Unsupervised tagging of semivisible jets with Wasserstein Normalized Autoencoders in CMS

Florian Eble, on behalf of the CMS collaboration

ETH zürich

31/07/2024

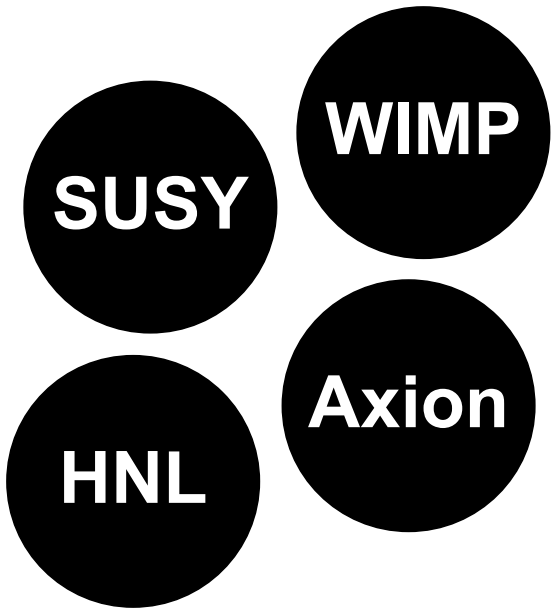
BOOST 2024, Genova

Why signal-agnostic searches for new physics in HEP?

What Dark Matter (DM) is:

- In practice: Anything that's not described by the Standard Model of Particle Physics
- Theoretically: Pick your poison! Supersymmetry (SUSY), Weakly Interacting Massive Particles (WIMPs)...

What if we are not searching at the right place?



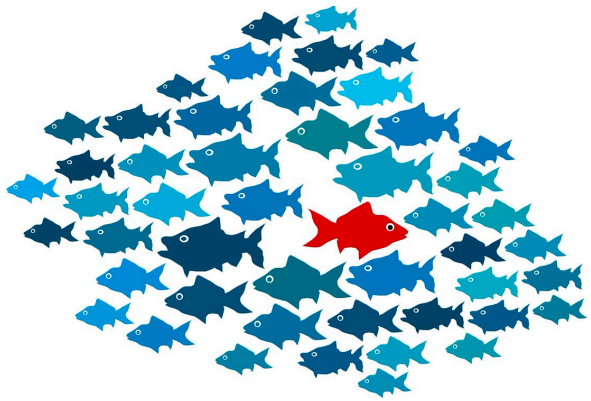
Why signal-agnostic searches for new physics in HEP?

Traditional search

- Targets a specific new physics signal model
- Maximum sensitivity to <u>this</u> signal
- Potentially very little sensitivity to different experimental signatures

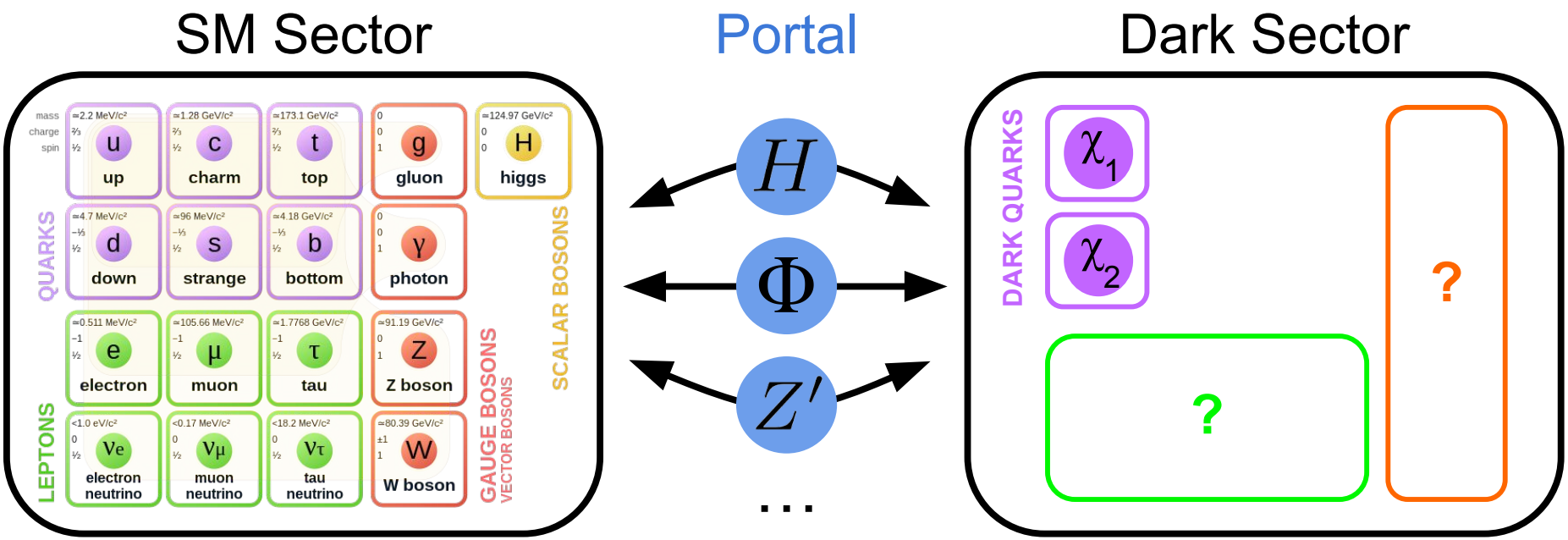
Anomaly detection

- Makes no/few assumptions about the new physics
- Smaller sensitivity compared to traditional search for the target signal
- Sensitive to a wide range of new physics scenarii!



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

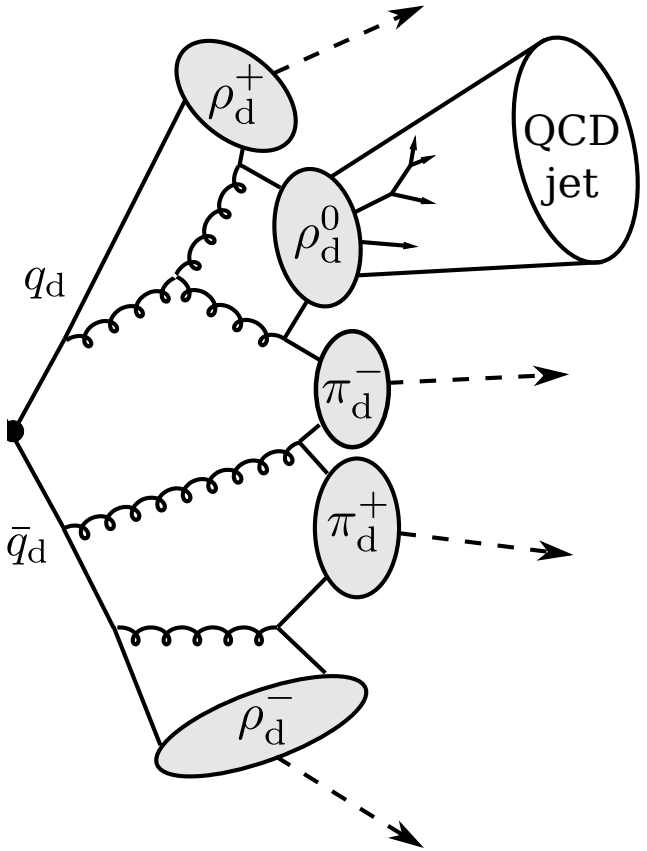


Production of semivisible jets (SVJs) and model parameters

- Dark quarks hadronize in the dark sector
- Unstable dark hadrons promptly decay to SM quarks
- SM quarks hadronize in the SM sector

➔ Semivisible jets (SVJs)
[arXiv:1503.00009, arXiv:1707.05326]

➔ **Different jet substructure**



Model parameters:

- m_{Φ}: Mass of the mediator
- Masses of all dark hadrons fixed to 20 GeV
- r_{inv}: Jet invisible fraction
  - Effective parameter
    Branching ratio DM $\rightarrow q\bar{q}$

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

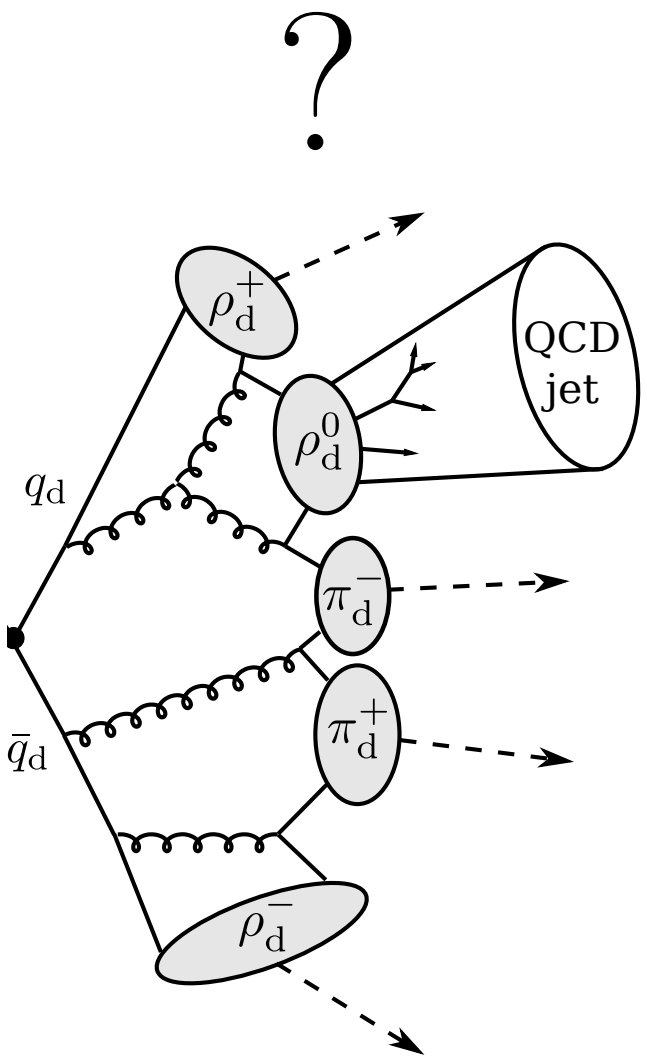
The details of the shower in the dark sector depend on many unknown parameters, e.g.:

- Number of dark colors
- Number of dark flavors
- Masses of the dark hadrons
- Dark hadronization scale

➔ SVJ substructure very model-dependent

➔ Large parameter space to cover

➔ Unsupervised taggers complementary to supervised strategies to explore the parameter space



Anomaly detection with autoencoders (AE)

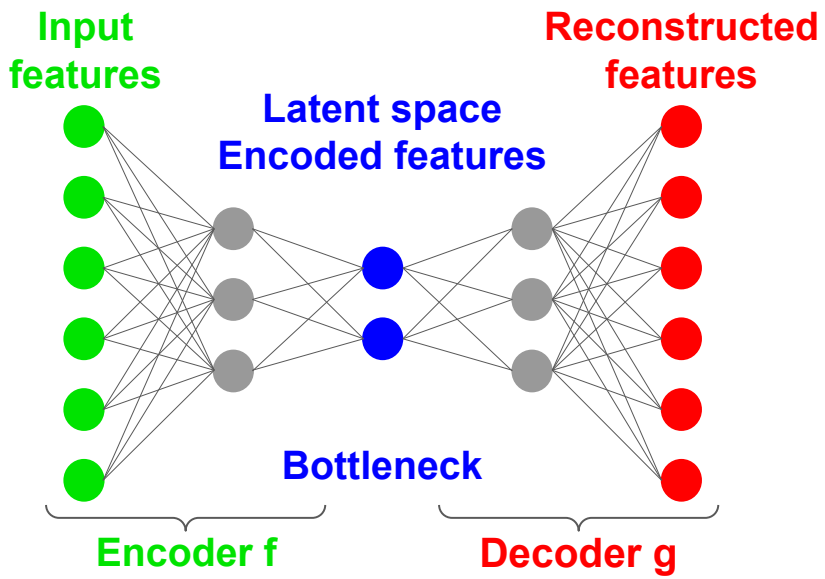
- AEs are trained to minimize the reconstruction error (e.g. MSE) between input and output:

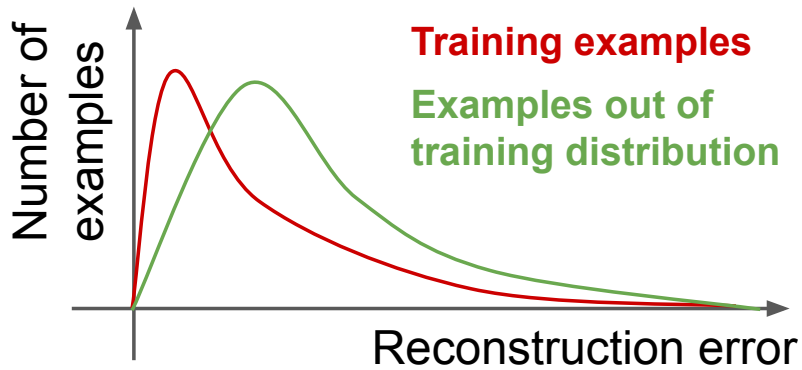
$$L(x) = ||g(f(x)) - x||$$

➔ 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"

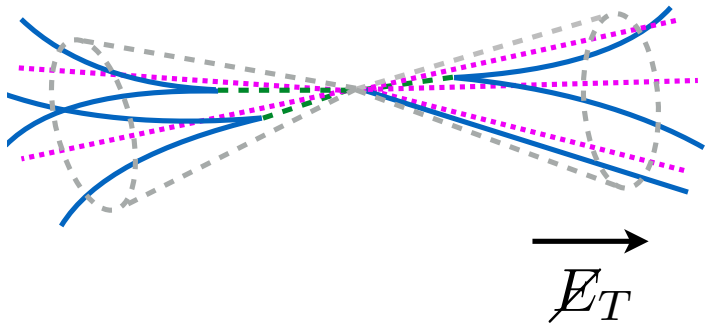




SVJ experimental signature:

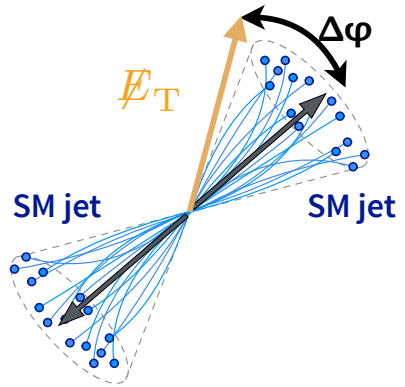
Missing transverse momentum ($\not{E}_{\mathrm{T}}$) aligned with a jet

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



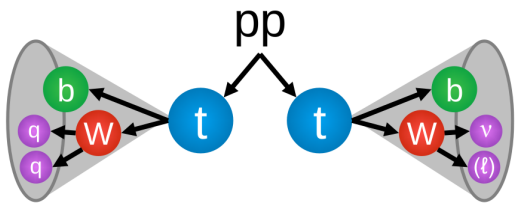
Instrumental $\not{E}_{\mathrm{T}}$

- Mostly QCD: artificial missing transverse energy $\not{E}_{\mathrm{T}}$ aligned with jet from jet energy mismeasurement
- Autoencoder-based anomaly detection proved to work well against QCD jets [arXiv:2112.02864]



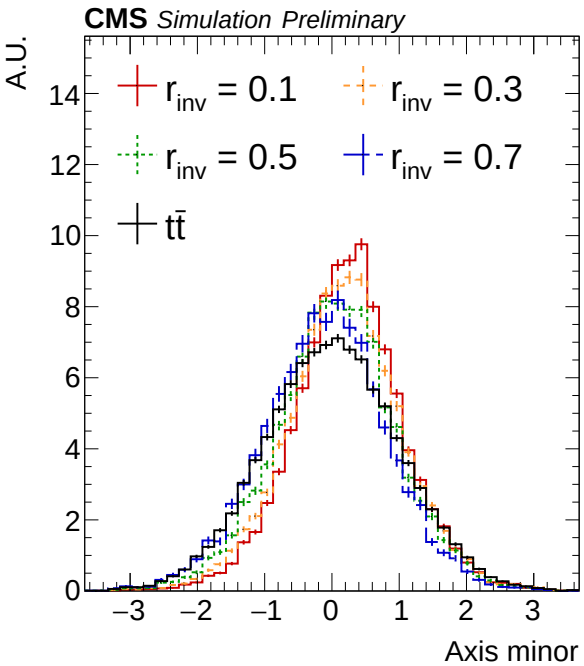
Genuine $\not{E}_{\mathrm{T}}$

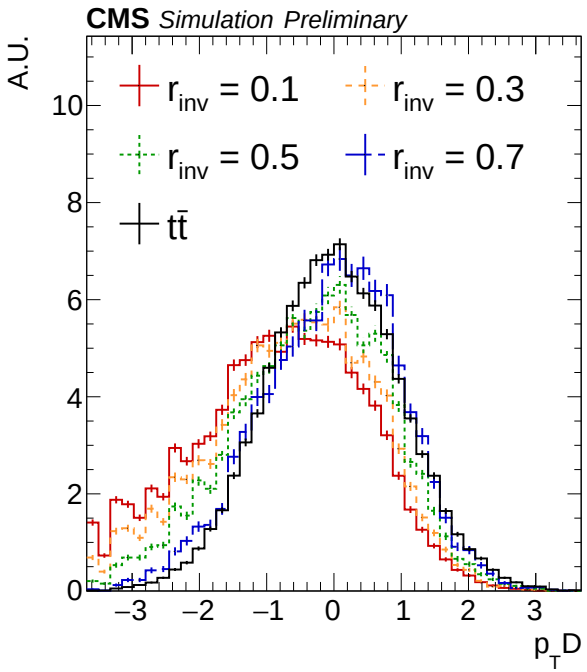
- $t\bar{t}$, W + jets, ... with $W(\rightarrow l\nu)$
- Lost lepton, genuine $\not{E}_{\mathrm{T}}$ from neutrino
- More challenging for anomaly detection

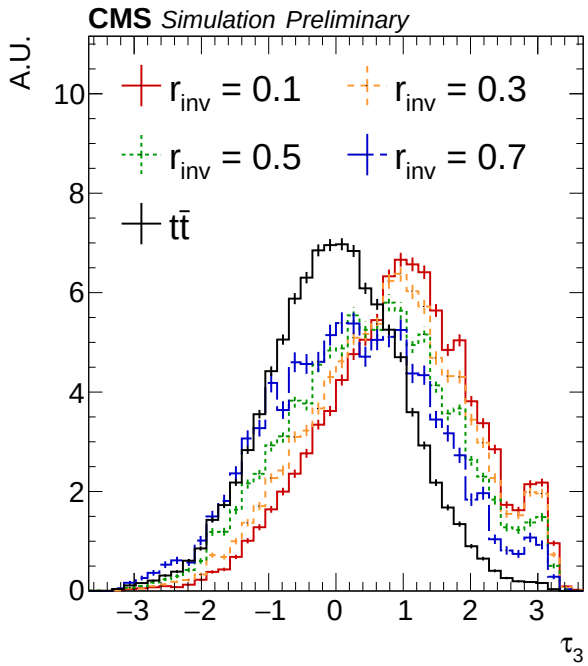


Input features and architecture

- The different ML models presented in the following have the same architecture (fully connected NN, 10-10-6-10-10 hidden neurons) and input features
- They differ only by their loss function
- Input features (quantile morphing to normal distribution), anti-k_t $R = 0.8$ jets:
  - Axis major
  - Axis minor
  - Fist energy flow polynomial EFP1
  - The $C_2^{\beta=0.5}$ energy correlation function
  - Transverse momentum dispersion $p_{\mathrm{T}}^{\mathrm{D}}$
  - Softdrop mass
  - 2-subjettiness τ_2
  - 3-subjettiness τ_3







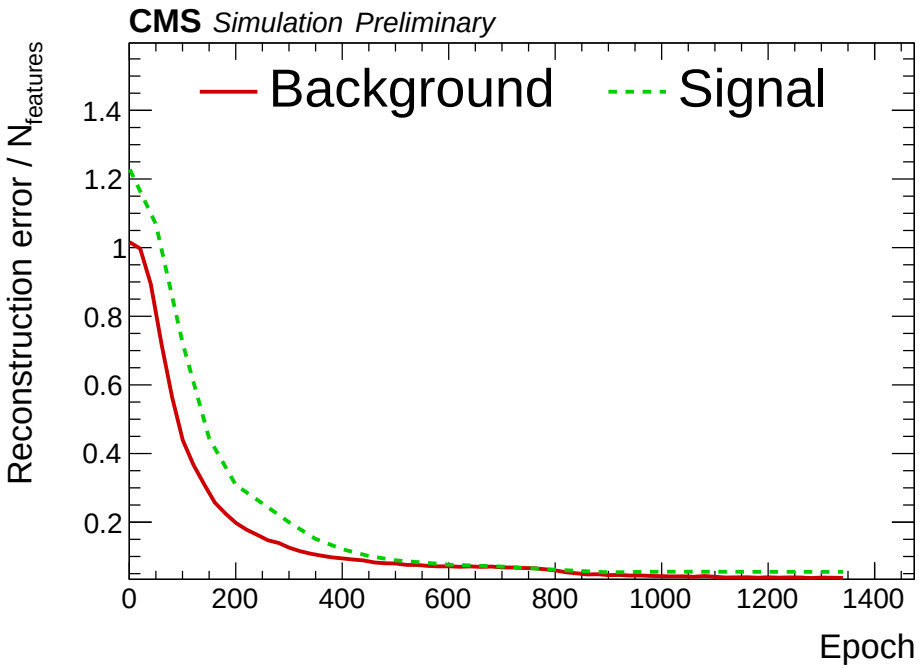
Shortcoming of standard autoencoder

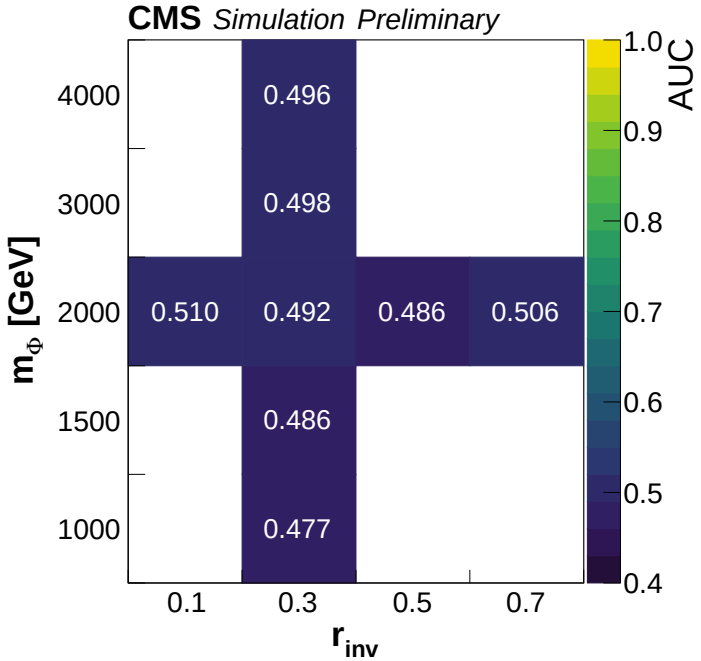
- Training standard AE on simulated background $t\bar{t}$ jets minimizing the MSE between input and reconstructed features[1]

➔ **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!**



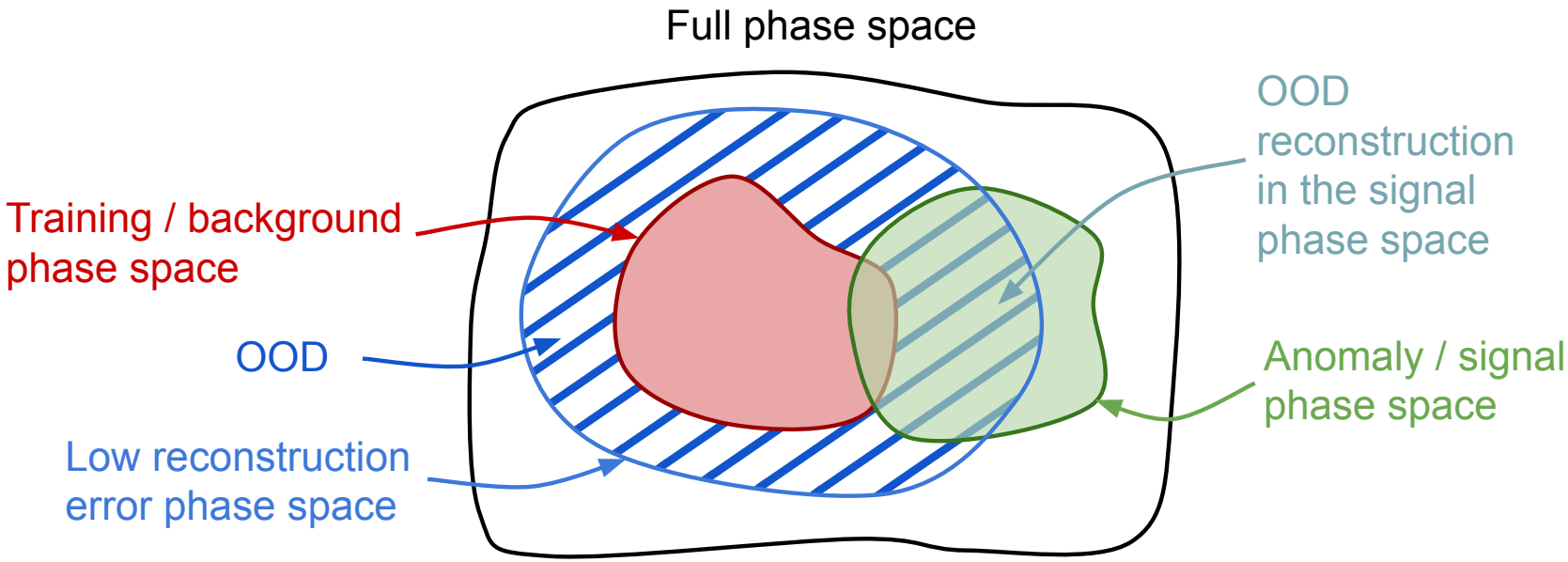


[1] See CMS DP -2023/071

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** / "complexity bias":



The Normalized Autoencoder (NAE) paradigm

Ensure that the low reconstruction error probability distribution matches that of the training data[2]

- Define a probability distribution p_θ so that regions with low reco error E_θ have high probability

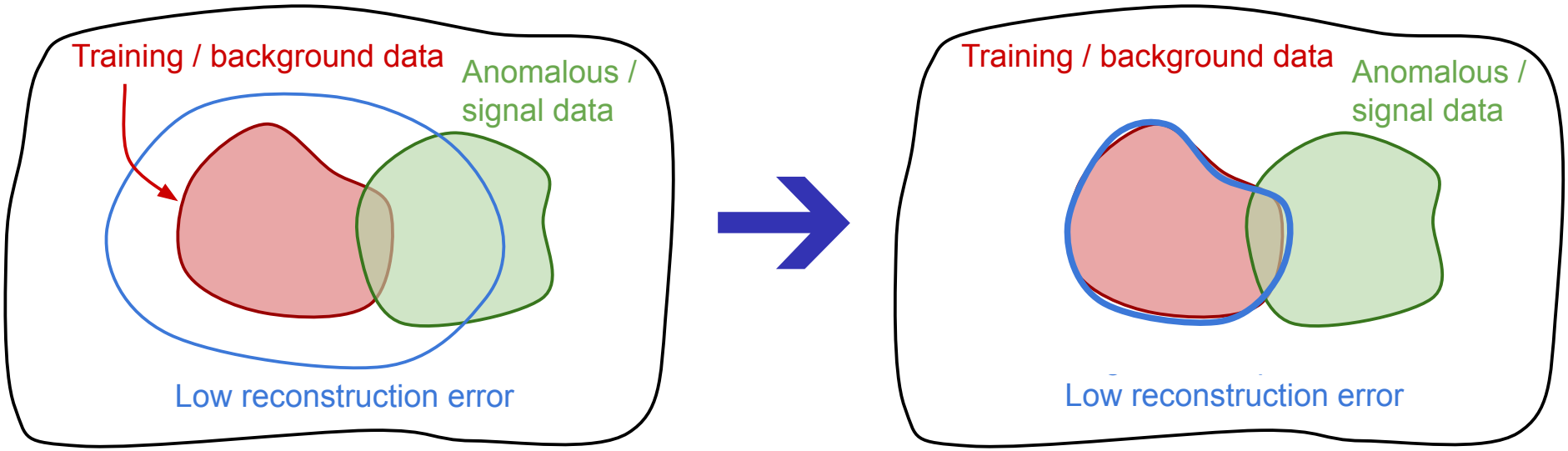
$$p_\theta(x) = \frac{1}{\Omega_\theta} \exp\left(-E_\theta(x)\right)$$

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

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

positive energy E_+ negative energy E_-

- MCMC to sample "negative samples" x' from p_θ and compute their reco error E_-



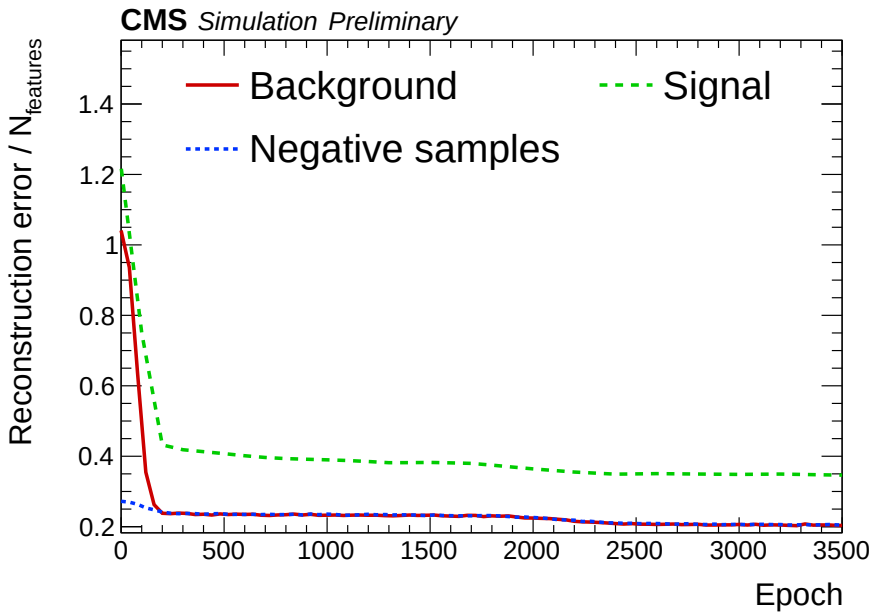
[2] NAE first introduced in arXiv:2105.05735 and used in HEP in arXiv:2206.14225

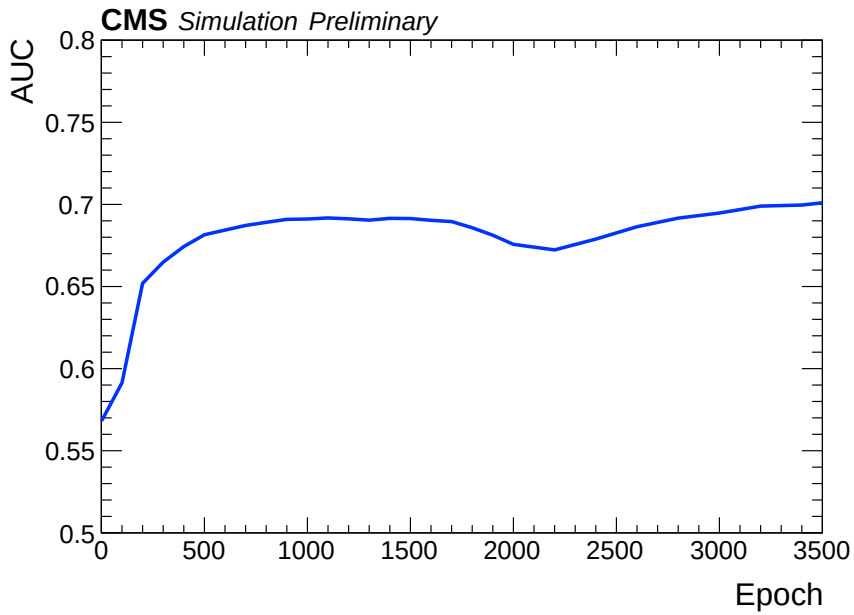
The naive fix

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

- prevent negative loss and the divergence of negative energy
- minimize the positive energy while the energy difference is close to 0[3]:

$$L = \log\left(\cosh\left(E_+ - E_-\right)\right) + \alpha E_+ \qquad \alpha > 0, \text{ hyper-parameter}^4$$





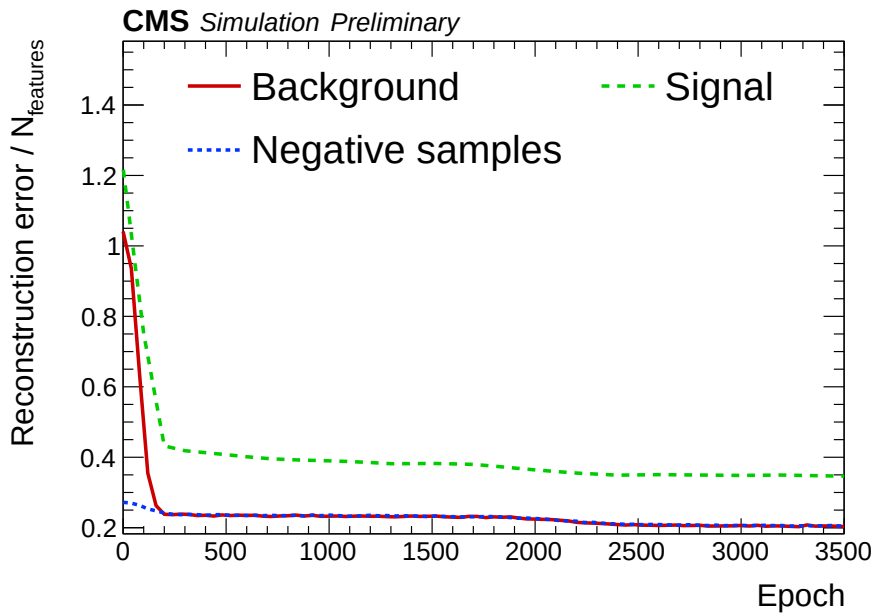
➔ **Signal SVJ reconstruction is efficiently suppressed!**

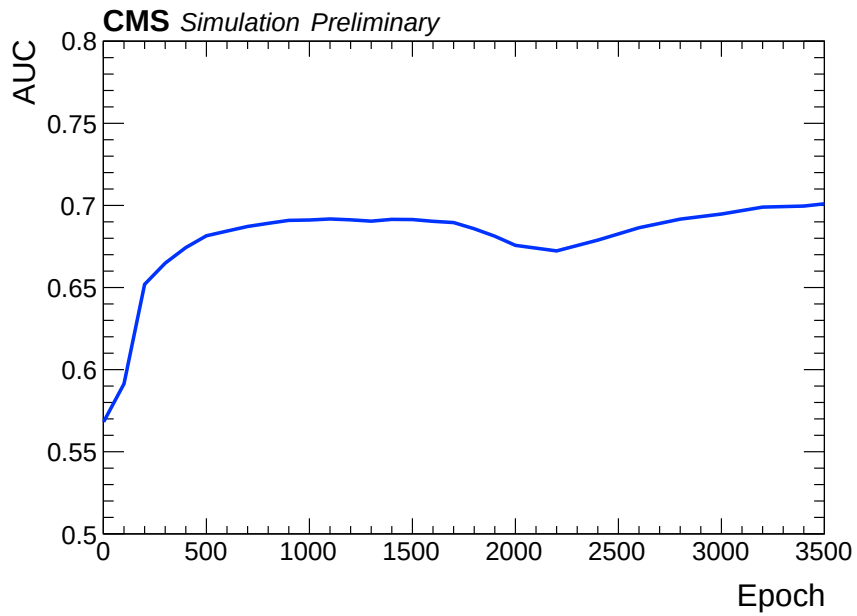
➔ How to define stopping condition in a fully signal-agnostic way?

[3] Relaxing this point, with $L = \log\left(\cosh\left(E_+ - E_-\right)\right)$, the issue developed in next slides is still observed

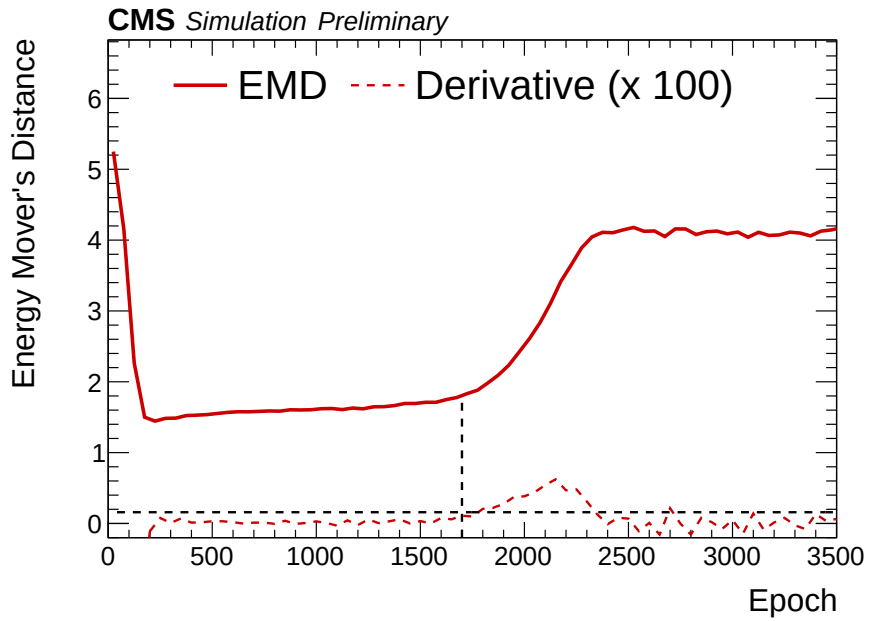
[4] $\alpha = 0.001$ in this case

Wasserstein distance as optimal metric





- Wasserstein distance between $x \sim p_{\text{data}}$ and $x' \sim p_\theta$ (EMD) is a robust measure of the distance between the background and AE probability distributions
- Direct measure of learning $p_\theta = p_{\text{data}}$



Wasserstein distance versus energy difference

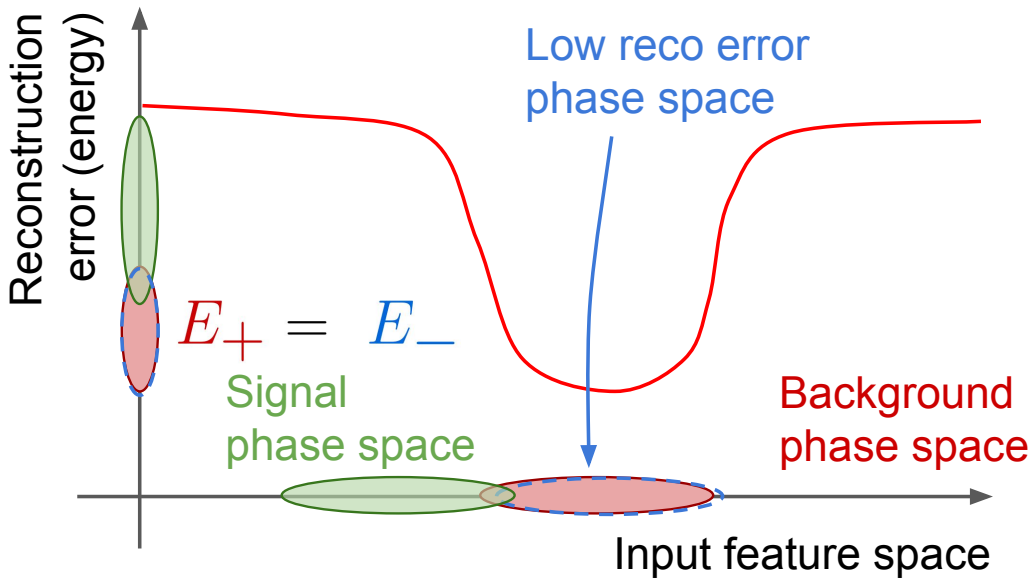


Illustration before collapse:

- Background (positive) and NN (negative) probability distributions match

➔ **Low EMD and low energy difference** between negative and positive probability distributions

➔ Anomalies have large reco error

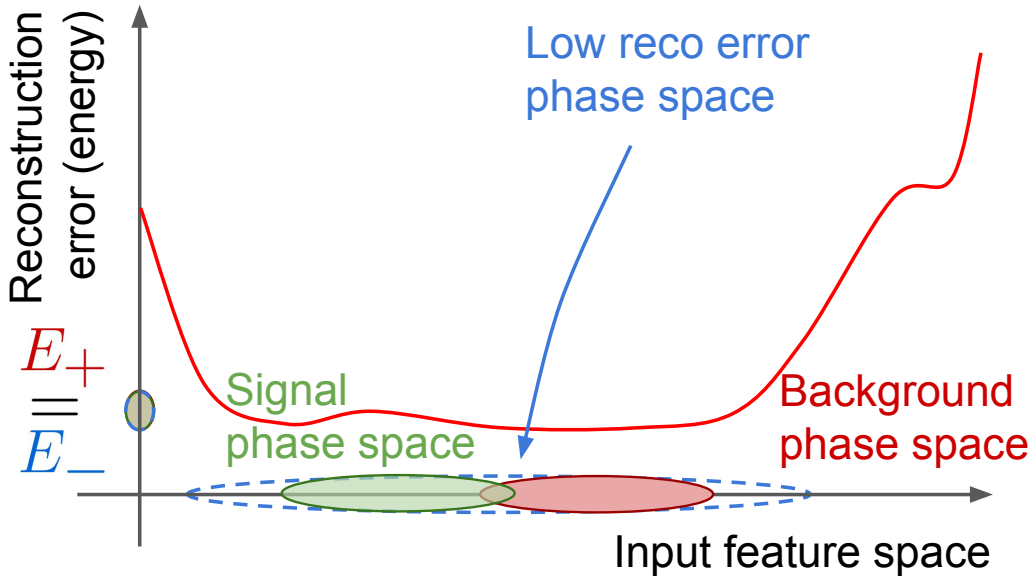


Illustration after collapse:

- Large discrepancy between background and NN probability distributions

➔ **Large EMD but low energy difference** between negative and positive probability distributions

➔ Anomalies are not distinguishable from background

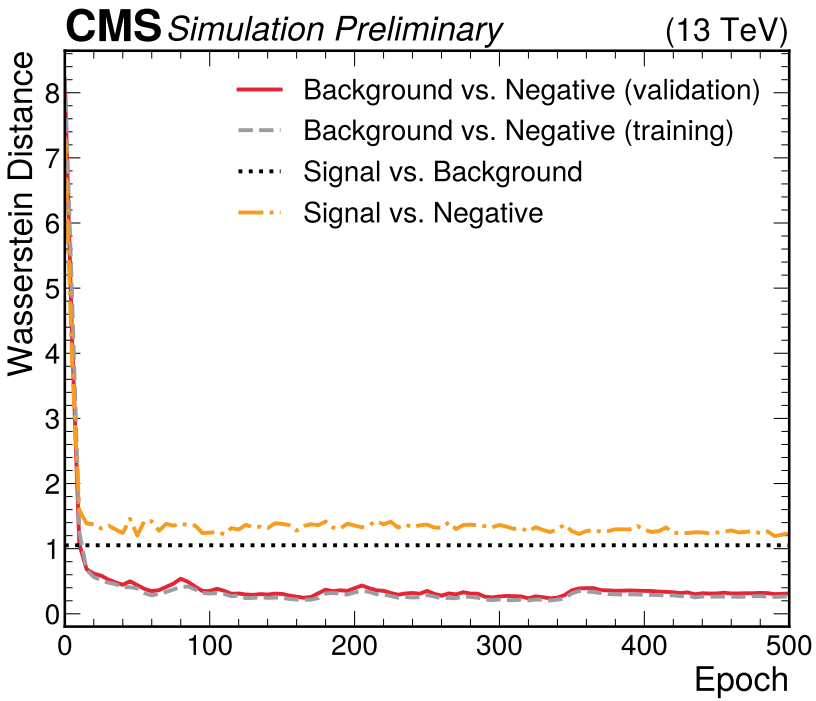
Minimizing the Wasserstein distance

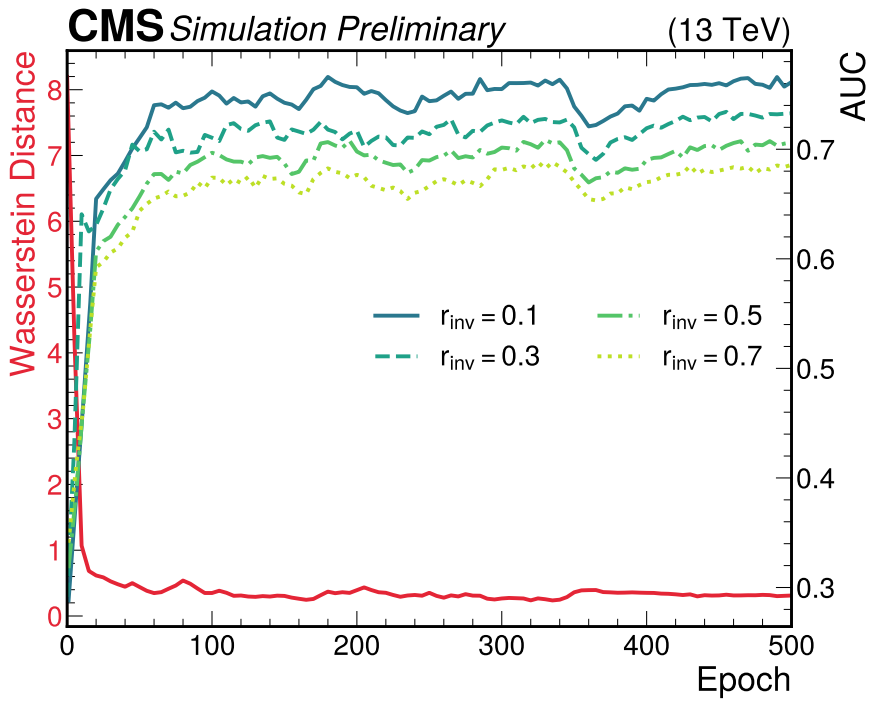
- Wasserstein Normalized Autoencoder (WNAE) loss function (CMS PAS MLG-24-002): Wasserstein distance between $x \sim p_{\text{data}}$ and $x' \sim p_\theta$

$$L_\theta(x) = \inf_{\gamma \in \Pi(p_{\text{data}}, p_\theta)} \mathbb{E}_{(x,x')\sim\gamma}[\|x - x'_\theta\|]$$

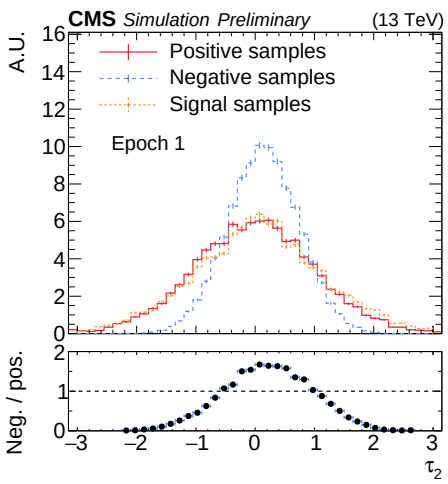
- Anti-correlation between Wasserstein distance and AUC!

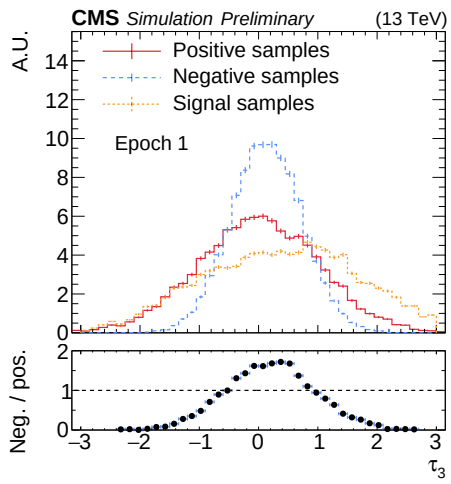
➔ **Fully signal-agnostic training procedure: best epoch is epoch with minimal Wasserstein distance!**

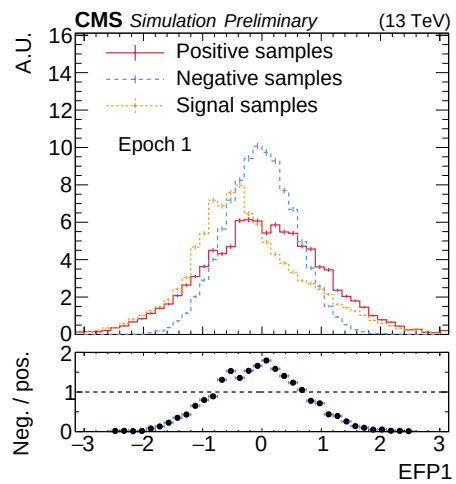


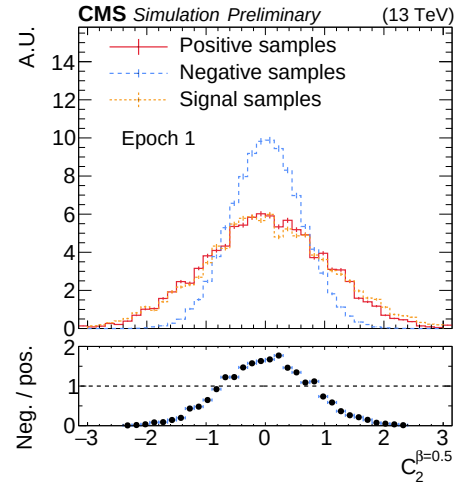


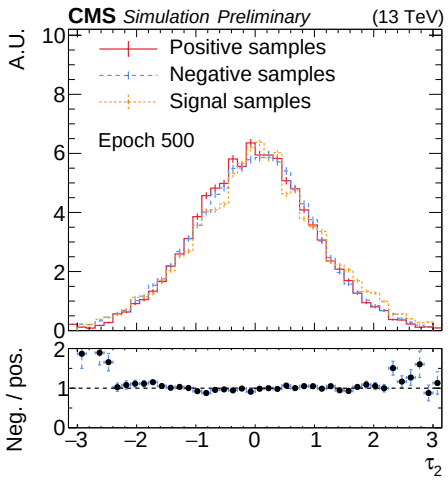
Epoch 1 vs Epoch 500

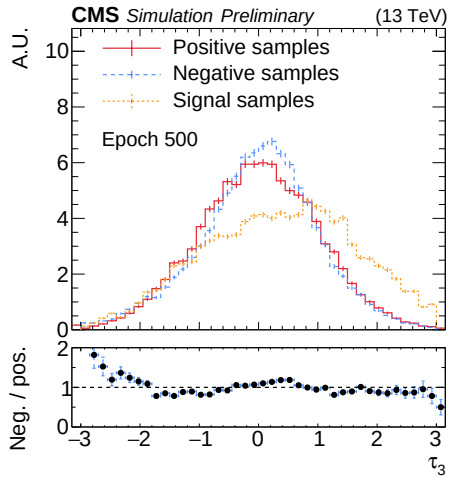


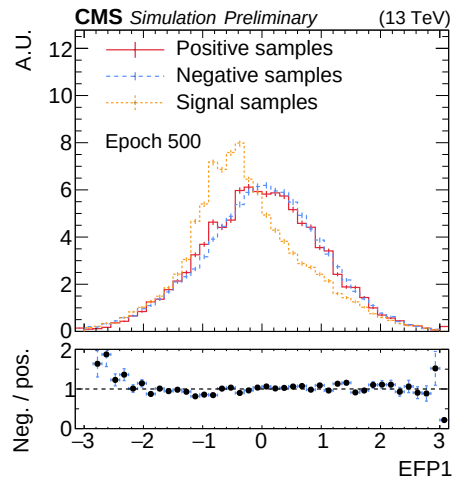


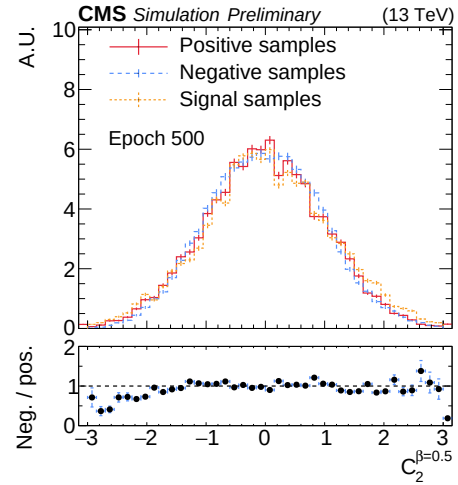




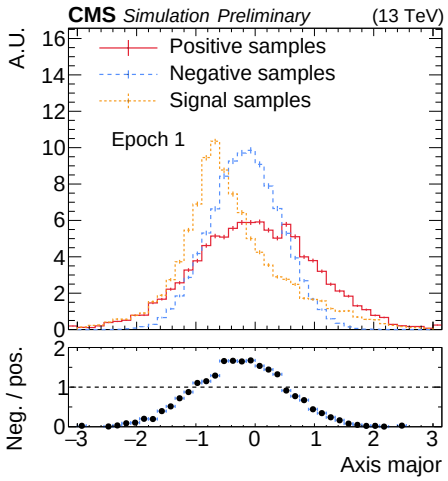


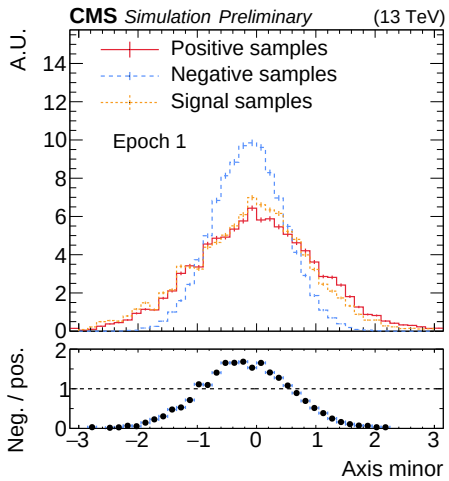


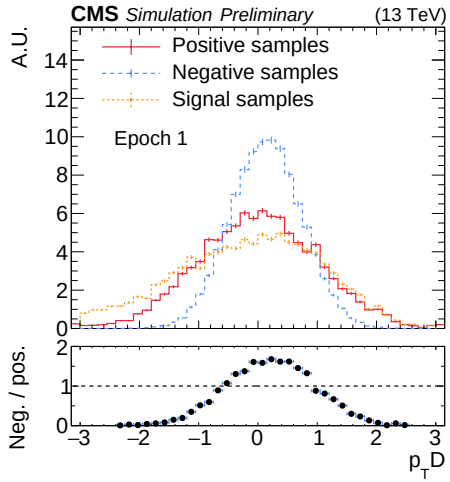


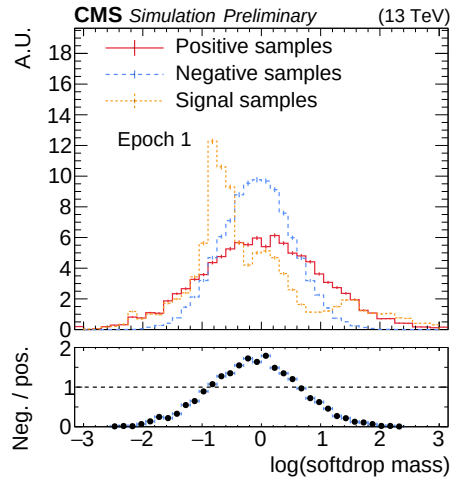


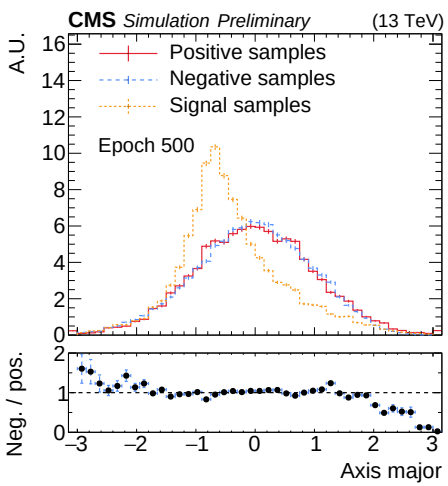
Epoch 1 vs Epoch 500

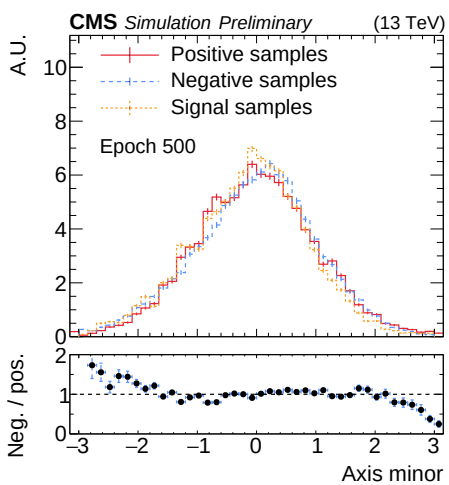


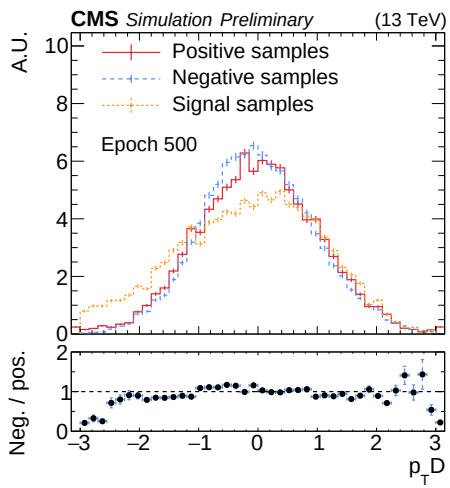


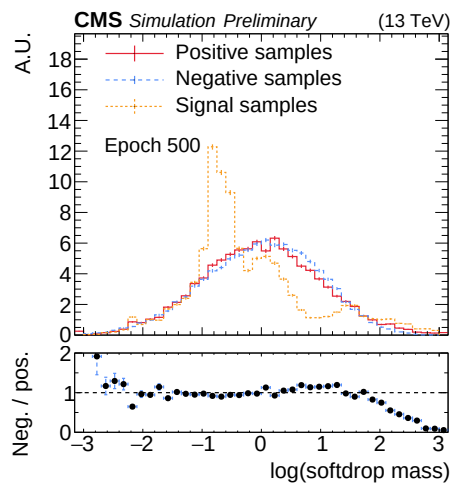






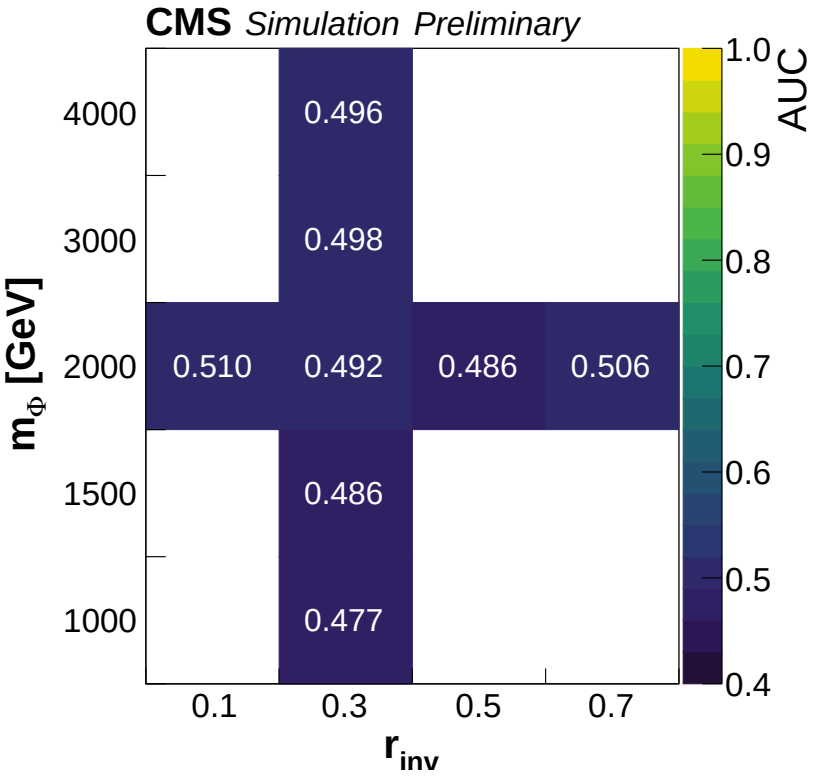




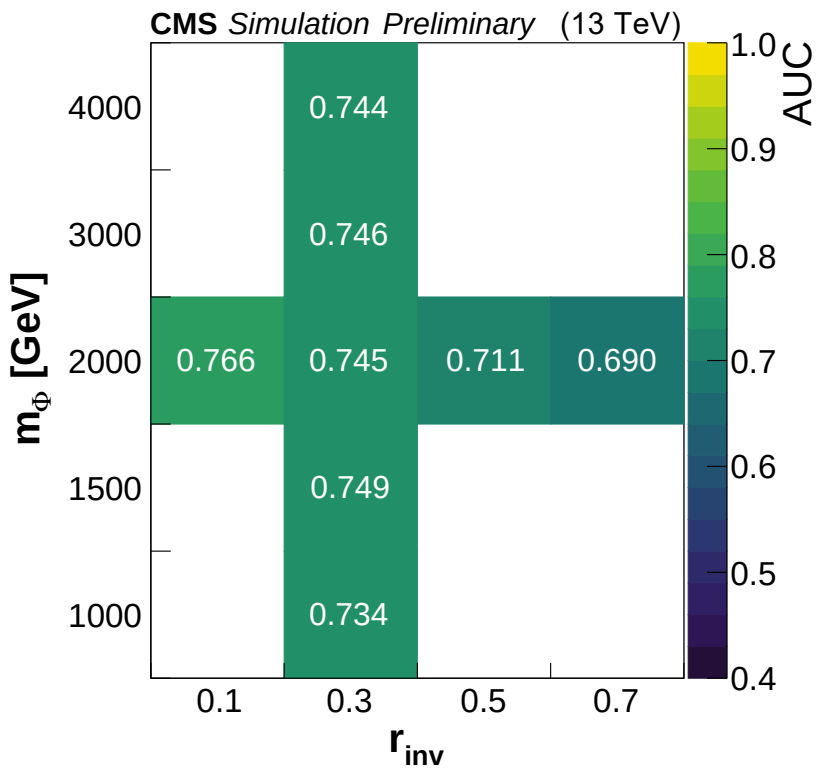


- The WNAE achieves sensible improvement compared to the standard AE!

Standard AE



WNAE



Conclusions

- Standard AEs are prone to out-of-distribution reconstruction because they are free to minimize the reconstruction error outside the training phase space

- Normalized AEs (NAE) propose a mechanism to ensure that the learned probability distribution matches that of the training data
  - The minimization of the NAE loss function is unstable, and ad-hoc regularization is employed to obtain a well-behave loss
  - Found that the minimization of the NAE loss function does not guarantee to suppress OOD reconstruction

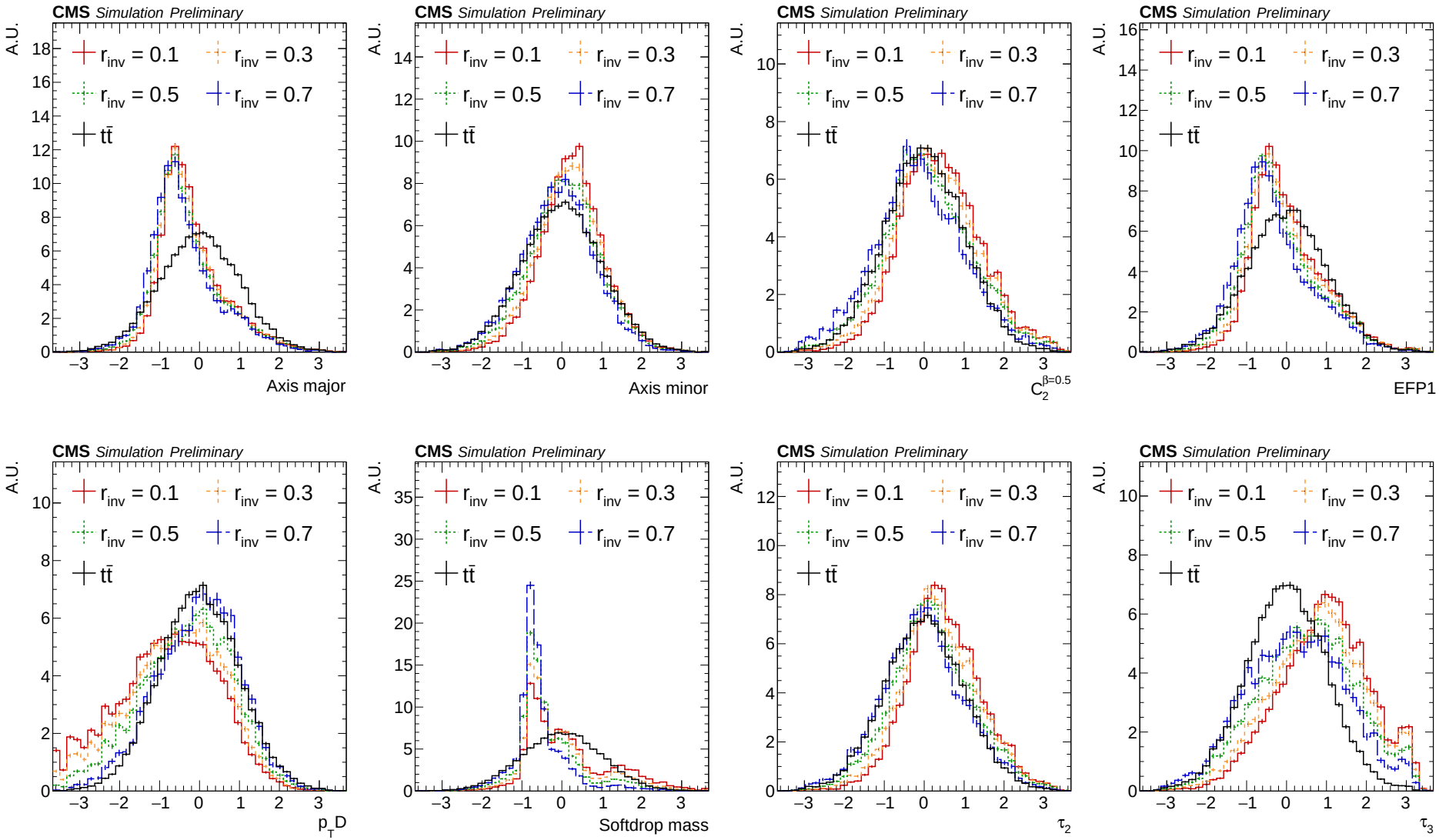
- Wasserstein Normalized AEs (WNAE) is an improvement over NAEs, directly minimizing the Wasserstein distance to between the AE probability distribution and that of the training data, solving the aforementioned issues, and can be trained in a fully signal-agnostic fashion

- The method proposed in this talk is general and not limited to the search for SVJs!

Backup

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



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:

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

Gradient of partition function

- Calculating the partition function Ω_θ is in general intractable
- Can be circumvented when using gradient descent to find the optimum, since the gradient of the partition function can be calculated:

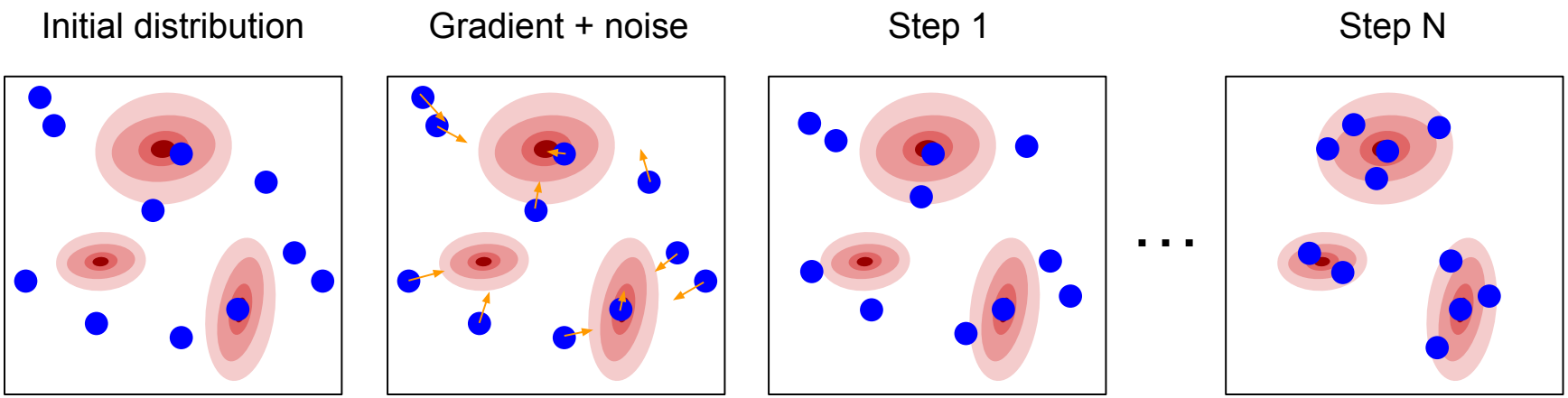
$$
\begin{aligned}
\nabla_\theta \log \Omega_\theta &= \frac{1}{\Omega_\theta} \nabla_\theta \Omega_\theta \\
&= \frac{1}{\Omega_\theta} \int_{\mathcal{B}} dx \nabla_\theta \exp(-E_\theta(x)) \\
&= \frac{1}{\Omega_\theta} \int_{\mathcal{B}} dx \exp(-E_\theta(x)) \nabla_\theta (-E_\theta(x)) \\
&= \int_{\mathcal{B}} dx \frac{1}{\Omega_\theta} \exp(-E_\theta(x)) \nabla_\theta (-E_\theta(x)) \\
&= \int_{\mathcal{B}} dx p_\theta(x) \nabla_\theta (-E_\theta(x)) \\
&= -\mathbb{E}_{x \sim p_\theta(x)} [\nabla_\theta E_\theta(x)],
\end{aligned}
$$

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 $\rho_\infty = p$



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

- T and σ chosen so that large AUC is obtained under the condition that the EMD is low and the MCMC samples 1D distributions match that of data

[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]

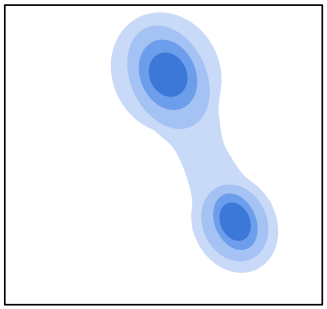
- Random initial distribution for first MCMC
- The model changes only slightly during parameter update
- Thus, for subsequent chains, initialize chain at the state in which it ended for the previous model
- Possibility to randomly re-initialize a small fraction of the samples

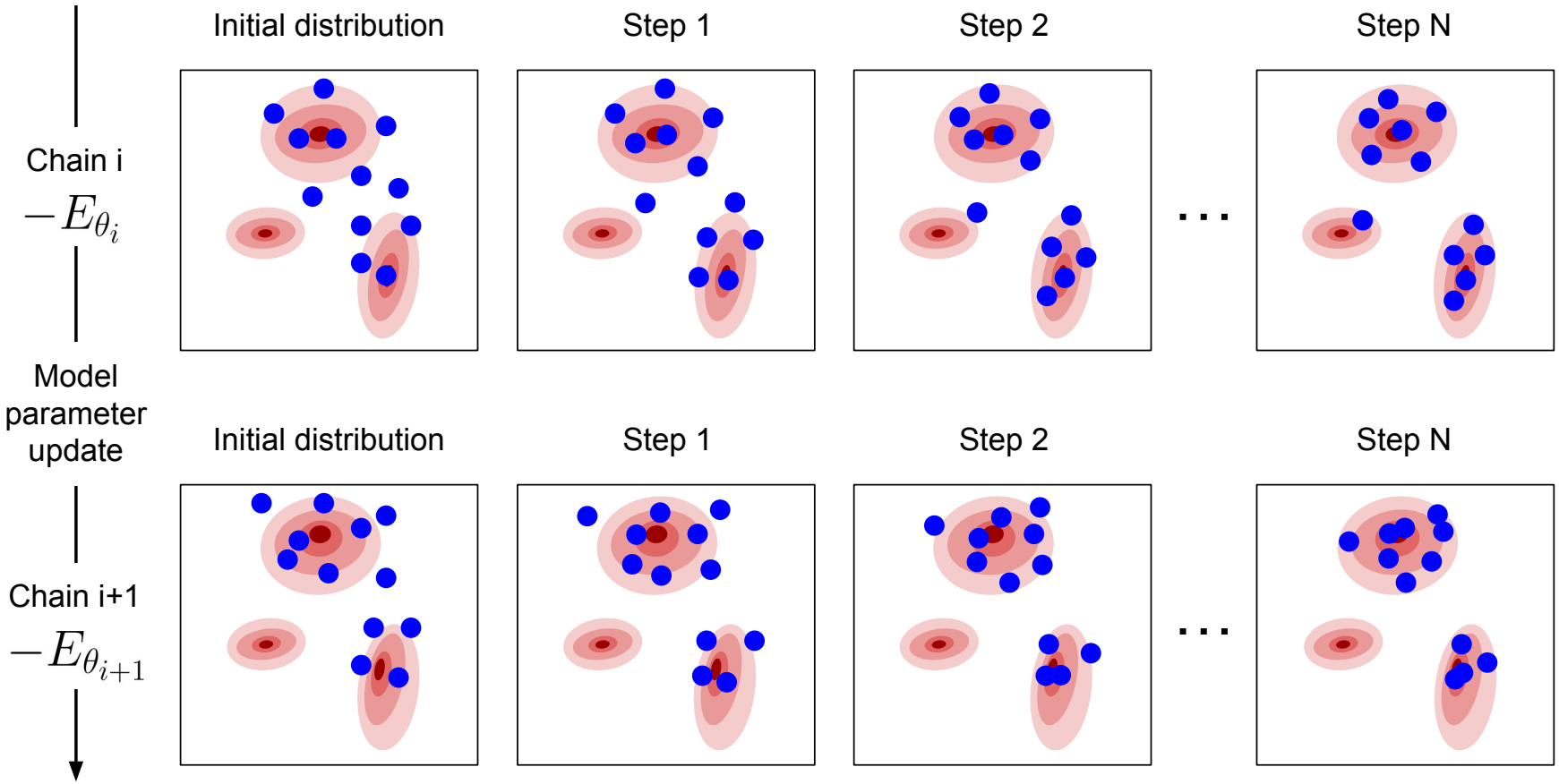
[1] Neural Comput 2002; 14 (8)
[2] PCD paper
[3] Illustration in backup slide 8
[4] Illustration in backup slide 9

Principle of CD

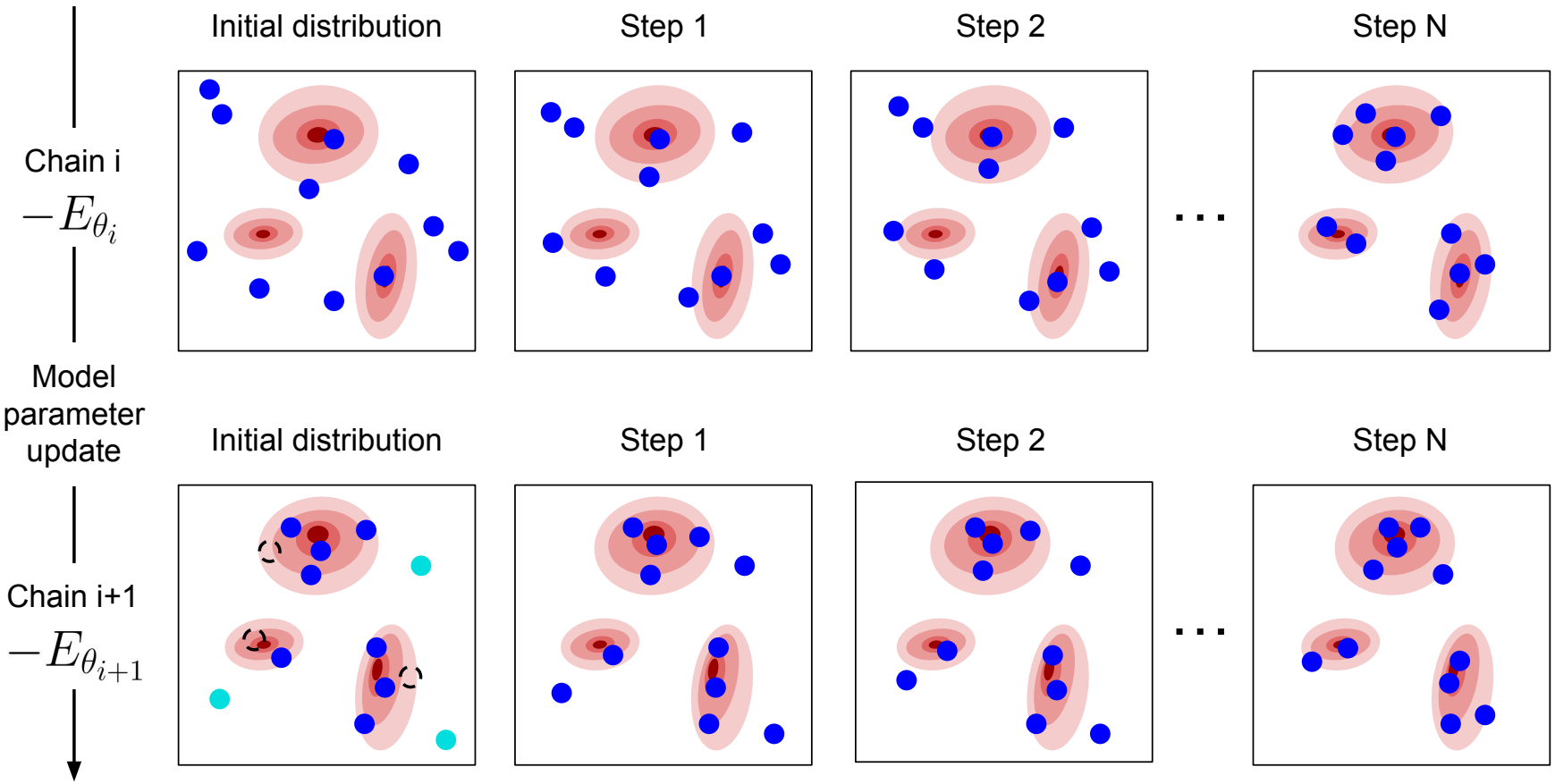
Example of a failure mode of CD: High probability mode far from training data distribution is not sampled

Training data distribution:





Principle of PCD



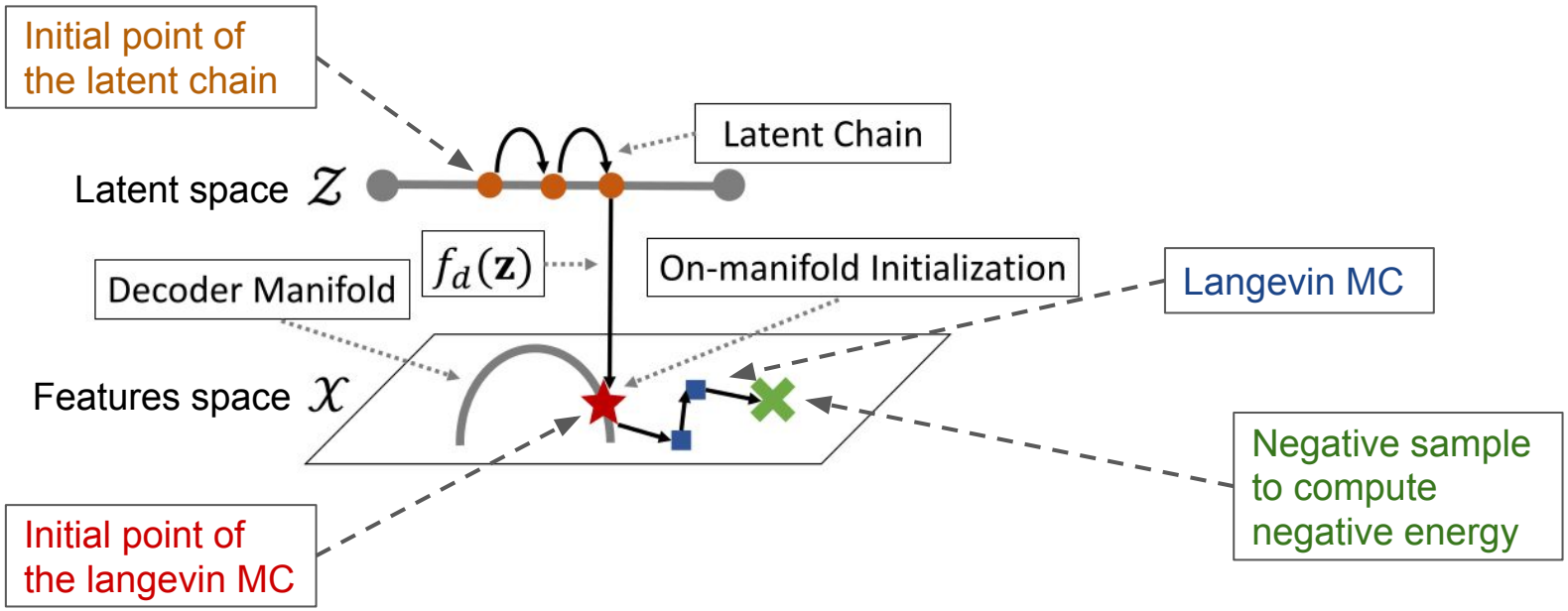
On-Manifold Initialization

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



MCMC in Normalized Autoencoder (NAE)

Loss

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

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:

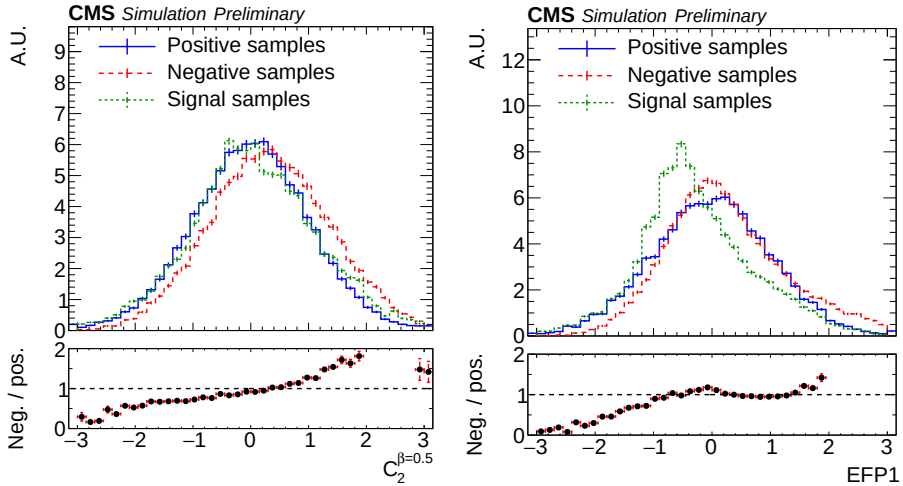
$$x'_{i+1} = x'_i - \lambda_i \nabla_x E_\theta(x'_i) + \sigma_i \epsilon \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

drift diffusion

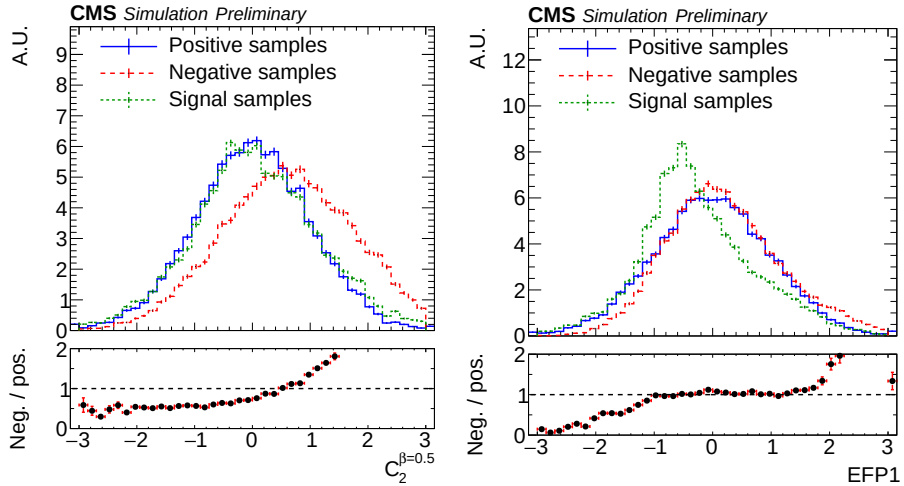
- Repeat with several points $x'^{(j)}_0$, the negative samples are the $x'^{(j)}_n$

NAE with “logcosh” loss - Negative samples histograms

- 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).

Minimizing the Wasserstein distance

- WNAE loss function:

$$L_\theta(x) = \inf_{\gamma \in \Pi(p_{\text{data}}, p_\theta)} \mathbb{E}_{(x,x')\sim\gamma}[\|x - x'_\theta\|]$$

- Differentiable implementation of the Wasserstein distance from POT

- **The dependency on θ enters in the sampling of the negative samples $x' \sim p_\theta$:**

$$x'_{i+1} = x'_i - \nabla_x E_\theta(x'_i) + \sigma\epsilon$$

- In PyTorch jargon, need to keep track of two separate computational graphs:
  - **MCMC step**: gradient wrt the input feature space, to compute $\nabla_x E_\theta(x'_{\theta,i})$
  - **Backpropagation step**: gradient wrt the NN weights θ, to compute $\nabla_\theta L_\theta$:

$$L_\theta(x) = \inf_{\gamma \in \Pi(p_{\text{data}}, p_\theta)} \mathbb{E}_{(x,x')\sim\gamma}[\|x - x'_\theta\|]$$