Versatile Energy-Based Models for High Energy Physics

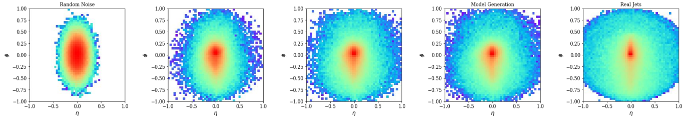
Taoli Cheng

Joint work with Aaron Courville (Mila, University of Montreal)

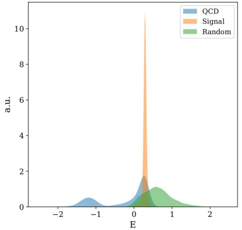
Partially based on arXiv: 2302.00695

Feb. 14, 2023 @ IML, CERN

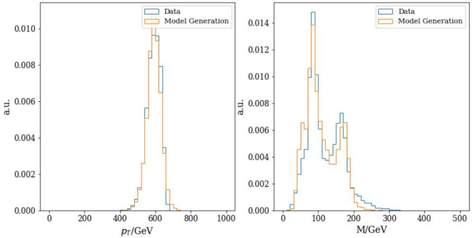
Generative Modelling



Anomaly Detection



Hybrid Modelling



$$p(\mathbf{x}) = \frac{\exp(-E(\mathbf{x}))}{Z}$$

Introduction to Energy-Based Models

- Probabilistic modeling:
  - x represents any high-dimensional data point
  - Model the probability of each occurrence p(x)
- Energy-based models (EBMs) $p(\mathbf{x}) = \frac{\exp(-E(\mathbf{x}))}{Z}$
  - Popular generative modeling method before deep learning (e.g., Restricted Boltzmann Machine)
  - Inspired by Gibbs distribution in statistical physics
  - Flexibility in the energy function: any scalar could serve as the energy, since **exp(-E)** gives a non-negative un-normalized probability
  - Bottom-up approach for generation (does not need a generator or a well-designed reconstruction error)

Introduction to Energy-Based Models

$$p(\mathbf{x}) = \frac{\exp(-E(\mathbf{x}))}{Z}$$

- x: the state of a system or an input configuration
- E(x): energy function, can be parameterized by modern deep neural networks
- Z: partition function or normalizing constant

$$Z = \int \tilde{p}(\mathbf{x}) d\mathbf{x} = \int \exp(-E_\theta(\mathbf{x})) d\mathbf{x}$$

Training EBMs | Contrastive Divergence

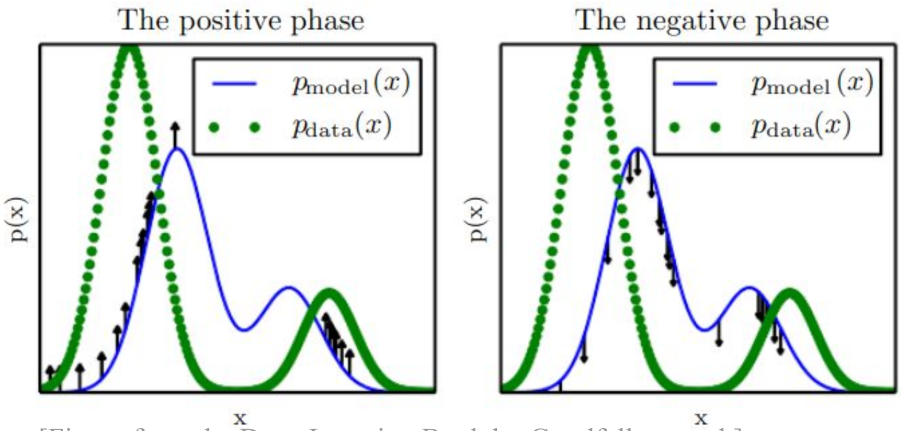
Training of EBMs can be achieved with Maximum Likelihood Estimation.

$$\log p(\mathbf{x}) = -E(\mathbf{x}) - \log Z$$ intractable

$$\nabla_\theta \mathcal{L}(\theta) = -\mathbb{E}_{p_D(\mathbf{x})}[\nabla_\theta \log p_\theta(\mathbf{x})]$$
$$\simeq \mathbb{E}_{p_D(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x}^+)] - \mathbb{E}_{p_\theta(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x}^-)],$$

Estimated with Markov Chain Monte Carlo

Usually takes the form of contrasting energies of *positive samples* and *negative samples*



[Figure from the Deep Learning Book by Goodfellow et al.]

Gradient-based MCMC

Negative phase: MCMC samples q(x) to estimate the model distribution p(x)

Langevin Dynamics (Welling & Teh, 2011) initializing from random noises. At each MCMC step:

$$\mathbf{x}_{k+1}^{-} = \mathbf{x}_{k}^{-} - \frac{\lambda^2}{2}\nabla_{\mathbf{x}} E_\theta(\mathbf{x}_k^{-}) + \lambda \cdot \epsilon, \text{ with } \epsilon \sim \mathcal{N}(0,1)$$

Gradient descent

Diffusion term

Kullback-Leibler Divergence-Improved Training (Optional)

KL-improved training (Du et al, 2020): include the KL divergence between the model distribution and the MCMC estimation

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{p_D(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x}^+)] - \mathbb{E}_{q_\theta(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x}^-)] - \frac{\partial q_\theta(\mathbf{x})}{\partial \theta} \frac{\partial D_{\mathrm{KL}}(q_\theta(\mathbf{x}) || p_\theta(\mathbf{x}))}{\partial q_\theta(\mathbf{x})}$$

$$\mathcal{L} = \mathcal{L}_{\mathrm{CD}} + \mathcal{L}_{\mathrm{KL}}, \text{ with } \mathcal{L}_{\mathrm{KL}} = \mathbb{E}_{q(\mathbf{x})}[E_{\hat{\theta}}(\mathbf{x})] + \mathbb{E}_{q_\theta(\mathbf{x})}[\log(q_\theta(\mathbf{x}))]$$

Kullback-Leibler Divergence-Improved Training (Optional)

KL-improved training (Du et al, 2020): include the KL divergence between the model distribution and the MCMC estimation

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{p_D(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x}^+)] - \mathbb{E}_{q_\theta(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x}^-)] - \frac{\partial q_\theta(\mathbf{x})}{\partial \theta} \frac{\partial D_{\mathrm{KL}}(q_\theta(\mathbf{x}) || p_\theta(\mathbf{x}))}{\partial q_\theta(\mathbf{x})}$$

$$\mathcal{L} = \mathcal{L}_{\mathrm{CD}} + \mathcal{L}_{\mathrm{KL}}, \text{ with } \mathcal{L}_{\mathrm{KL}} = \mathbb{E}_{q(\mathbf{x})}[E_{\hat{\theta}}(\mathbf{x})] + \mathbb{E}_{q_\theta(\mathbf{x})}[\log(q_\theta(\mathbf{x}))]$$

Entropy term, difficult to estimate

Kullback-Leibler Divergence-Improved Training (Optional)

KL-improved training (Du et al, 2020): include the KL divergence between the model distribution and the MCMC estimation

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{p_D(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x}^+)] - \mathbb{E}_{q_\theta(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x}^-)] - \frac{\partial q_\theta(\mathbf{x})}{\partial \theta} \frac{\partial D_{\mathrm{KL}}(q_\theta(\mathbf{x}) || p_\theta(\mathbf{x}))}{\partial q_\theta(\mathbf{x})}$$

$$\mathcal{L} = \mathcal{L}_{\mathrm{CD}} + \mathcal{L}_{\mathrm{KL}}, \text{ with } \mathcal{L}_{\mathrm{KL}} = \mathbb{E}_{q(\mathbf{x})}[E_{\hat{\theta}}(\mathbf{x})] + \mathbb{E}_{q_\theta(\mathbf{x})}[\log(q_\theta(\mathbf{x}))]$$

In our work, we ignore the entropy term and thus optimize the upper-bound of the KL term

EBMs for High Energy Physics: A Framework

- Modelling high-dimensional data distribution directly
- Physics inductive biases or incorporate sophisticated architectures
- Multiple use-cases
- High performance and less spurious correlation

Topic	Practice
Generative modeling	Parameterized event generation
OOD detection	Model-independent new physics search
Hybrid modeling	Classifier combined with EBMs

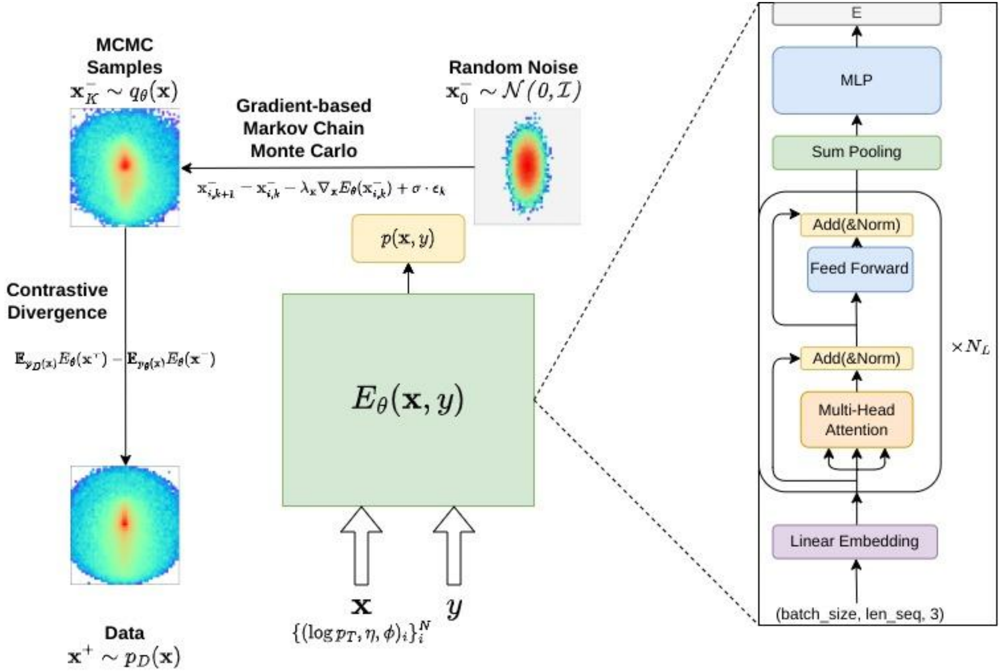
Setup

- We work on simulated jets produced from 13 TeV LHC pp collision.
- Inputs from particle-flow objects: $\{(p_T, \eta, \phi)\}_i^N$
- p(x): train on large-radius (R=1.0) QCD jets or QCD/W/Top jets (for hybrid modelling)
- Note: fewer MCMC steps (24) in training, more steps in validation

Data	
input features	$\{(\log(p_T), \eta, \phi)_i\}_{i=1}^N$
input length	N=40 with zero-padding
Energy Function	
Number of layers	8
Model dimension	128
Number of heads	16
Feed-forward dimension	1024
Dropout rate	0.1
Normalization	None
MCMC	
Number of steps	24
Step size	0.1
Buffer size	10000
Resample rate	0.05
Noise	$\epsilon = 0.005$
Regularization	
L2 Regularization	0.1
Training	
Optimizer	Adam ($\beta_1 = 0.0, \beta_2 = 0.999$)
Learning rate	1e-4 (decay rate $\gamma = 0.98$)

Schematic

- Energy function: maps high-dimensional inputs to a scalar (X, y) -> E
- Flexibility in the energy function: can be modelling with sophisticated architectures (here we use a transformer) without bothering designing an explicit generation or effective reconstruction error (as in VAEs)
- Low-level inputs with or w/o labels



Applications | Generative Modelling

Once we have a well-trained energy function E(x), we have

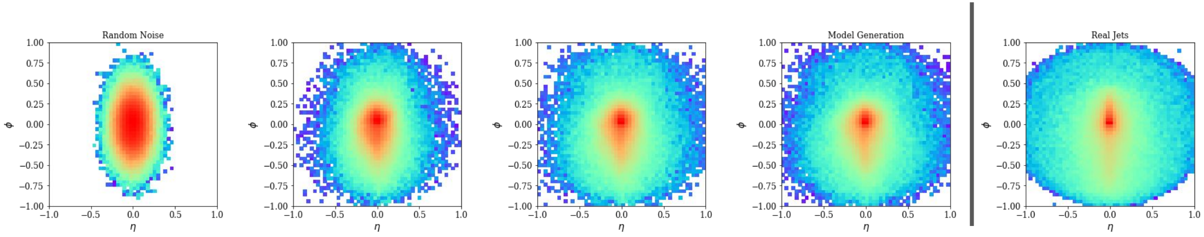
- Implicit generation: $\mathbf{x}_{k+1}^{-} = \mathbf{x}_{k}^{-} - \frac{\lambda^2}{2}\nabla_{\mathbf{x}} E_\theta(\mathbf{x}_{k}^{-}) + \lambda \cdot \epsilon, \text{ with } \epsilon \sim \mathcal{N}(0,1)$
    - Sample from noises → Gradient-based Langevin Dynamics →realistic samples
- Flexibility at test-time generation, as long as the energy function is well trained, we can use different sampling strategies (step size, dynamic sampling, other sampling strategies, etc.).

Applications | Generative Modelling

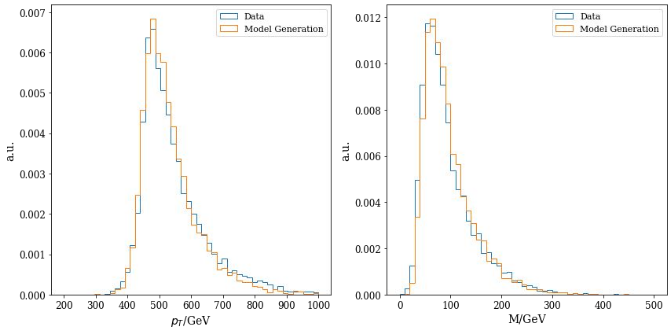
Random Noises → Gradient-based MCMC → Data distribution

$$\mathbf{x}_{k+1}^{-} = \mathbf{x}_{k}^{-} - \frac{\lambda^2}{2}\nabla_{\mathbf{x}} E_\theta(\mathbf{x}_{k}^{-}) + \lambda \cdot \epsilon, \text{ with } \epsilon \sim \mathcal{N}(0,1)$$

Jet images



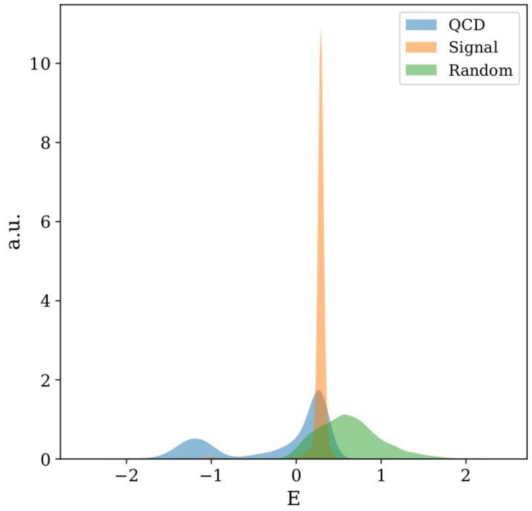
High-level observables

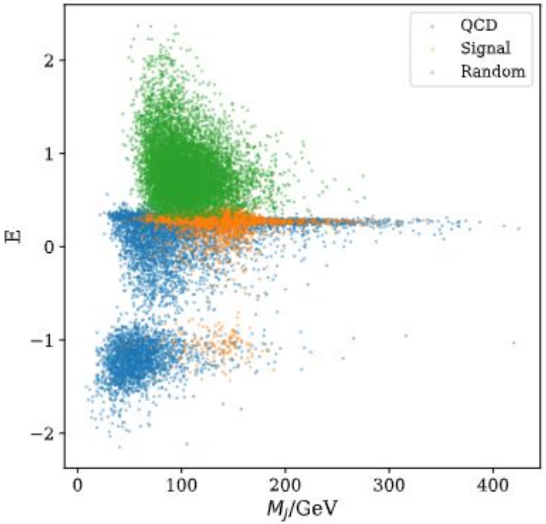


- Use a colder model (lower temperature ~ small MCMC step size) at test-time generation

Applications | Model-Independent New Physics Searches

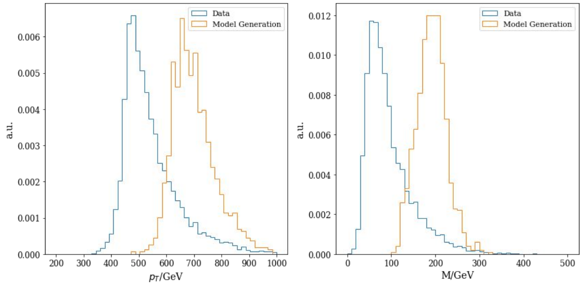
Method: model p(x) of QCD jets → (thresholding p(x) < s: **E(x) > e**) → detect non-QCD signal jets with higher energies



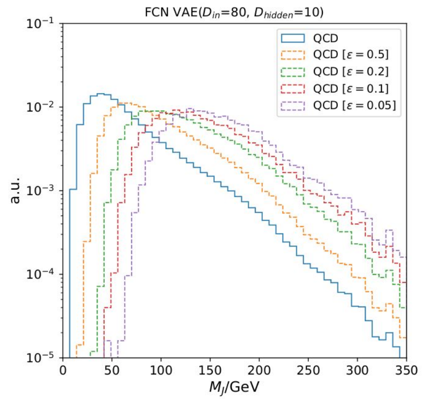


Applications | Model-Independent New Physics Searches

- Mass correlation in anomalous jet tagging
  - (Vatiational) Autoencoder (reconstruction error-based): jet constituent numbers, jet complexity
  - Jet Classifier: in-distribution jet masses
- Underlying reason for EBMs not presenting mass correlation: larger mass modes already be covered during the negative sampling process

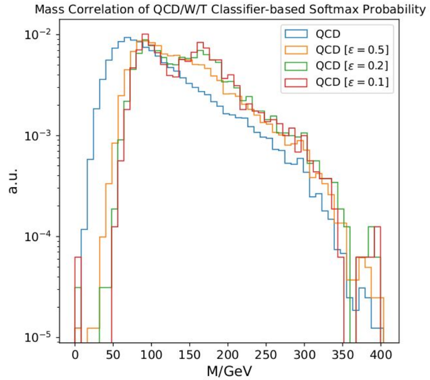


Here shows MCMC samples from an early stage model



Variational Autoencoder

[arXiv:2007.01850]



Multiclass SM Jet Classifier

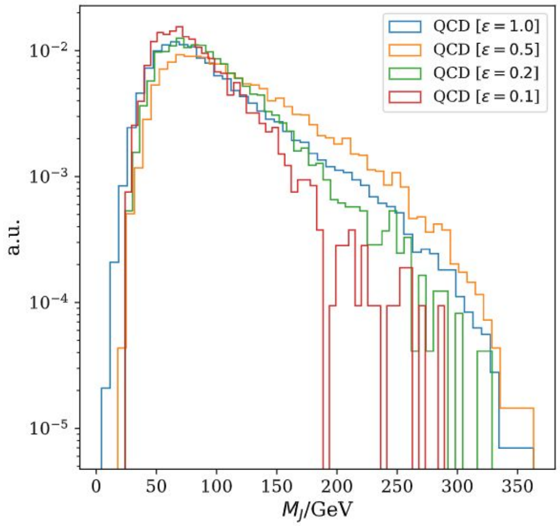
[arXiv:2201.07199]

Applications | Model-Independent New Physics Searches

- Free of mass correlation → readily effective in general resonance searches such as bump-hunt
- Without other auxiliary tasks (and trained on a relatively smaller dataset), the EBM already performs very well

(H→hh→bbbb)

Model	AUC (Top)	AUC (OOD H)
DisCo-VAE ($\kappa = 1000$) (Cheng et al, 2023)	0.593	0.481
KL-OE-VAE (Cheng et al, 2023)	0.744	0.625
EBM ($E(\mathbf{x})$)	0.682 ± 0.004	0.770 ± 0.054



Applications | Classification Augmented with Density Estimation

Hybrid Modelling: joint probability p(x, y)

$$\log p(\mathbf{x}, y) = \log p(\mathbf{x}) + \log p(y|\mathbf{x}) \,.$$

Generative model	Discriminative model
Event simulation	Classifiers

Can be used for semi-supervised learning, OOD detection, etc.

Applications | Classification Augmented with Density Estimation

Hybrid Modelling: joint probability p(x, y)

$$\log p(\mathbf{x}, y) = \log p(\mathbf{x}) + \log p(y|\mathbf{x}) \, .$$

Generative model	Discriminative model
Event simulation	Classifiers

Re-interpret classifiers: see logits as negative energies $g(\mathbf{x})_y = -E(\mathbf{x}, y)$, to re-interpret $p(y|\mathbf{x}) = \mathrm{softmax}(g(\mathbf{x})_y)$

[Grathwohl et al, 2020]

$$p(\mathbf{x}, y) = \frac{\exp(g(\mathbf{x})_y)}{Z}$$

$$p(\mathbf{x}) = \frac{\sum_y \exp(g(\mathbf{x})_y)}{Z}$$

$$p(y|\mathbf{x}) = \frac{\exp(g(\mathbf{x})_y)}{\sum_y \exp(g(\mathbf{x})_y}$$

$$E(\mathbf{x}) = -\log \sum_y \exp(g(\mathbf{x})_y)$$

Applications | Classification Augmented with Density Estimation

Hybrid Modelling: joint probability p(x, y)

$$\log p(\mathbf{x}, y) = \log p(\mathbf{x}) + \kappa \log p(y|\mathbf{x})$$

Optimization: Contrastive divergence with $E(\mathbf{x}) = -\log \sum_y \exp(g(\mathbf{x})_y)$ Cross entropy

Re-interpret classifiers: see logits as negative energies $g(\mathbf{x})_y = -E(\mathbf{x}, y)$, to re-interpret $p(y|\mathbf{x}) = \text{softmax}(g(\mathbf{x})_y)$

[Grathwohl et al, 2020]

$$p(\mathbf{x}, y) = \frac{\exp(g(\mathbf{x})_y)}{Z}$$

$$p(\mathbf{x}) = \frac{\sum_y \exp(g(\mathbf{x})_y)}{Z}$$

→

$$p(y|\mathbf{x}) = \frac{\exp(g(\mathbf{x})_y)}{\sum_y \exp(g(\mathbf{x})_y}$$

$$E(\mathbf{x}) = -\log \sum_y \exp(g(\mathbf{x})_y)$$

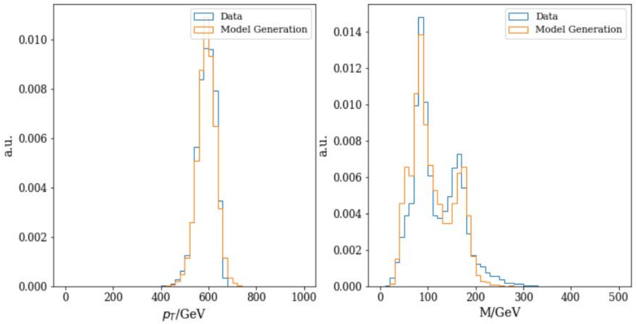
Applications | Classification Augmented with Density Estimation

Hybrid Modelling: joint probability p(x, y)

$$\log p(\mathbf{x}, y) = \log p(\mathbf{x}) + \kappa \log p(y|\mathbf{x})$$

Generative model

Event simulation



Applications | Classification Augmented with Density Estimation

Hybrid Modelling: joint probability p(x, y)

$$\log p(\mathbf{x}, y) = \log p(\mathbf{x}) + \kappa \log p(y|\mathbf{x})$$

Model	Top-1 Accuracy	Top-2 Accuracy
EBM-CLF ($\kappa = 1.0$)	0.848	0.969
ParticleNet	0.871	0.976

Discriminative model

Classifiers

EBM-CLF trained on a smaller dataset is already performing classification tasks on par with dedicated jet classifier.

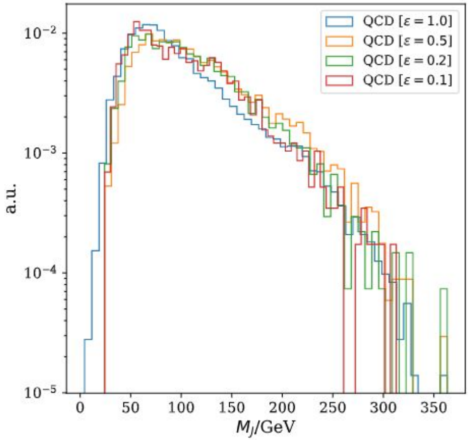
Applications | Classification Augmented with Density Estimation

OOD detection: QCD/Signal

- Now we have a generative model and a discriminative model at the same time
  - p(x) → E(x)
  - Softmax probability p(y=0|x)
  - Logit of the classifier g(x) ~ E(x, y)
- Again E(x) displays mass decorrelation
  - However, anomaly scores from the discriminative part usually remain mass correlated

(H→hh→bbbb)

Model	AUC (Top)	AUC (OOD H)
DisCo-VAE ($\kappa = 1000$) (Cheng et al, 2023)	0.593	0.481
KL-OE-VAE (Cheng et al, 2023)	0.744	0.625
EBM-CLF ($E(\mathbf{x})$)	–	0.817
EBM-CLF ($\mathbf{g}(\mathbf{x})_y$)	0.922	0.877
EBM-CLF ($p(y\|\mathbf{x})$)	0.929	0.870



Summary

- Energy-based probabilistic modelling framework for High Energy Physics events
- Improved training stability (upper-bounded KL-improved training)
- Excellent generation quality with the energy function estimated via a self-attention-based transformer
- Elegantly adapted to different application use-cases:
  - Parameterized event simulation
  - Anomaly detection
  - Classification augmented with density estimation
- Paves for more advanced multi-tasking deep learning models for HEP

Thanks!