD, and dark quarks
- Portal between the SM and dark sectors via a heavy mediator
  - Considering **non-resonant** production of dark quarks via t-channel mediator


SM Sector
Portal
Dark Sector
mass
charge
spin
≈2.2 MeV/c² ⅔ ½ u up
≈1.28 GeV/c² ⅔ ½ c charm
≈173.1 GeV/c² ⅔ ½ t top
0 0 1 g gluon
≈124.97 GeV/c² 0 0 H higgs
≈4.7 MeV/c² −⅓ ½ d down
≈96 MeV/c² −⅓ ½ s strange
≈4.18 GeV/c² −⅓ ½ b bottom
0 0 1 γ photon
≈0.511 MeV/c² −1 ½ e electron
≈105.66 MeV/c² −1 ½ μ muon
≈1.7768 GeV/c² −1 ½ τ tau
≈91.19 GeV/c² 0 1 Z Z boson
<1.0 eV/c² 0 ½ νe electron neutrino
<0.17 MeV/c² 0 ½ νμ muon neutrino
<18.2 MeV/c² 0 ½ ντ tau neutrino
≈80.39 GeV/c² ±1 1 W W boson
QUARKS
LEPTONS
GAUGE BOSONS
VECTOR BOSONS
SCALAR BOSONS
H
Φ
Z'
...
DARK QUARKS
χ_1
χ_2
?
?

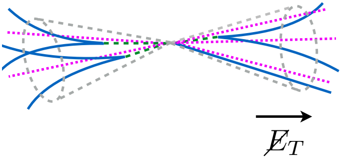
Production of semivisible jets

- Dark quarks hadronize in the dark sector
- A fraction of dark hadrons promptly decays to SM quarks which hadronize in the SM sector
- Remaining dark hadrons are stable and invisible $\Longrightarrow$ DM candidates

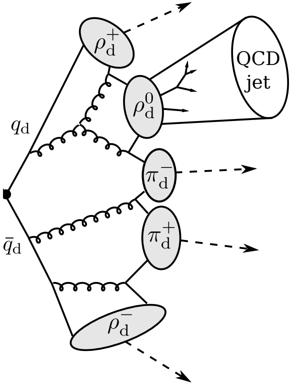
➔ Production of semivisible jets (SVJ) [arXiv:1503.00009, arXiv:1707.05326]

➔ **Different jet substructure due to double hadronization**

$$\not{E}_{\mathrm{T}} = \left\|\sum \vec{p_{\mathrm{T}}}\right\|$$



SM hadrons
Stable dark hadrons



Unsupervised ML to search for SVJs

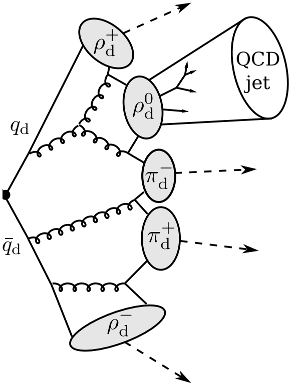
The details of the shower in the dark sector depend on many unknown parameters, e.g.:

- Number of colors and flavors in the dark sector
- Masses of the dark hadrons
- Dark QCD hadronization scale

?

➔ Simulation of SVJs very model-dependent

➔ Use unsupervised ML to tag SVJs



Autoencoders (AE)

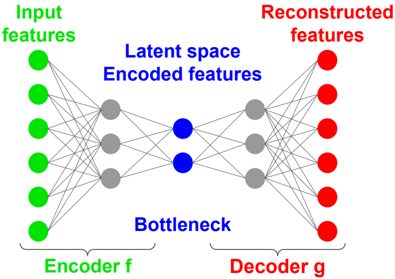
- AEs are trained to minimize the reconstruction error (e.g. MSE) between input and output:

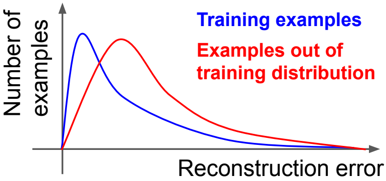
$$L(x) = ||g(f(x)) - x||$$

➔ Aim: that examples out of the training distribution, i.e. anomalies, have a higher reconstruction error

➔ Trained on SM data, AEs can perform signal-agnostic searches for new physics [arXiv:1808.08979, arXiv:1808.08992]

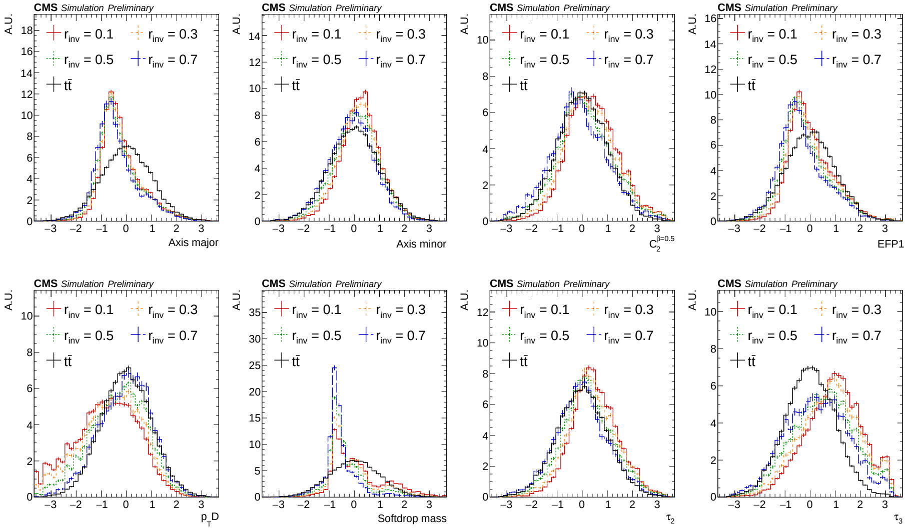
➔ Will use interchangeably:
  - "training" and "background"
  - "anomaly" and "signal"





Input features

- Input features to the AE are 8 jet substructure variables (CMS simulation)
- Normalized using quantile transformation to a normal distribution
- AE architecture: fully connected NN with 10, 10, 6, 10, 10 neurons



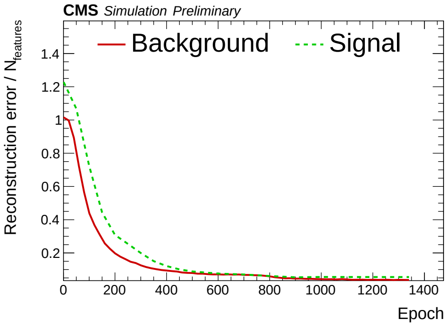
Shortcoming of standard autoencoder

- Training standard AE on background $t\bar{t}$ jets minimizing the MSE between input and reconstructed features

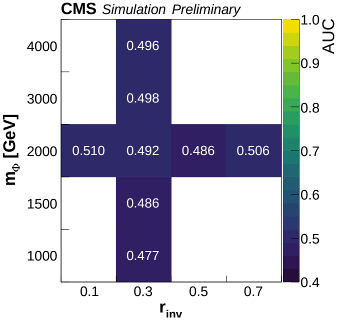
➔ **When the background MSE is minimal, the AE reconstructs background and signal jets equally well!**

➔ **The reconstruction error is not a good metric!**

➔ **Cannot optimize on AUC without introducing signal model dependence!**



Background ($t\bar{t}$) and signal (SVJ) MSE.



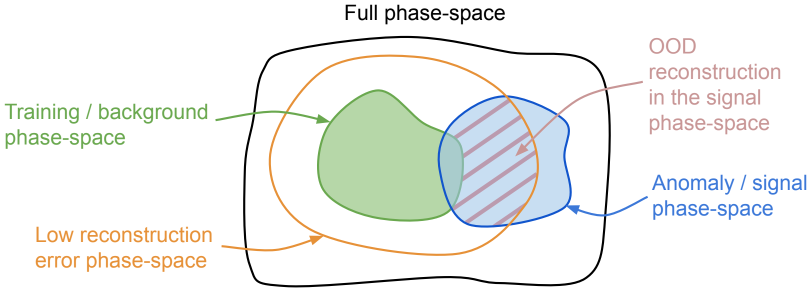
m_Φ [GeV] \ r_{inv}	0.1	0.3	0.5	0.7
4000		0.496		
3000		0.498		
2000	0.510	0.492	0.486	0.506
1500		0.486		
1000		0.477		

Average AUCs for 10 independent AE trainings, evaluated at minimal background MSE.

The problem of out-of-distribution (OOD) reconstruction

- Standard AEs are trained to minimize reco error in the background phase-space
- but **AEs are free to minimize reco error outside the background phase-space!** including the unknown signal phase-space...

➔ This is the problem of **OOD reconstruction**:

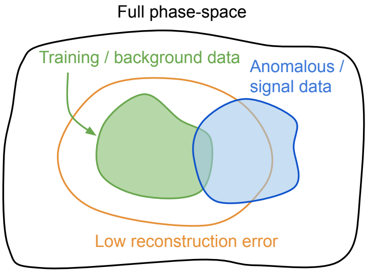

Full phase-space
OOD reconstruction in the signal phase-space
Training / background phase-space
Anomaly / signal phase-space
Low reconstruction error phase-space

- Normalized AE (NAE) features a mechanism to suppress OOD reconstruction
- 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**

➔ Need a way to sample from the low reco error phase-space, independent from the training dataset



- The low reco error distribution p_θ is constructed from the reco error E_θ via the Boltzmann distribution[1]:

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

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

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

- The positive energy E_+ is the reconstruction error of the training examples

- Markov Chain Monte Carlo (MCMC)[2] employed to sample examples from the low reco phase-space ("negative samples") x' and compute their reconstruction error E_-

[1] More on Energy Based Models in backup slide 4

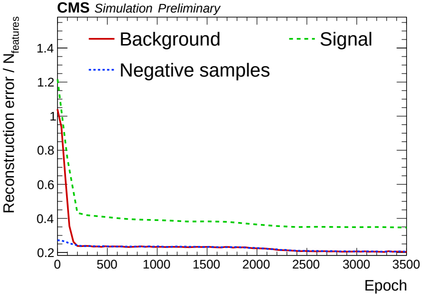
[2] More on MCMC in backup slide 5

Modified loss function and training dynamics

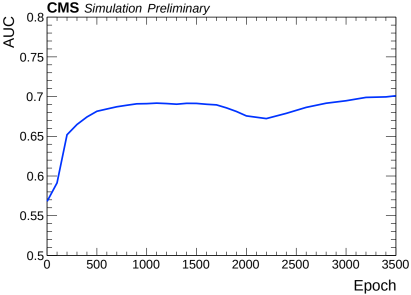
Modified default loss function, compared to arXiv:2105.05735, to:

- prevent the divergence of negative energy
- minimize the positive energy while the energy difference is close to 0:

$$L = \log\left(\cosh\left(E_+ - E_-\right)\right) + \alpha E_+ \qquad \alpha > 0,\ \text{hyper-parameter}$$



Reconstruction errors (energies) for a representative training.

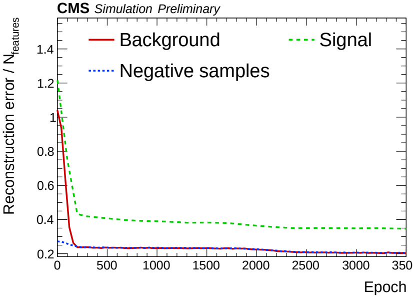


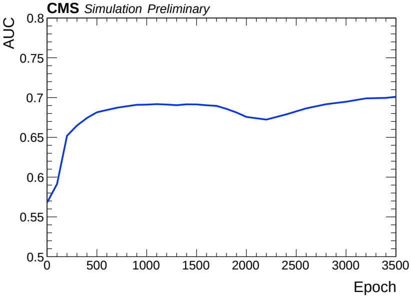
Average AUC over an ensemble of signal hypotheses for a representative training.

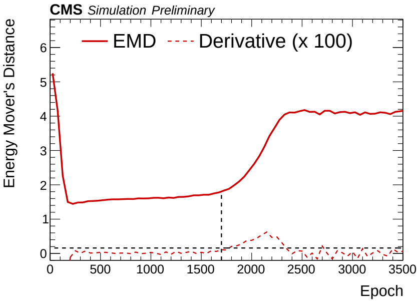
➔ **Signal SVJ reconstruction is efficiently suppressed!**

➔ How to define stopping condition in a fully signal-agnostic way?

Training stopping condition for optimal performance







- **The Energy Mover's Distance (EMD) between the positive and negative samples is a measure of the distance between the background and NAE low reco error phase-spaces directly in the input features space**

- Always observing a "collapse" of the NAE: as the MSE is further minimized, the EMD increases

➔ Best epoch before the NAE collapse

EMD versus energy difference: illustration of the collapse

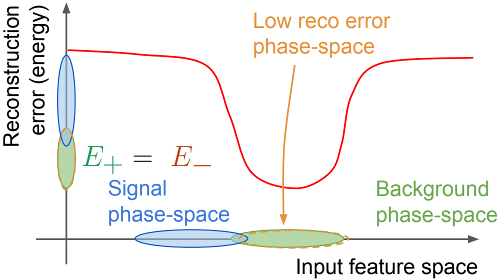


Illustration before collapse:

- Background (positive) and low error (negative) phase-spaces match

➔ **Low EMD and low energy difference** between negative and positive phase-spaces

➔ Anomalies have large reco error

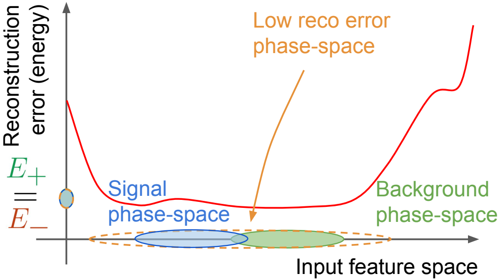


Illustration after collapse:

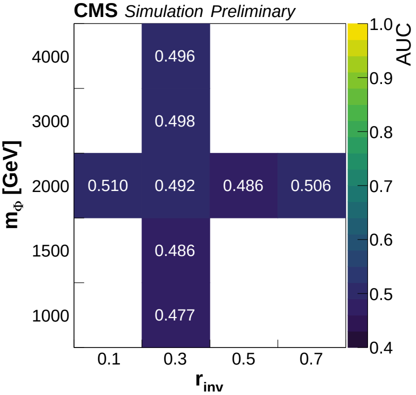
- Large discrepancy between background and low error phase-spaces

➔ **Large EMD but low energy difference** between negative and positive phase-spaces

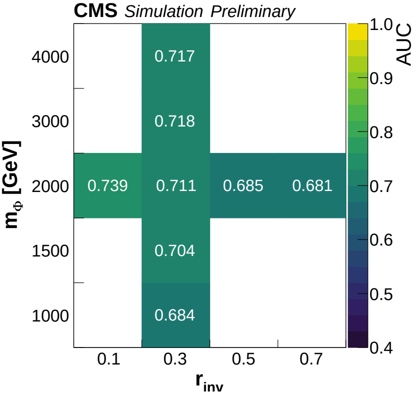
➔ Anomalies are not distinguishable from background

Results: AE versus NAE

- The NAE achieves sensible improvement in performance compared to the standard AE



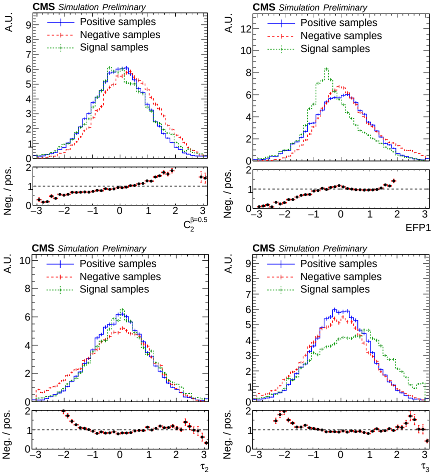
Average AUCs for 10 independent AE trainings, evaluated at minimal background MSE.



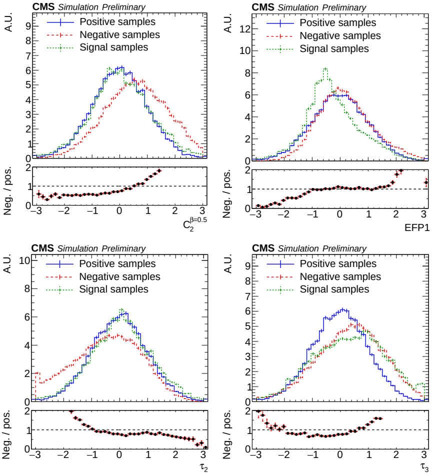
Average AUCs for 10 independent NAE trainings, evaluated before the “phase-space collapse”.

Histograms of negative and positive samples

- Can visualize negative samples for individual input features



Histograms of positive, negative and signal samples before the "phase-space collapse".



Histograms of positive, negative and signal samples after the "phase-space collapse" (epoch 2200).

Conclusions

- Standard AEs suffer from out-of-distribution reconstruction, as they are free to minimize the reconstruction error outside the training phase-space
- NAEs propose a mechanism to ensure that the low reco error phase-space matches that of the training data, by minimizing the difference in reco error between these phase-spaces
- This can fail as different phase-space regions may have same reco error
- The Energy Mover's Distance between training examples and low reco error examples in the input features space lifts this degeneracy and provides a robust distance measure between training and low reco error phase-spaces
- This work provides techniques to gain insight in the NAE dynamics and a fully model-independent optimization to reach optimal performance
- We believe the method proposed in this talk is general and not limited to the SVJ search

Backup slides

Backup

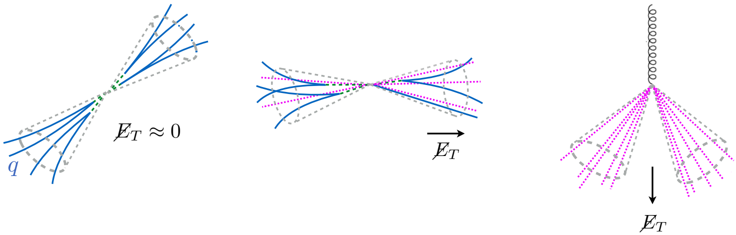
SVJ model parameters

Model parameters:

- m_Φ: Mass of the mediator
- m_D: Mass of the dark hadrons (π_D, ρ_D)
  - Same for all dark hadrons
- y_D: Yukawa coupling between SM and dark quarks
- r_inv: Jet invisible fraction
  - Effective parameter in the simulation Branching ratio DM → $q\bar{q}$

$$r_\mathrm{inv} = \left\langle \frac{\text{Number of stable dark hadrons}}{\text{Number of dark hadrons}} \right\rangle$$

$r_\mathrm{inv} = 0$ $0 < r_\mathrm{inv} < 1$ $r_\mathrm{inv} = 1$



Dijet search SVJ search WIMP search

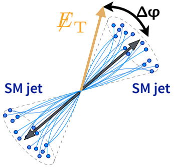
SM hadrons

Stable dark hadrons

Backgrounds

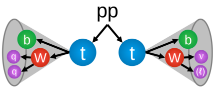
QCD multijet

- Artificial missing transverse energy $\not{E}_{\mathrm{T}}$ aligned with jet from jet energy mismeasurement
- Large cross-section



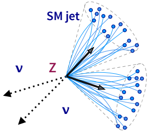
$\mathbf{t\bar{t}}$

- Large jet from boosted t
- Semi-leptonic channel $W(\rightarrow l\nu)$ with lost lepton, genuine $\not{E}_{\mathrm{T}}$ from neutrino
- Jet aligned with $\not{E}_{\mathrm{T}}$



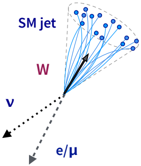
Z + jets

- Genuine $\not{E}_{\mathrm{T}}$ from $Z \rightarrow \nu\nu$



W + jets

- $W \rightarrow l\nu$ with lost/not reconstructed lepton or hadronic decay of τ
- Genuine $\not{E}_{\mathrm{T}}$ from neutrino



Energy-based models

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

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

where the normalization constant Ω_θ is

$$\Omega_\theta = \int \exp\left(-E_\theta(x)/T\right) dx \tag{2}$$

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

$$L_\theta(x) = -\log p_\theta(x) = E_\theta(x)/T + \log \Omega_\theta \tag{3}$$

- The gradient of the EBM loss is thus:

$$\nabla_\theta L_\theta(x) = \nabla_\theta E_\theta(x) - \mathbb{E}_{x' \sim p_\theta}\left[\nabla_\theta E_\theta(x')\right] \tag{4}$$

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

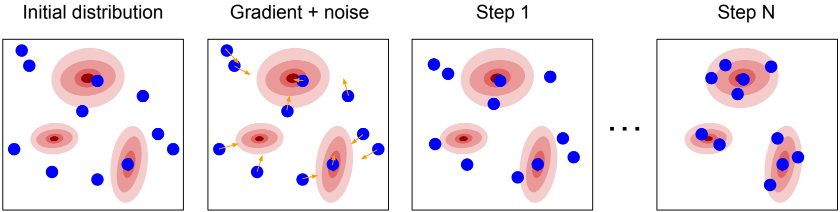
$$\mathbb{E}_{x \sim p_{\text{data}}}\left[\nabla_\theta L_\theta(x)\right] = \mathbb{E}_{x \sim p_{\text{data}}}\left[\nabla_\theta E_\theta(x)\right] - \mathbb{E}_{x' \sim p_\theta}\left[\nabla_\theta E_\theta(x')\right] \tag{5}$$

Principle of MCMC (Langevin Monte Carlo)

- Let p be a probability distribution on $\mathbb{R}^d$
- Consider x_0 a random initial set of n points in $\mathbb{R}^d$
- With the update rule:

  $$x_{t+1} = x_t + \lambda \nabla \log\left(p(x_t)\right) + \sqrt{2 \cdot \lambda} \cdot \epsilon_t$$

  where ϵ_t is a sample of n points drawn from a multivariate normal distribution on $\mathbb{R}^d$
- Let ρ_t denote the probability distribution of x_t
- In the limit $t \to \infty$, ρ_t approaches a stationary distribution ρ_∞, and $\rho_\infty = p$



MCMC in Normalized Autoencoder (NAE)

Loss

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

positive energy E_+ negative energy 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:

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

drift diffusion

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

Training samples and hyper-parameters

Input features
Using AK8 jets because SVJ are expected to be wide

Jet width	Axis major axis minor
N-pronginess	τ_2, τ_3 $C_2^{\beta=0.5}$, $D_2^{\beta=0.5}$
Other	$p_{\mathrm{T}}^{\mathrm{D}}$, EFP1 log(softdrop mass)

Architecture
Fully connected neural net

Hidden layers: $10, 10, 6, 10, 10$

Hyper-parameters

Hyper-parameter	Value
Batch size	256
Reconstruction loss	MSE
Activation	ReLU
Output encoder/ decoder activation	Linear
Optimizer	Adam
Learning rate	1e-5
Dropout	0.
MCMC	PCD
Sampling phase-space	[-3, 3] hypercube

Number of events

m_Φ [GeV]	1000	1500	2000				3000	4000	QCD	$t\bar{t}$
r_{inv}	0.3	0.3	0.1	0.5	0.3	0.7	0.3	0.3		
Number of events	23k	25k	23k	18k	16k	11k	14k	14k	83k	23k

Number of AK8 jets

Background jets	Leading 2 jets
Signal jets	Only SVJ in leading 2 jets

Train/validation/test splitting
0.7/0.15/0.15

Understanding the MCMC hyper-parameters

- Recall the MCMC equation:

$$x'_{i+1} = x'_i - \lambda \nabla_x E_\theta(x'_i) + \sigma\epsilon \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

- A theoretically motivated choice[1] for the MCMC hyper-parameters is:

$$2 \cdot \lambda = \sigma^2$$

- The MCMC is run on every batch: in practice, for training in a reasonable amount of time, the MCMC is rather short

- To speed up the convergence of the MCMC, the temperature T is introduced:

$$x'_{i+1} = x'_i - \frac{\lambda}{T} \nabla_x E_\theta(x'_i) + \sigma\epsilon \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

- Tweaking the gradient step size can be seen as adjusting the temperature T:
  the strength of the gradient term is increased for $T < 1$

- The parameter space where σ and T are set independently, with $T < 1$ and $\lambda = \sigma^2/2$ is in theory a good region

[1] For an infinitely long chain, see backup slide 5

MCMC initialization

MCMC initialization:

- In theory, MCMC convergence independent on the initial point
- However, in practice with short chain, initialization is crucial

Several commonly used initialization algorithms of the MCMC:

- Contrastive Divergence[1] (CD)
- Persistent CD[2] (PCD)

CD[3]

- Initial distribution from training data
- Re-initialization after each parameter update (*i.e.* epoch)

PCD[4]

- Random initial distribution for first MCMC
- The model changes only slightly during parameter update
- Thus, for subsequent chains, initialize chain at the state in which it ended for the previous model
- Possibility to randomly re-initialize a small fraction of the samples

[1] Neural Comput 2002; 14 (8)

[2] PCD paper

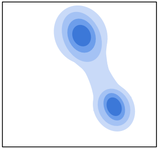
[3] Illustration in backup slide 10

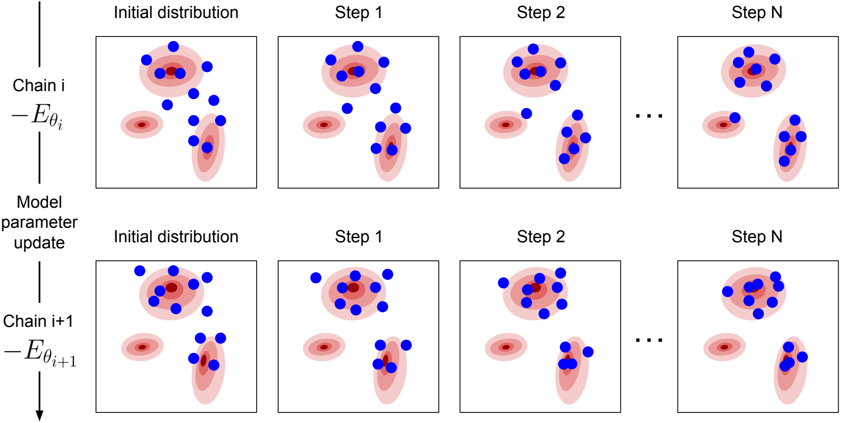
[4] Illustration in backup slide 11

Principle of CD

Example of a failure mode of CD: High probability mode far from training data distribution is not sampled

Training data distribution:





Chain i

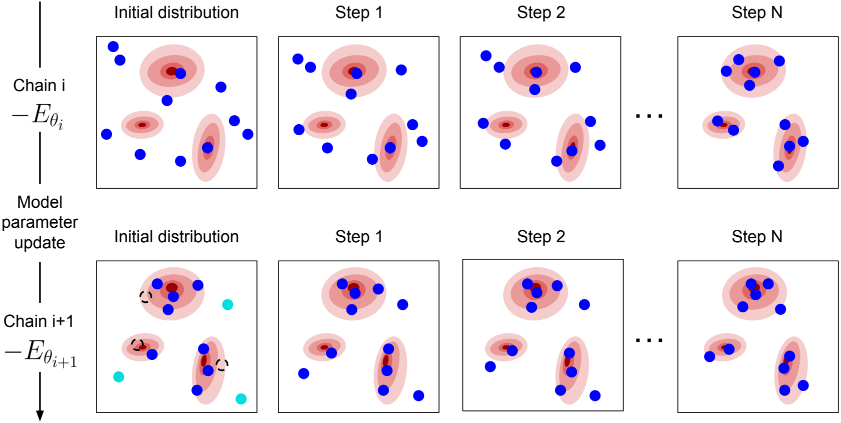
$-E_{\theta_i}$

Model parameter update

Chain i+1

$-E_{\theta_{i+1}}$

Principle of PCD



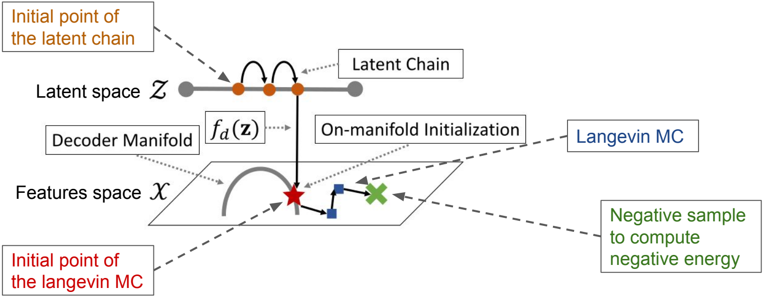
On-Manifold Initialization

Tailored MCMC initialization algorithm for AEs:

- CD and PCD have failure modes
  - CD failure mode: spurious low reconstruction error phase-space far from the training dataset
  - PCD failure mode: MCMC chains very correlated, spurious low reconstruction error phase-space can be missed

➔ Tailored algorithm for AE: On-Manifold Initialization (OMI) [arXiv:2105.05735]

- Run a first MCMC in the latent space to generate samples lying near the decoder manifold
- Use them as initial points for the usual MCMC



Initial point of the latent chain
Latent Chain
Latent space $\mathcal{Z}$
$f_d(\mathbf{z})$
Decoder Manifold
On-manifold Initialization
Langevin MC
Features space $\mathcal{X}$
Negative sample to compute negative energy
Initial point of the langevin MC

Training samples and hyper-parameters

Input features
Using AK8 jets because SVJ are expected to be wide

Jet width	Axis major axis minor
N-pronginess	τ_2, τ_3 $C_2^{\beta=0.5}$, $D_2^{\beta=0.5}$
Other	p_T^D, EFP1 log(SoftDrop mass)

Architecture

Fully connected neural net

Hidden layers: $10, 10, 6, 10, 10$

Hyper-parameters

Hyper-parameter	Value
Batch size	256
Reconstruction loss	MSE
Activation	ReLU
Output encoder/ decoder activation	Linear
Optimizer	Adam
Learning rate	1e-5
Dropout	0.
MCMC	PCD
Sampling phase-space	[-3, 3] hypercube

Number of events

m_Φ [GeV]	1000	1500	2000				3000	4000	QCD	$t\bar{t}$
r_inv	0.3	0.3	0.1	0.5	0.3	0.7	0.3	0.3		
Number of events	23k	25k	23k	18k	16k	11k	14k	14k	83k	23k

Number of AK8 jets

Background jets	Leading 2 jets
Signal jets	Only SVJ in leading 2 jets

Train/validation/test splitting

0.7/0.15/0.15

Differences with previous applications

- Existing classification tasks[1,2] are quite different from this one:

Classification task	Computer science paper[1] / MNIST task	HEP paper[2] / task	SVJ search
Data	MNIST images	Jet images	1D array of JSS features
Data representation	32×32 in [0, 1]	40×40 in [0, 1]	8 features, not all bounded
Number of dimensions	1024	1600	8
Network architecture	2D CNN	2D CNN	DNN
Classification	1 MNIST class as OOD	QCD vs $t\bar{t}$ $t\bar{t}$ vs QCD QCD vs SVJ	$t\bar{t}$ vs SVJ

[1] arXiv:2105.05735

[2] arXiv:2206.14225