Boosting New Physics Sensitivity with Variational Autoencoders

ML4Jets

06.07.2021

Thea Aarrestad, Javier Duarte, Jennifer Ngadiuba, Thong Nguyen,
Maurizio Pierini, Kinga Anna Wozniak

Introduction

Searches at LHC usually

- Given background (standard model, data)
- Assume signal hypothesis

→ Test observations against expectations (MC, data control samples, etc.)

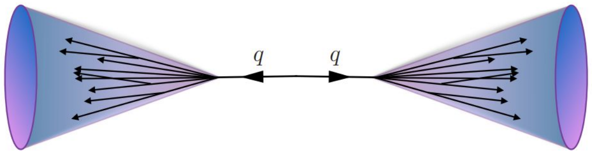
→ Have to know what you are looking for

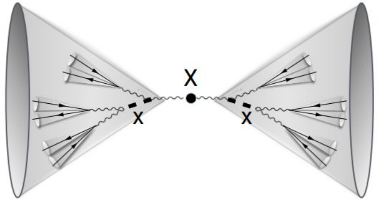
→ Limited by model / theory of choice

Different approach

- Use unsupervised ML to relax assumptions
- No explicit signal model
- Handle to remove background

Introduction





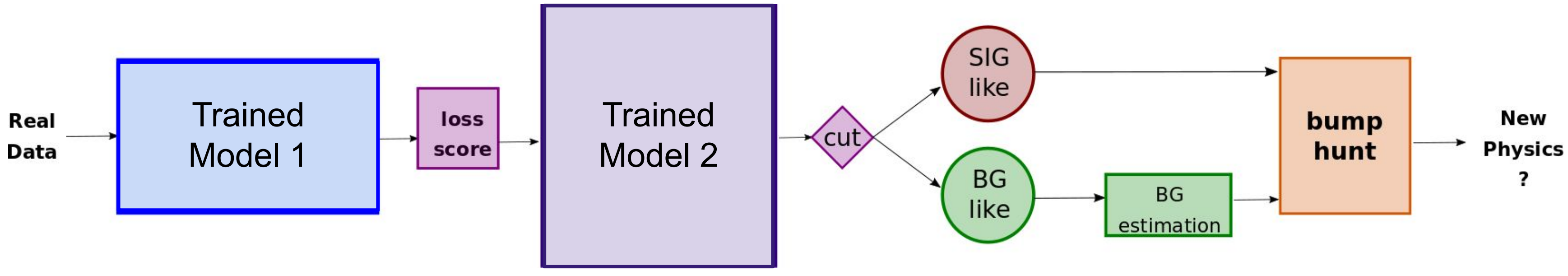
Challenge

- Find mJJ resonant new physics (narrow or broad) in dijet final state events
- Model-agnostic: signal (SIG) (and background (BG))
- Data-driven: train directly on data, no truth-labels available

Idea

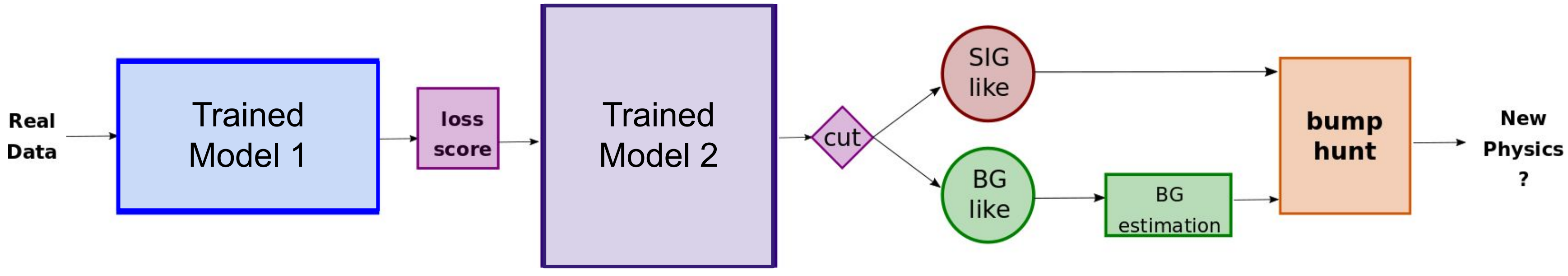
- Selection based on SM veto
- Define this veto using an Variational Autoencoder (VAE) trained on data
- No specific BSM scenario

General Procedure



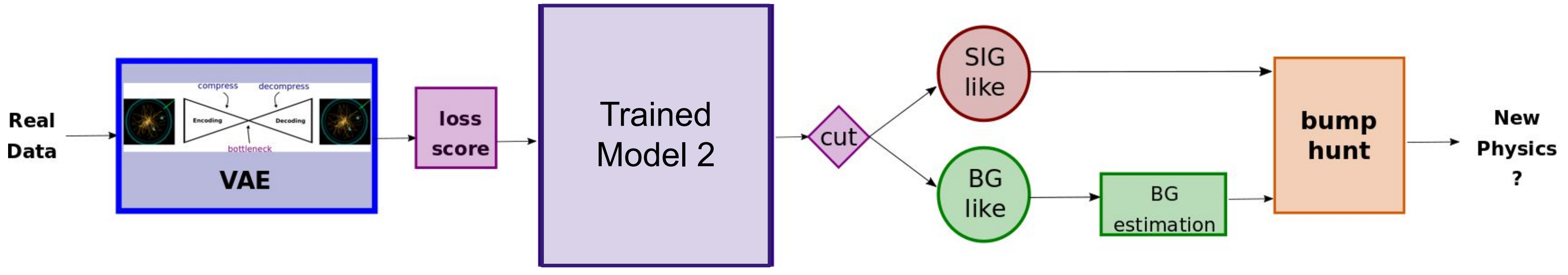
Start from
observations

General Procedure



Train unsupervised model 1 that assesses anomaly (no need for signal MC)

General Procedure



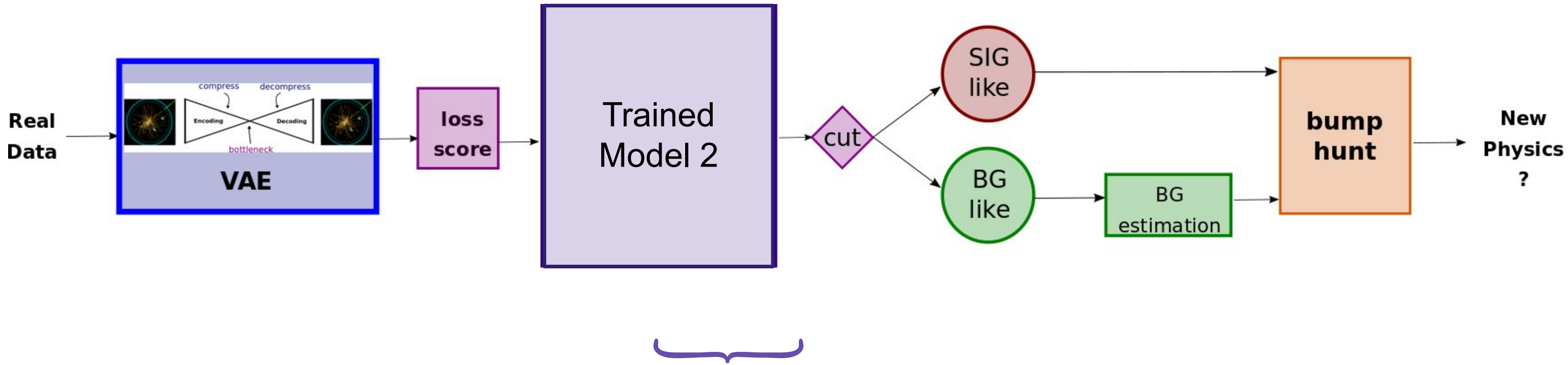
Train unsupervised model 1 that assesses anomaly (no need for signal MC)

↓

Variational Autoencoder

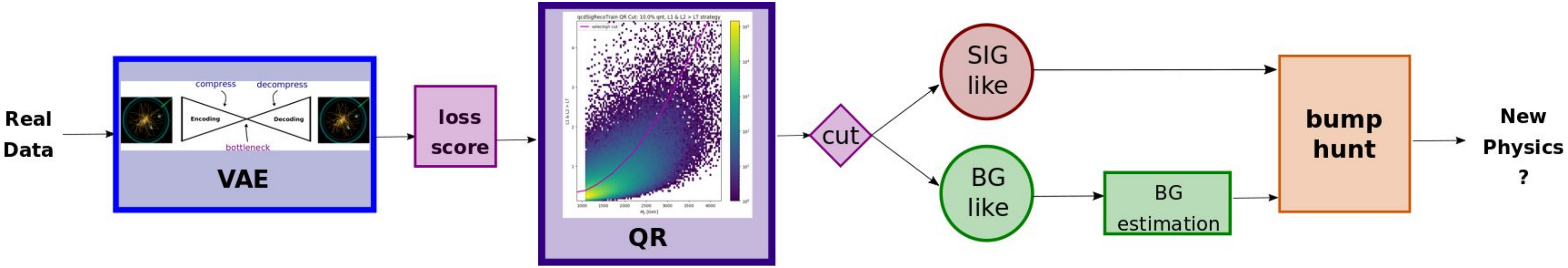
Analysis independent

General Procedure



Train unsupervised model 2 to obtain operating point on model 1 (decorrelated variable of interest)

General Procedure

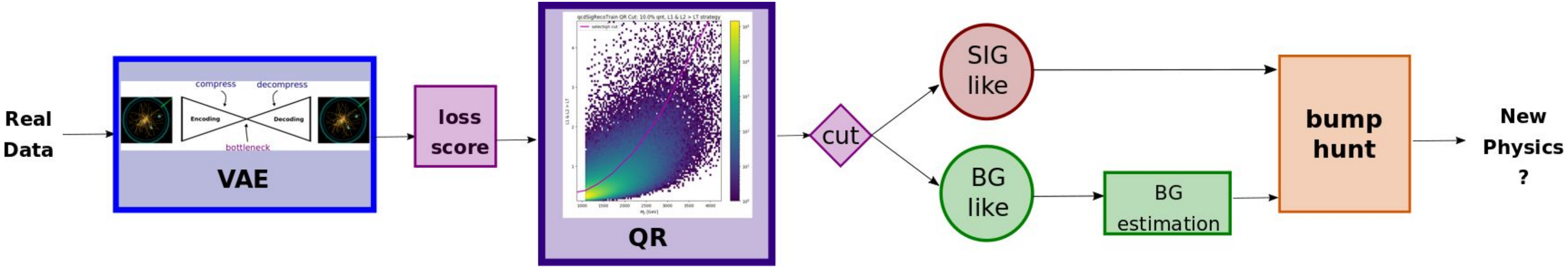


Train unsupervised model 2 to obtain operating point on model 2 (decorrelated variable of interest)

Quantile Regression
Anomaly score - mJJ plane

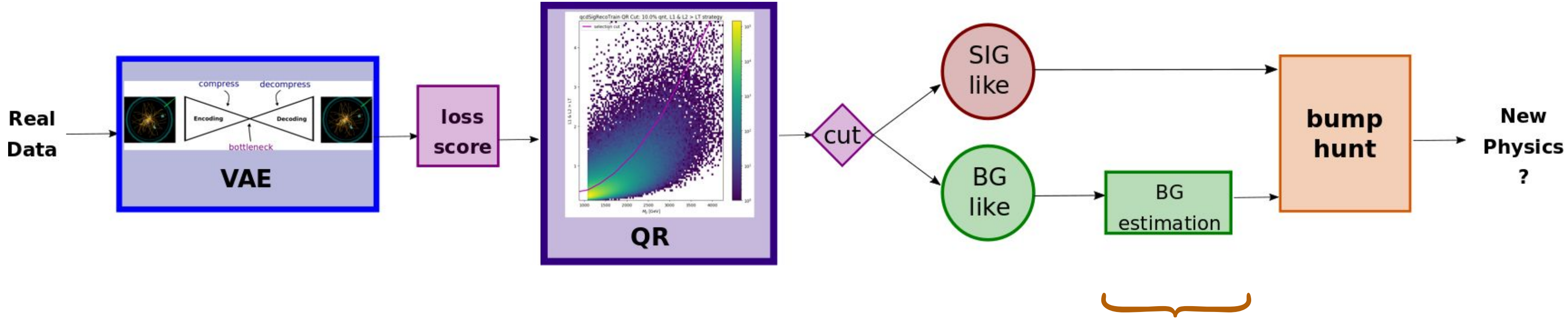
Analysis Specific

General Procedure



Apply cuts and perform hypothesis test

General Procedure

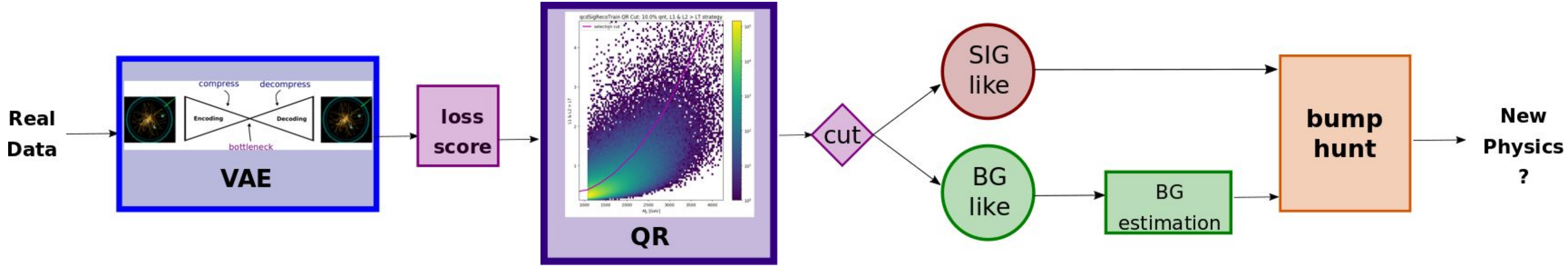


Apply cuts and perform hypothesis test

Dijet-Fit/Bump Hunt On mJJ

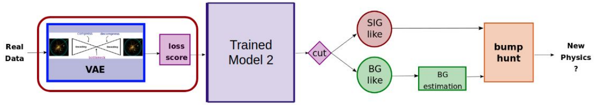
Analysis dependent

General Procedure

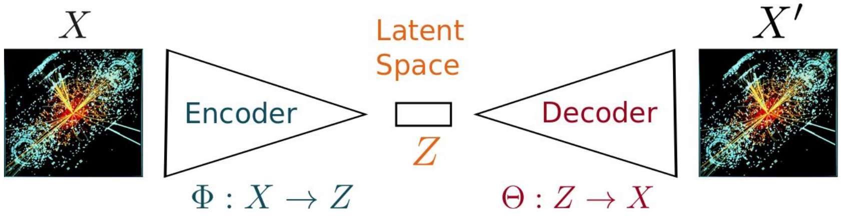


- Signal hypothesis does not enter workflow until very end
- VAE trained once, applicable as selection method
- Generalizable to any analysis with jets in final state

(Variational) Autoencoding (V)AE



Originally designed to compress and decompress inputs, passing through bottleneck (latent space)

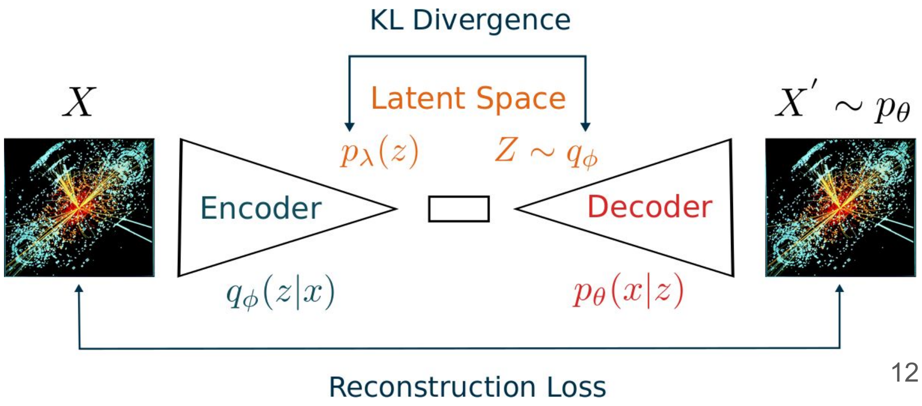


<u>Idea</u>

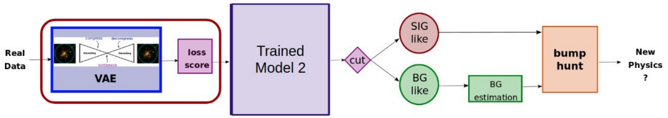
Make AE learn how to compress BG, it will fail when seeing SIG event (reconstruction error)

Use Variational Autoencoder: ([1312.6114](https://arxiv.org/abs/1312.6114))

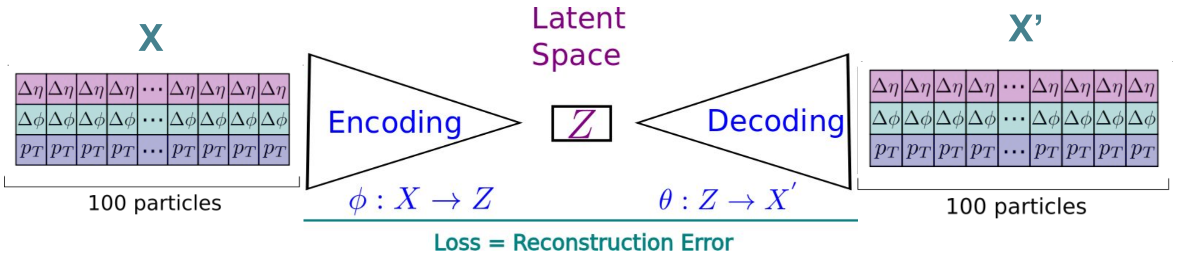
- Learn distribution
- Impose prior on latent space, add divergence to total loss



Variational Autoencoder (VAE)

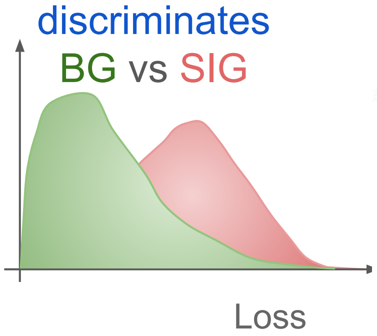


Input: Particle list (Δη, Δφ, pt), Jet1 & Jet2



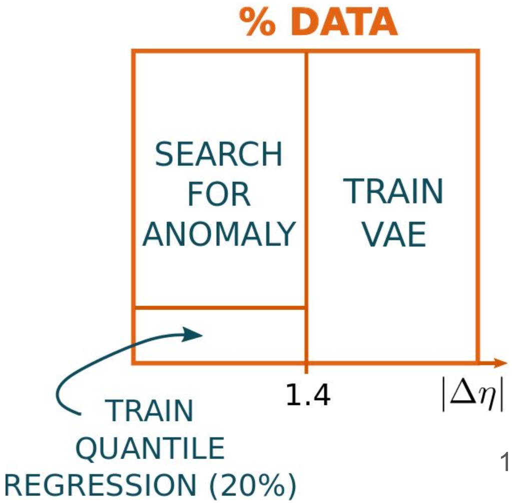
Training:
Define data sideband (dominated by BG) as |Δη| > 1.4

Output: metric for anomaly
⇒ Chamfer reconstruction + Kullback-Leibler divergence

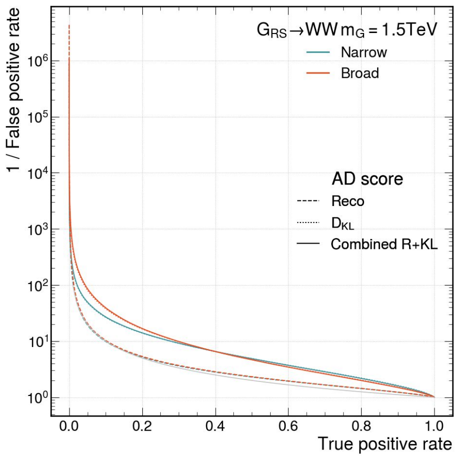


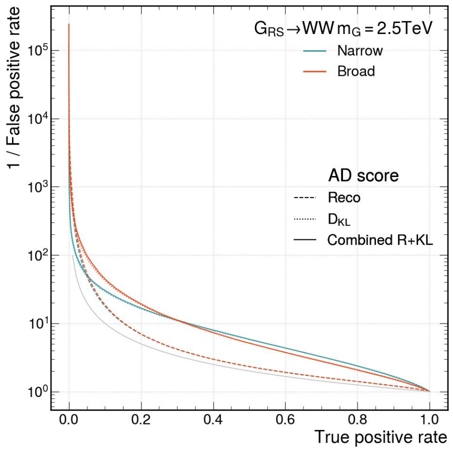
$$L_R = \sum_{i \in input} \min_j ((x^{(i)} - x^{(j)})^2) + \sum_{j \in output} \min_i ((x^{(j)} - x^{(i)})^2)$$

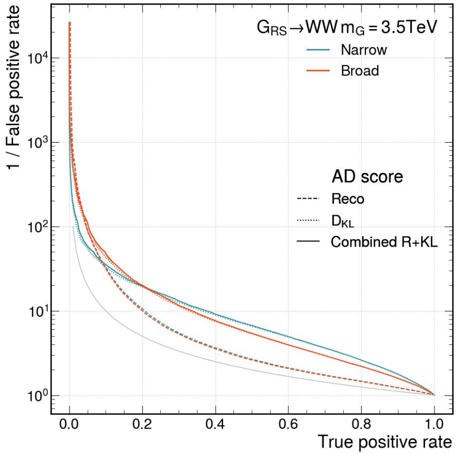
$$D_{KL}(q||p) = \sum_x p(x) \log \frac{q(x)}{p(x)}$$

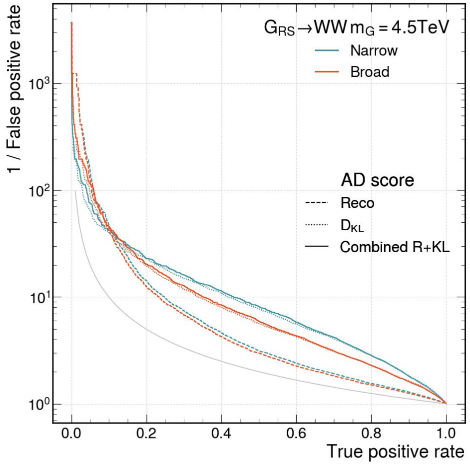


Anomaly Score QCD vs G_{RS} at m_G in {1.5, 2.5, 3.5, 4.5}TeV





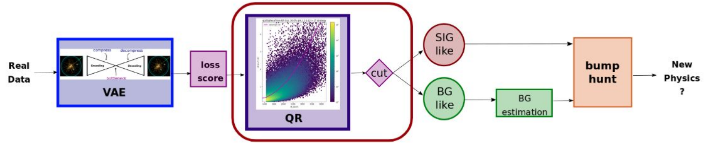


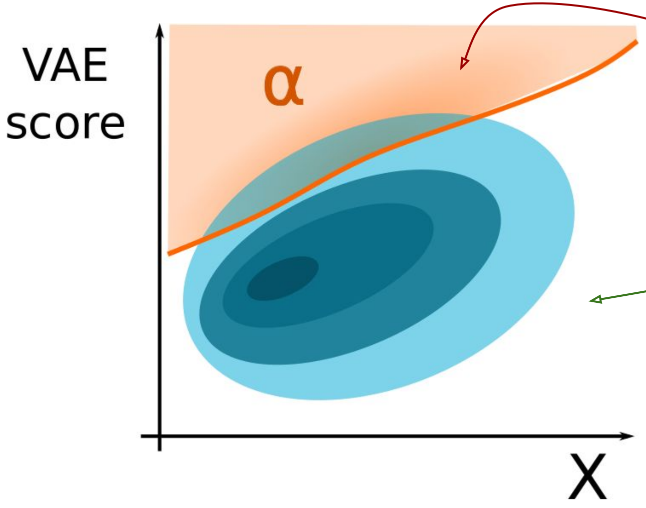


Reasonable for broad and narrow Resonances

Combined = best compromise

Quantile Regression





Train Deep NN to split data into

SIG like

BG like

On the plane

VAE score - Variable of interest X

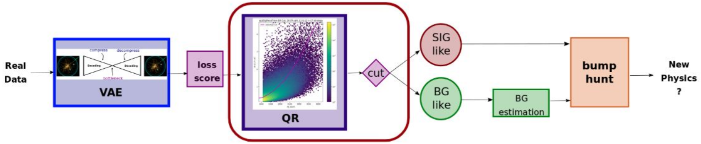
=> Decorrelation

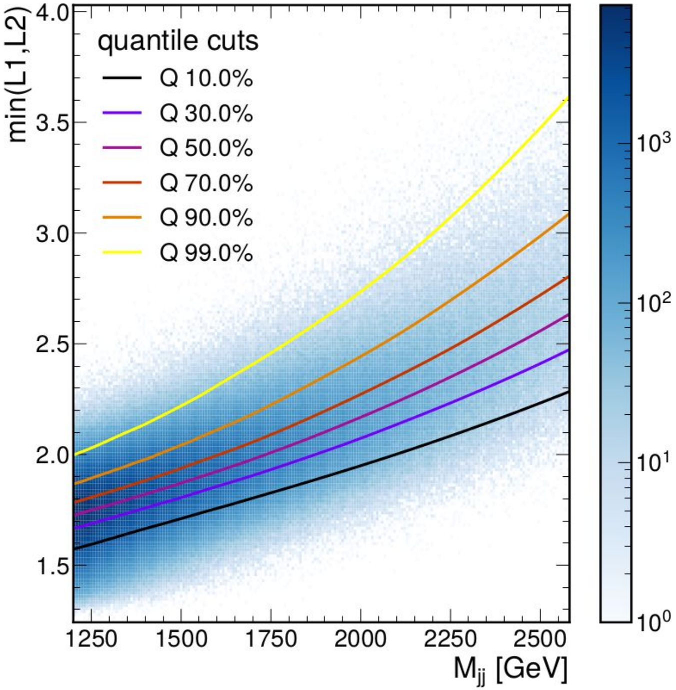
=> s.t. BG events follow same distribution in the two regions

For a given **quantile α efficiency cut**

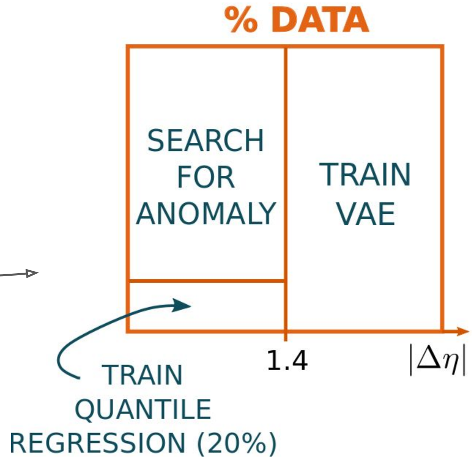
QR Neural Network:
- 5 Dense Layers
- Nonlinear Activations
- Input: X=mJJ, Output: cut

Quantile Regression





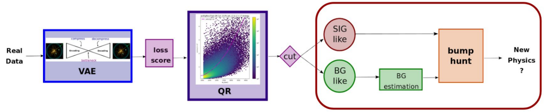
Train the QR on signal region



Make multiple quantile cuts

=> Orthogonal categories

Bottom Quantile cut: Most BG like

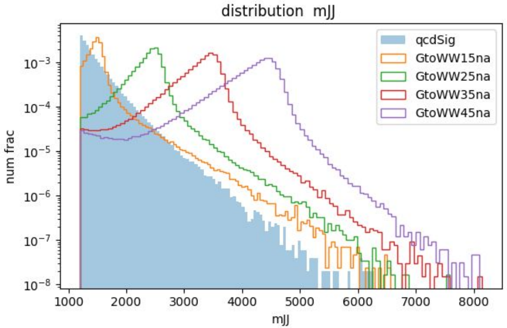


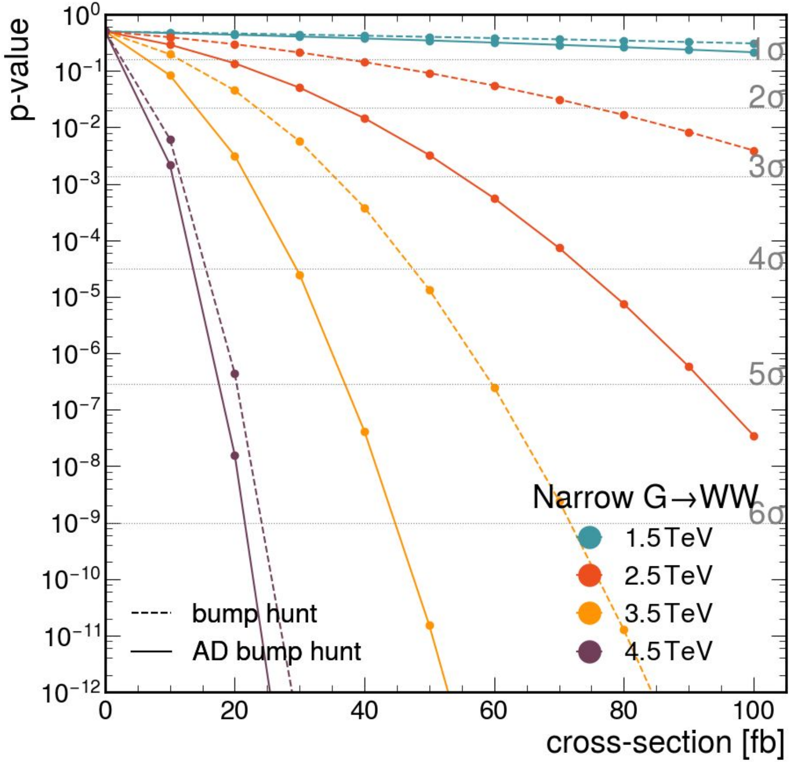
Dijet Fit ("Bump Hunt")

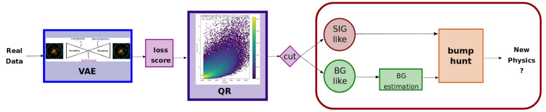
- Perform typical LHC bump hunt
- Fit S + B simultaneously
- Extract signal strength μ

If test statistics falls beyond threshold → claim discovery

Narrow resonance

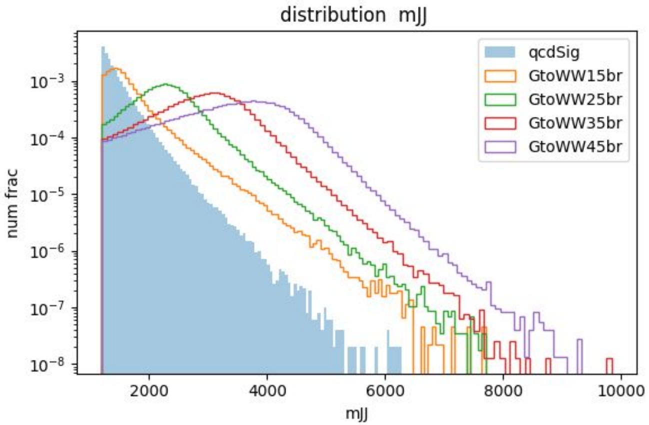


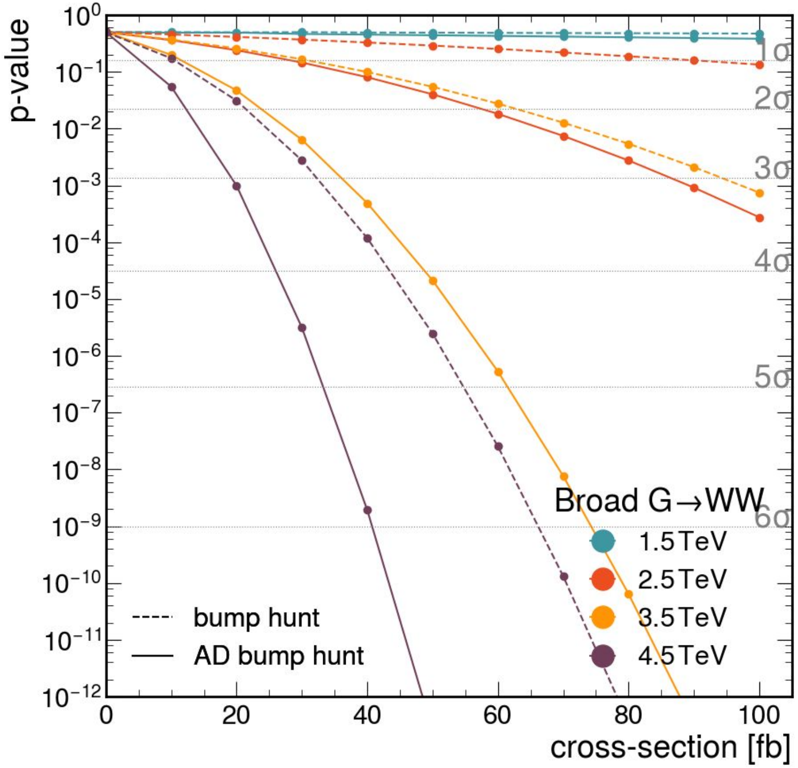




Dijet Fit ("Bump Hunt")

Broad resonance mJJ spectrum





Conclusion

- Presented model-agnostic anomaly detection procedure to boost LHC searches
- We consider VAE but method applies to any unsupervised AD algorithm
- BG estimate not disrupted by using QR to define selection cuts
- Models trainable directly on data (no extra MC dependence)

Original Contribution:

- Showed application of ADA boost to state-state-of-the-art bump-hunt analysis and its potential to enhance sensitivity
- Procedure applicable to **any** BSM search with jets in the final state provided a data sideband

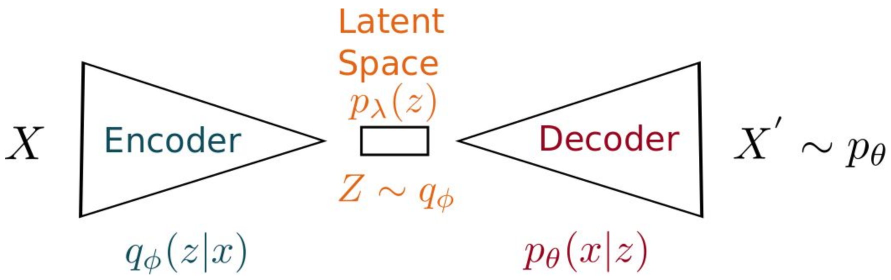
Thank you!

Backup

Variational Autoencoding Theory

Two parts:

- Inference model (encoder)
- Generative model (decoder)



Want to maximize likelihood of data and latent variables:

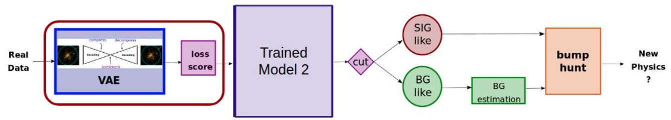
$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

$$log p_\theta(D) = \sum_{x \in D} log\, p_\theta(x)$$

But true data distribution p(x) is intractable →approximate with variational inference

$$q_\phi(z|x) \qquad p_\theta(x|z)$$

Variational Autoencoding Theory



Training Loss = Reconstruction Loss + β * KL-Divergence

→ No assumption on anomaly model

$$p_\theta(x|z) = \mathcal{N}(\mu(z), I):$$

$$-log\, p(X|\mu(z)) = -\sum_i log(e^{-\frac{(x_i-\mu_i(z))^2}{2}})$$

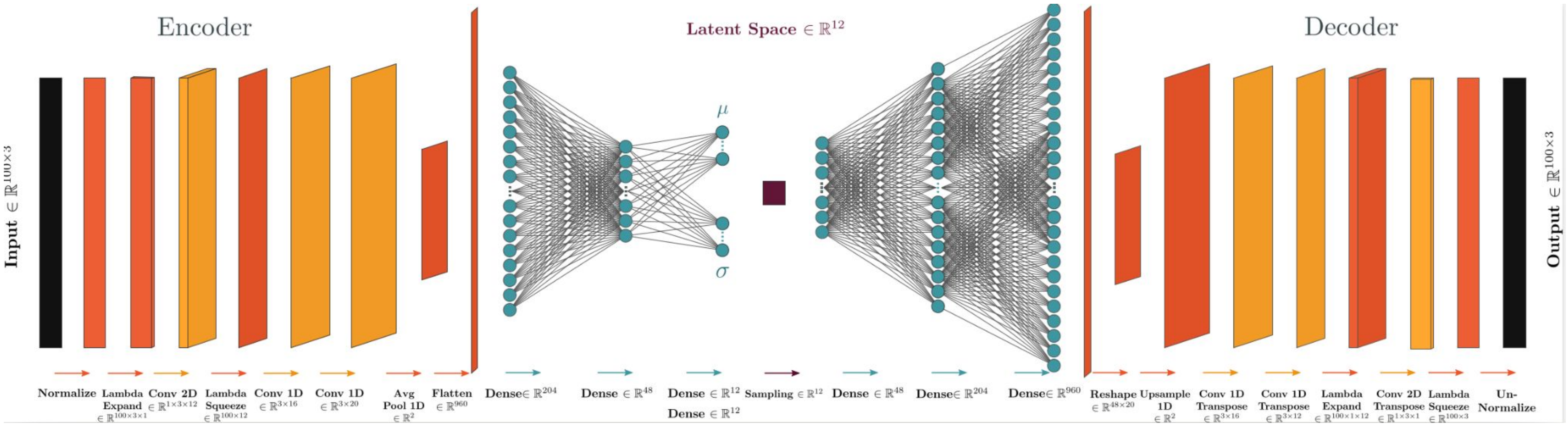
$$= \sum (x_i - \mu_i(z))^2$$

where x_i is pixel i of jet image X

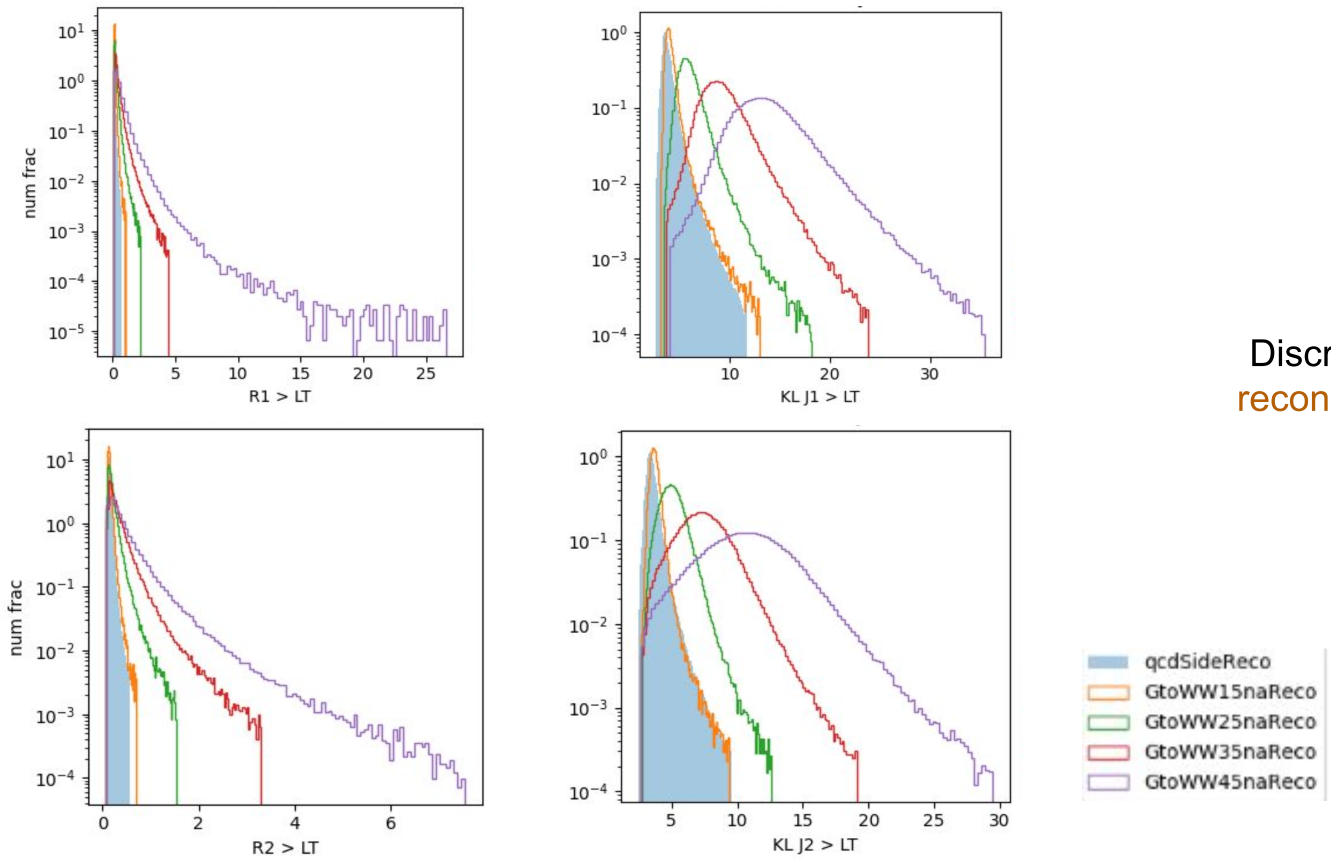
KL-Divergence: Relative Entropy

$$D_{KL}(q_\phi(z|x)||p_\lambda(z)) = \int p(z) \log \frac{q(z|x)}{p(z)} dz$$

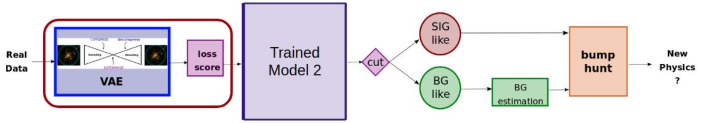
VAE Architecture



Anomaly Score QCD vs G_{RS} at m_G in {1.5, 2.5, 3.5, 4.5}TeV



Discriminative power in reconstruction and latent Space
→VAE



VAE: Methodology

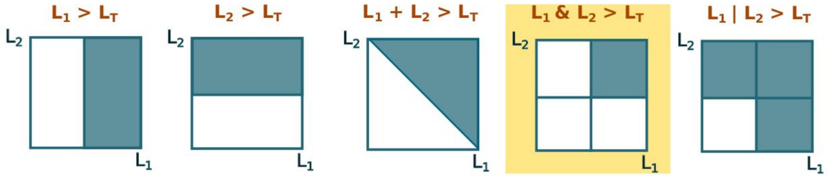
How do you formulate **anomaly metric**?

Pairwise Minimum Distance (Chamfer): reconstruction
+ Kullback-Leibler: latent divergence

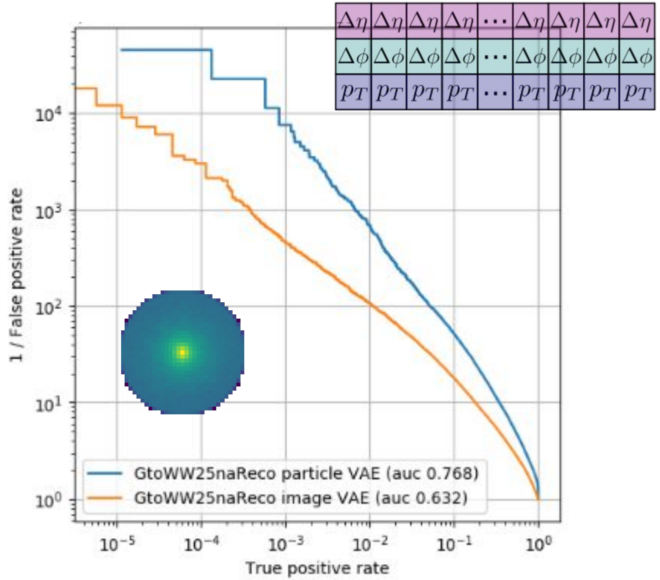
$$L_R = \sum_{i \in input} \min_j((x^{(i)} - x^{(j)})^2) + \sum_{j \in output} \min_i((x^{(j)} - x^{(i)})^2)$$

$$D_{KL}(q||p) = \sum_x p(x) \log \frac{q(x)}{p(x)}$$

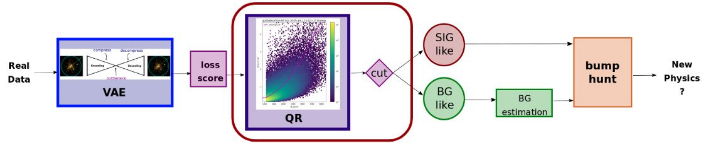
Strategy = Combine losses L1 of Jet1 & L2 of Jet2

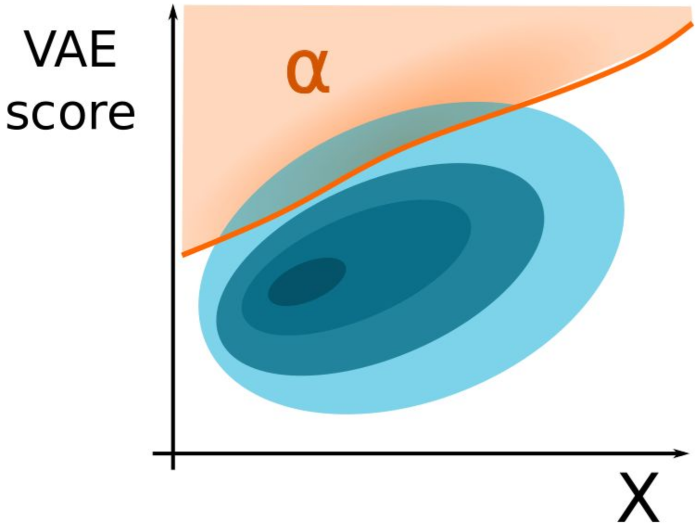


Input Representation
Permutation invariant loss:
Dijet Image + MSE worse than particle list + Chamfer



Quantile Regression



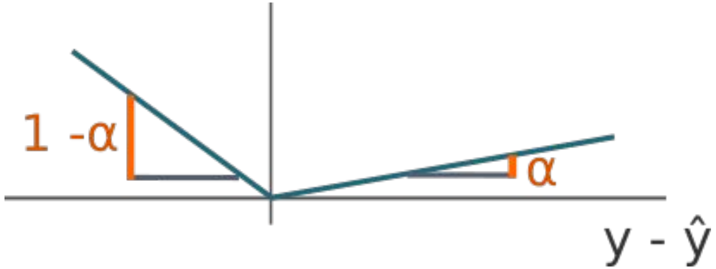


QR Neural Network:

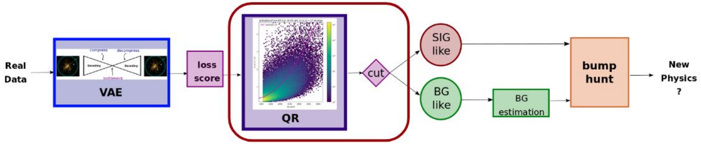
- 5 Dense Layers
- Nonlinear Activations
- Input: X=mJJ, Output: cut

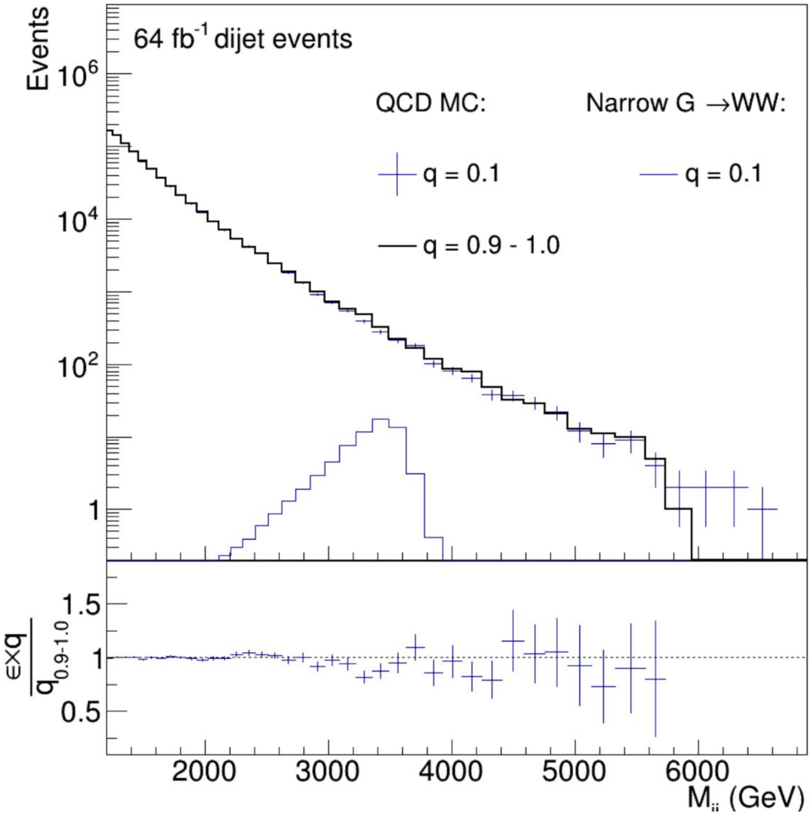
Loss:

$$L(y,\hat{y}) = \begin{cases} \alpha \cdot (y-\hat{y}), & \text{if} \quad y \geq \hat{y} \\ (\alpha - 1) \cdot (y - \hat{y}), & \text{otherwise} \end{cases}$$



Quantile Regression

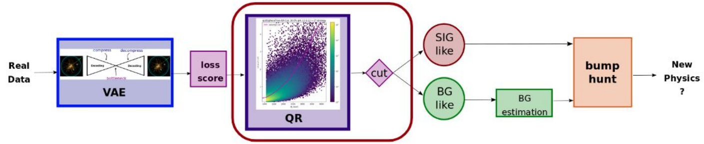




Decorrelated QR gives

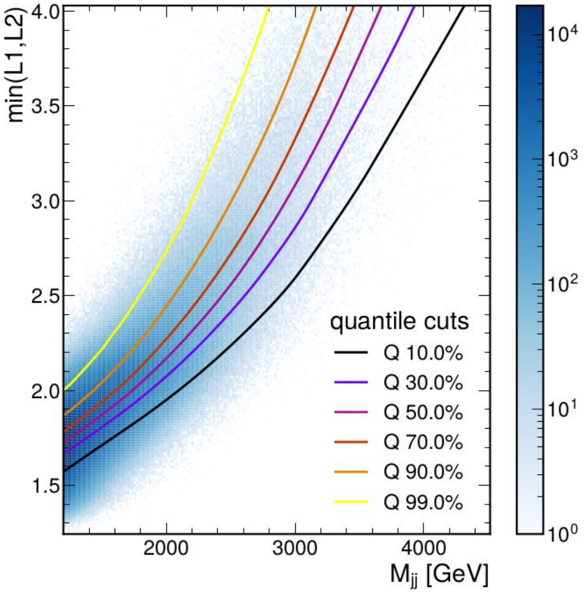
Equivalent spectrum shapes of variable of interest, e.g. mJJ for Background = smooth falling shape for all quantiles

Quantile Regression

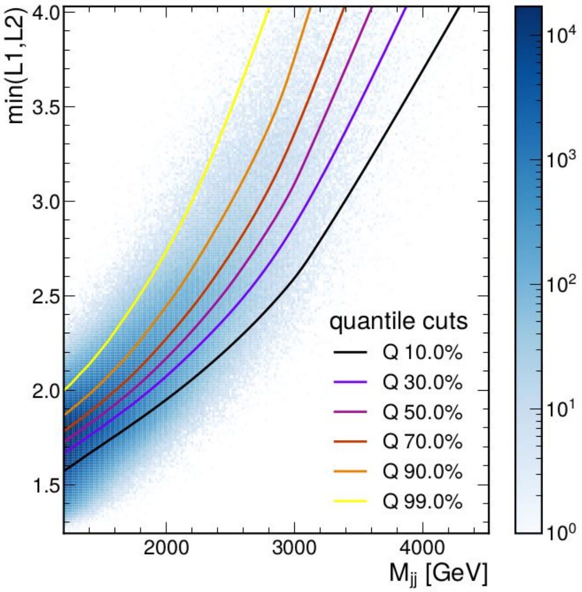


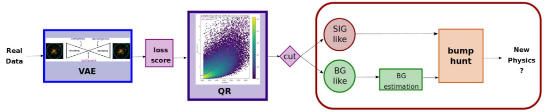
QR trained on dataset with injected signal → no difference → considered stable

Pure QCD training



G_RS injected at 100 fb





Dijet Fit (“Bump Hunt”)

- Get BG shape from standard model and signal shape from MC
- Split the sample in N bins with same shape (by construction) and run a simultaneous bump hunt
- then compare to inclusive bump hunt
- Inject signal, retrain and scan the xsec
- Fit SIG + BG simultaneously with likelihood

$$\mathcal{L}(data|\mu,\theta) = \text{Poisson}(data|\mu \cdot s(\theta) + b(\theta)) \cdot p(\tilde{\theta}|\theta).$$

- If test statistics falls beyond threshold -> claim discovery