CATHODE

Part 1: introducing a new model-agnostic search strategy for resonant new physics at the LHC

Anna Hallin
Rutgers University
ML4Jets, 2021-07-06

Don’t miss! CATHODE Part 2, today 14:40, M. Sommerhalder

CATHODE

Part 1: introducing a new model-agnostic search strategy for resonant new physics at the LHC

Anna Hallin
Rutgers University
ML4Jets, 2021-07-06

Don't miss! CATHODE Part 2, today 14:40, M. Sommerhalder

Paper out shortly:

Classifying Anomalies THrough Outer Density Estimation (CATHODE)

AH, J. Isaacson, G. Kasieczka, C. Krause, T. Lösche, B. Nachman, M. Schlaffer, D. Shih, M. Sommerhalder
arXiv 2107.xxxxx

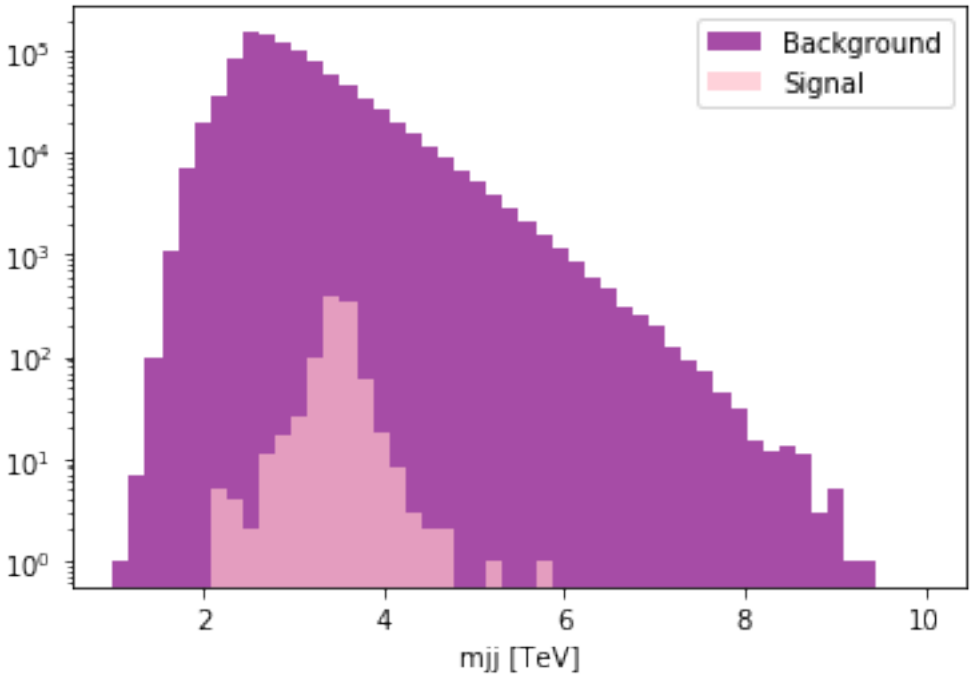
Introduction

CATHODE is a new **bump-hunt enhancing** method for **model-agnostic anomaly searches**: want to be able to look for new physics without necessarily knowing what we are looking for.

The general idea behind a bump hunt is to find a way to enhance the signal bump to make it visible over the smooth background; for this we need additional handles.

Assume we have a resonant variable m_{JJ}, and some other discriminating features $\mathbf{x}$: place cuts on these features to reject background while retaining signal.

How do we know where to place these cuts, when all we have is the data, and don't know the individual distributions of the signal and background?



Probability densities and likelihood ratios

The Neyman-Pearson lemma:

*The **optimal binary classifier** between two simple hypotheses H1 and H2 is the **likelihood ratio** R(x) = p(x|H1)/p(x|H2)*

In our case, the likelihood ratio we are interested in is R(x) = p(x|data)/p(x|SM)

If R(x) = 1, the data looks like the Standard Model.

We can either try to **find the optimal classifier**, which will give us the likelihood ratio, or we can **try to learn the probability densities** directly and then take their ratio.

Probability densities and background

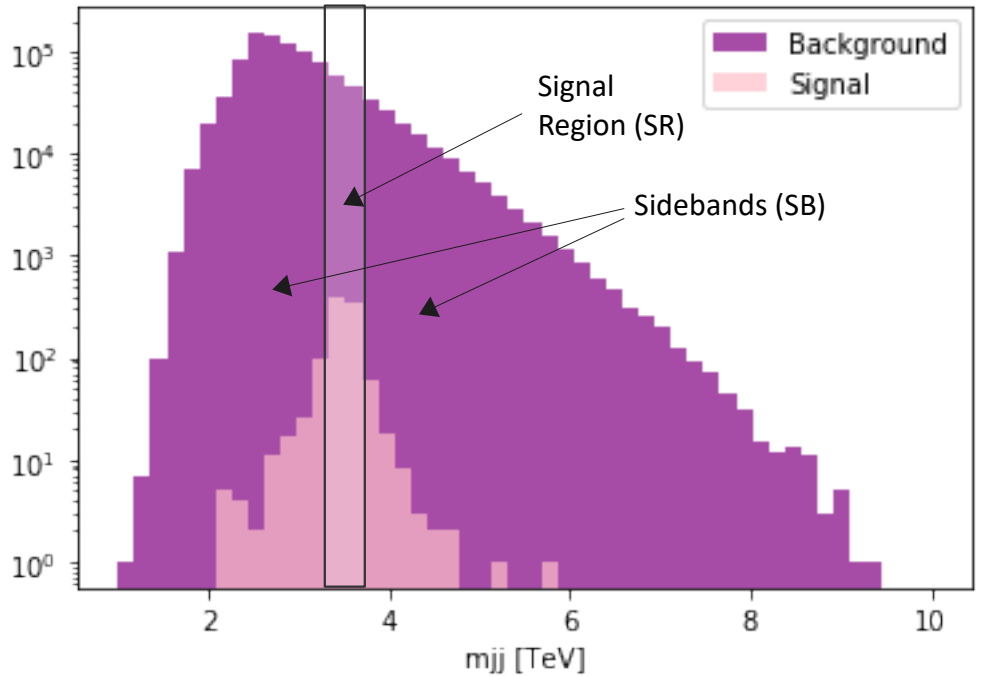
Assume we have a resonant variable m_{JJ}, and some other features $\mathbf{x}$.

$$p_{data}(m_{JJ}, \mathbf{x}) = \varepsilon\, p_{signal}(m_{JJ}, \mathbf{x}) + (1-\varepsilon) p_{background}(m_{JJ}, \mathbf{x})$$

How to find $p_{bg}(m_{JJ}, \mathbf{x})$ for a localized signal?

3 different approaches:

- Find $p_{bg}(m_{JJ}, \mathbf{x})$ via **simulation** – but does this accurately represent the background in the data?



Probability densities and background

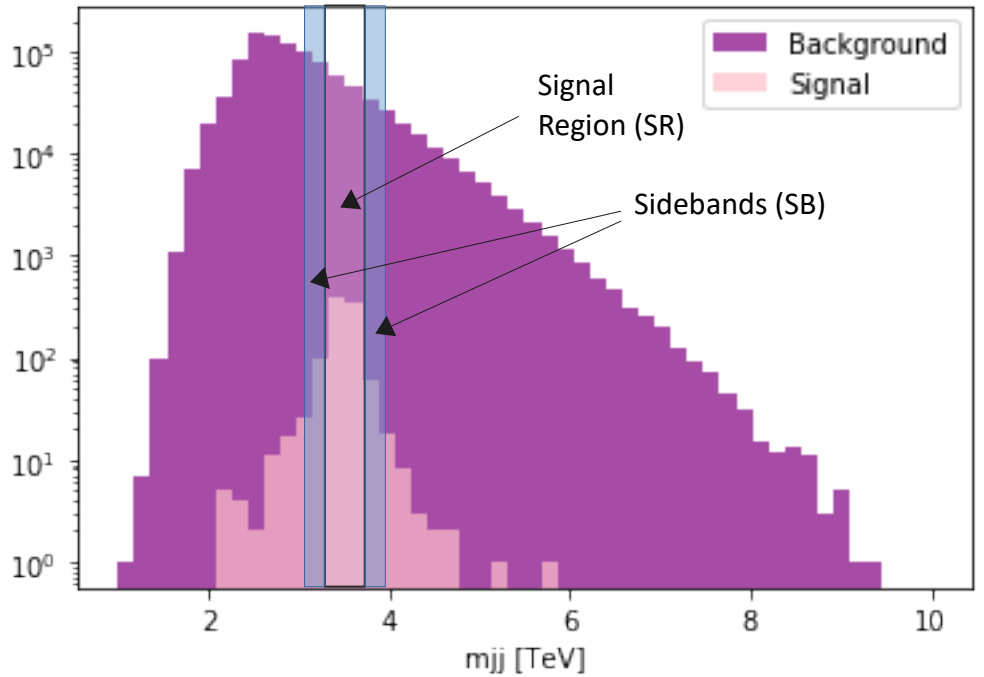
Assume we have a resonant variable m_{JJ}, and some other features $\mathbf{x}$.

$$p_{data}(m_{JJ}, \mathbf{x}) = \varepsilon\, p_{signal}(m_{JJ}, \mathbf{x}) + (1-\varepsilon) p_{background}(m_{JJ}, \mathbf{x})$$

How to find $p_{bg}(m_{JJ}, \mathbf{x})$ for a localized signal?

3 different approaches:

- Find $p_{bg}(m_{JJ}, \mathbf{x})$ via simulation – good enough?
- Assume $p_{bg,SR}(m_{JJ}, \mathbf{x}) = p_{data,SB}(m_{JJ}, \mathbf{x})$ and train a classifier to distinguish between **data in the (narrow) sidebands** and the signal region (CWoLa) – not robust against correlations between m_{JJ} and $\mathbf{x}$.



CWoLA: E. M. Metodiev, B. Nachman, J. Thaler, 1708.02949; J.H. Collins, K. Howe, B. Nachman, 1805.02664 and 1902.02634

Probability densities and background

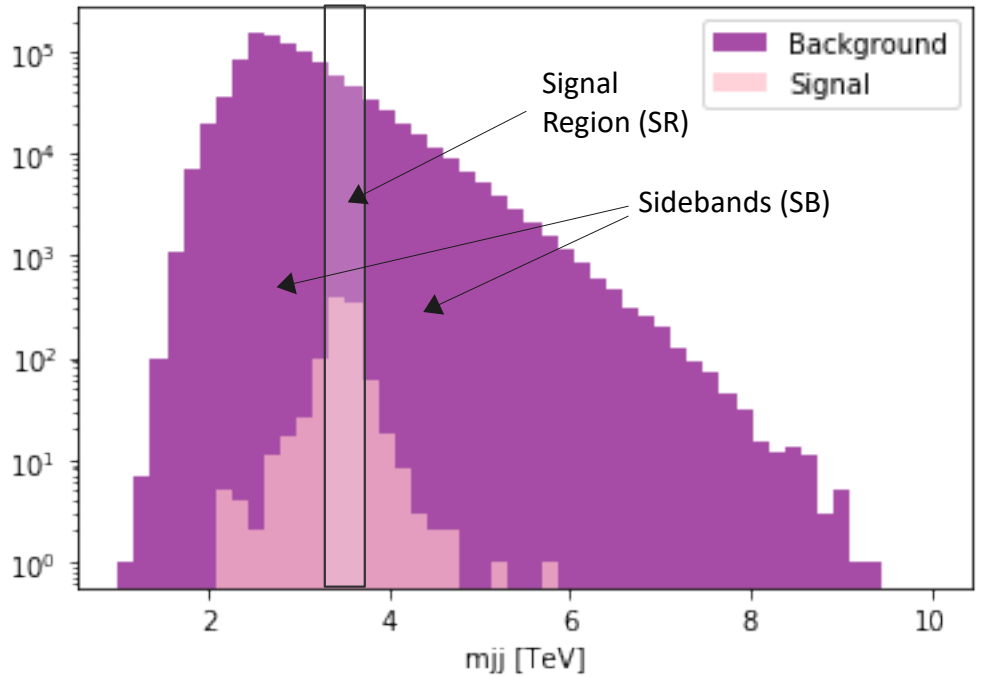
Assume we have a resonant variable m_{JJ}, and some other features $\mathbf{x}$.

$$p_{data}(m_{JJ}, \mathbf{x}) = \varepsilon\, p_{signal}(m_{JJ}, \mathbf{x}) + (1-\varepsilon) p_{background}(m_{JJ}, \mathbf{x})$$

How to find $p_{bg}(m_{JJ}, \mathbf{x})$ for a localized signal?

3 different approaches:

- Find $p_{bg}(m_{JJ}, \mathbf{x})$ via simulation – good enough?
- Assume $p_{bg,SR}(m_{JJ}, \mathbf{x}) = p_{data,SB}(m_{JJ}, \mathbf{x})$ and train a classifier to distinguish between data in the (narrow) sidebands and the signal region (CWoLa) – correlations?
- Train a **conditional density estimator** on $p_{data,SB}(\mathbf{x}|m_{JJ})$ and interpolate into the signal region. Separately train another density estimator on $p_{data,SR}(\mathbf{x}|m_{JJ})$ and **calculate the likelihood ratio** (ANODE) – a much more difficult task than training a classifier, but more robust to correlations.



ANODE: B. Nachman, D. Shih 2001.04990

The idea behind CATHODE

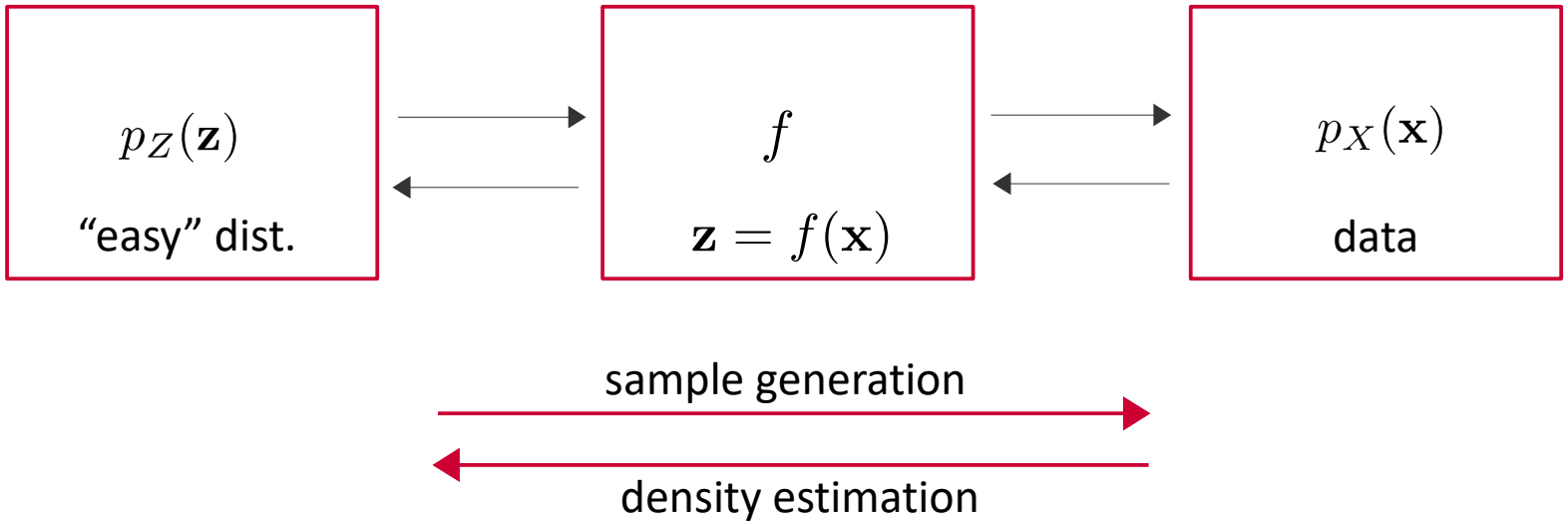
Take the best parts of CWoLa and ANODE:

- Train a **conditional density estimator** on data in the **sidebands** and **interpolate into the signal region**. This protects against collapse due to correlations (see talk by M. Sommerhalder, CATHODE Part 2).
- **Generate samples** from the learned probability density, in the signal region. This is the background model.
- Train a **classifier** to **distinguish between data and samples** in the signal region. The combination of one density estimator and one classifier is easier than having to do two density estimations.

Quick intro to normalizing flows

The density estimation is performed using a Masked Autoregressive Flow (MAF), which is a type of normalizing flow.

Let f be a **bijective** map from a latent space with distribution $p_Z(\mathbf{z})$ to the feature space with distribution $p_X(\mathbf{x})$, such that $\mathbf{z} = f(\mathbf{x})$.



By the **change of variables** formula for random variables, $p_X(\mathbf{x}) = p_Z(\mathbf{z}) \left| \det \frac{f^{-1}(\mathbf{z})}{\partial \mathbf{z}} \right|^{-1}$

L. Dihn *et al* 1410.8516 | D. Jimenez Rezende *et al* 1505.05770 | M. Germain *et al* 1502.03509 | G. Papamakarios *et al* 1705.07057

Density estimation: Expressivity and Jacobian

A **chain of bijective maps** is also bijective: $f = f_1 \circ f_2 \circ ... \circ f_n$

In this way, we can use functions f_i that are easily invertible, while still obtaining **expressivity**.

Density estimation: Expressivity and Jacobian

A **chain of bijective maps** is also bijective: $f = f_1 \circ f_2 \circ ... \circ f_n$

In this way, we can use functions f_i that are easily invertible, while still obtaining **expressivity**.

In general, a number of $\mathcal{O}(d^3)$ operations are needed to evaluate a d-dimensional Jacobian.

This is made tractable by turning it into a **triangular matrix**, which only requires $\mathcal{O}(d)$ operations for evaluation.

$$
\begin{aligned}
x_1 &= f_1(z_1) \\
x_2 &= f_2(z_1, z_2) \\
&... \\
x_d &= f_d(z_1, ..., z_d)
\end{aligned}
\quad \Longrightarrow \quad \text{Triangular matrix}
$$

Density estimation: Masked Autoregressive Flow

f is not a **neural network**, since that wouldn't be invertible, but its **parameters** (eg. $\mu, \alpha, ...$) are. The parameters will be functions of z, such that $f(z) = f(\mu(z), \alpha(z), ...)$.

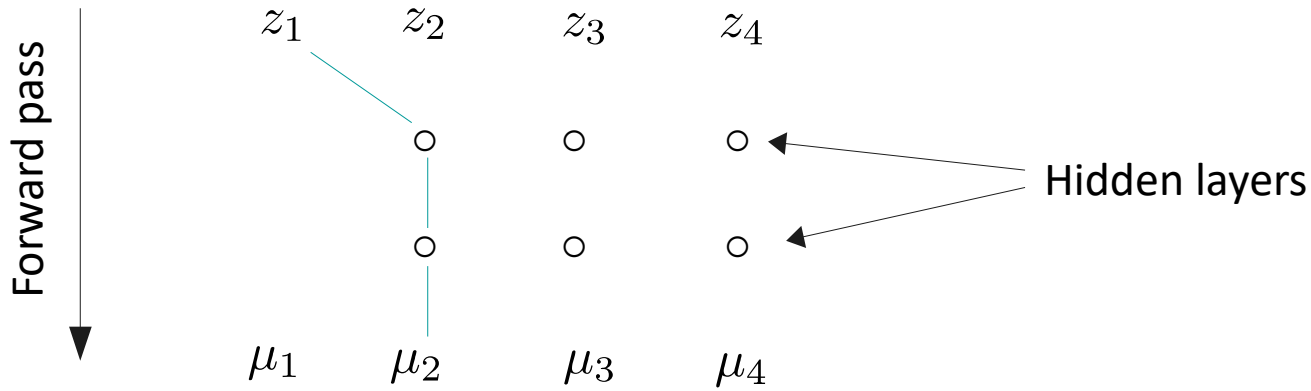
Density estimation: Masked Autoregressive Flow

f is not a **neural network**, since that wouldn't be invertible, but its **parameters** (eg.$\mu, \alpha, \ldots$) are. The parameters will be functions of z, such that $f(z) = f(\mu(z), \alpha(z), ...)$.

Use binary masks on the weights to ensure that **each output is conditioned only on the previous** outputs:

$\mu_i(z_1, ..., z_{i-1})$

This ensures the **autoregressive property** (→triangular Jacobian), and goes by the name **Masked Autoregressive Flow** (MAF).



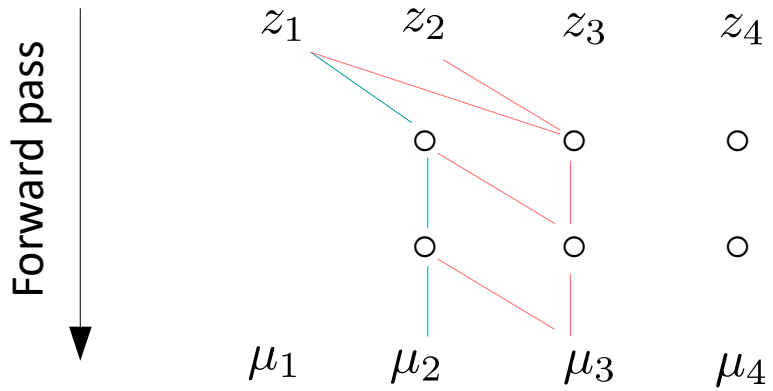
Density estimation: Masked Autoregressive Flow

f is not a **neural network**, since that wouldn't be invertible, but its **parameters** (eg.$\mu, \alpha, \ldots$) are. The parameters will be functions of z, such that $f(z) = f(\mu(z), \alpha(z), ...)$.

Use binary masks on the weights to ensure that **each output is conditioned only on the previous** outputs:

$\mu_i(z_1, ..., z_{i-1})$

This ensures the **autoregressive property** (→triangular Jacobian), and goes by the name **Masked Autoregressive Flow** (MAF).



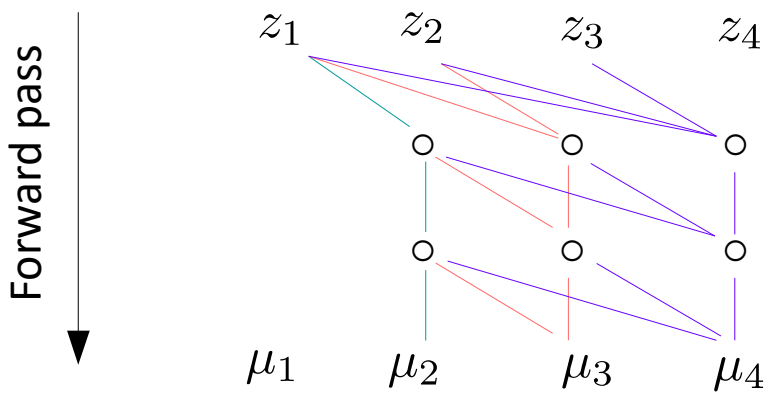
Density estimation: Masked Autoregressive Flow

f is not a **neural network**, since that wouldn't be invertible, but its **parameters** (eg.$\mu, \alpha, \ldots$) are. The parameters will be functions of z, such that $f(z) = f(\mu(z), \alpha(z), ...)$.

Use binary masks on the weights to ensure that **each output is conditioned only on the previous** outputs:

$\mu_i(z_1, ..., z_{i-1})$

This ensures the **autoregressive property** (→triangular Jacobian), and goes by the name **Masked Autoregressive Flow** (MAF).



For a conditional density estimation, add a feature z_5 which is not masked – all μ_i can depend on it.

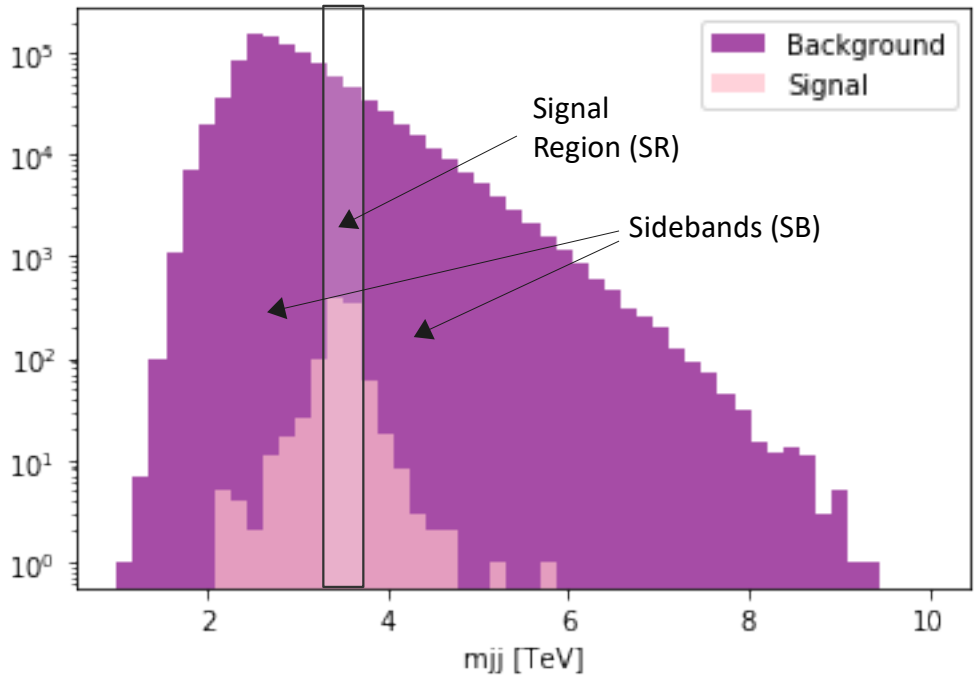
Note that we get all of this in a **single forward pass** through the network. The MAF is very fast at density estimation. However, if we want to go the other way and produce samples, we will have to loop sequentially over all x_i, which is slow*.

*For sampling, one can use the Inverse Autoregressive Flow (IAF), which is fast for sampling but slow for density estimation.

Dataset

From the LHC Olympics R&D dataset, we use:

- 1,000,000 QCD dijet events
- 1,000 $W' \to X(\to qq)Y(\to qq)$ events
- $m_{W'} = 3.5\ \mathrm{TeV}, m_X = 500\ \mathrm{GeV}, m_Y = 100\ \mathrm{GeV}$
- In signal region, $3.3\ \mathrm{TeV} < m_{JJ} < 3.7\ \mathrm{TeV}$:
  - 121,352 background events
  - 772 signal events
- Initial $S/B = 6 \times 10^{-3}$, $S/\sqrt{B} = 2.2$



LHCO: G. Kasieczka *et al*, 2101.08320

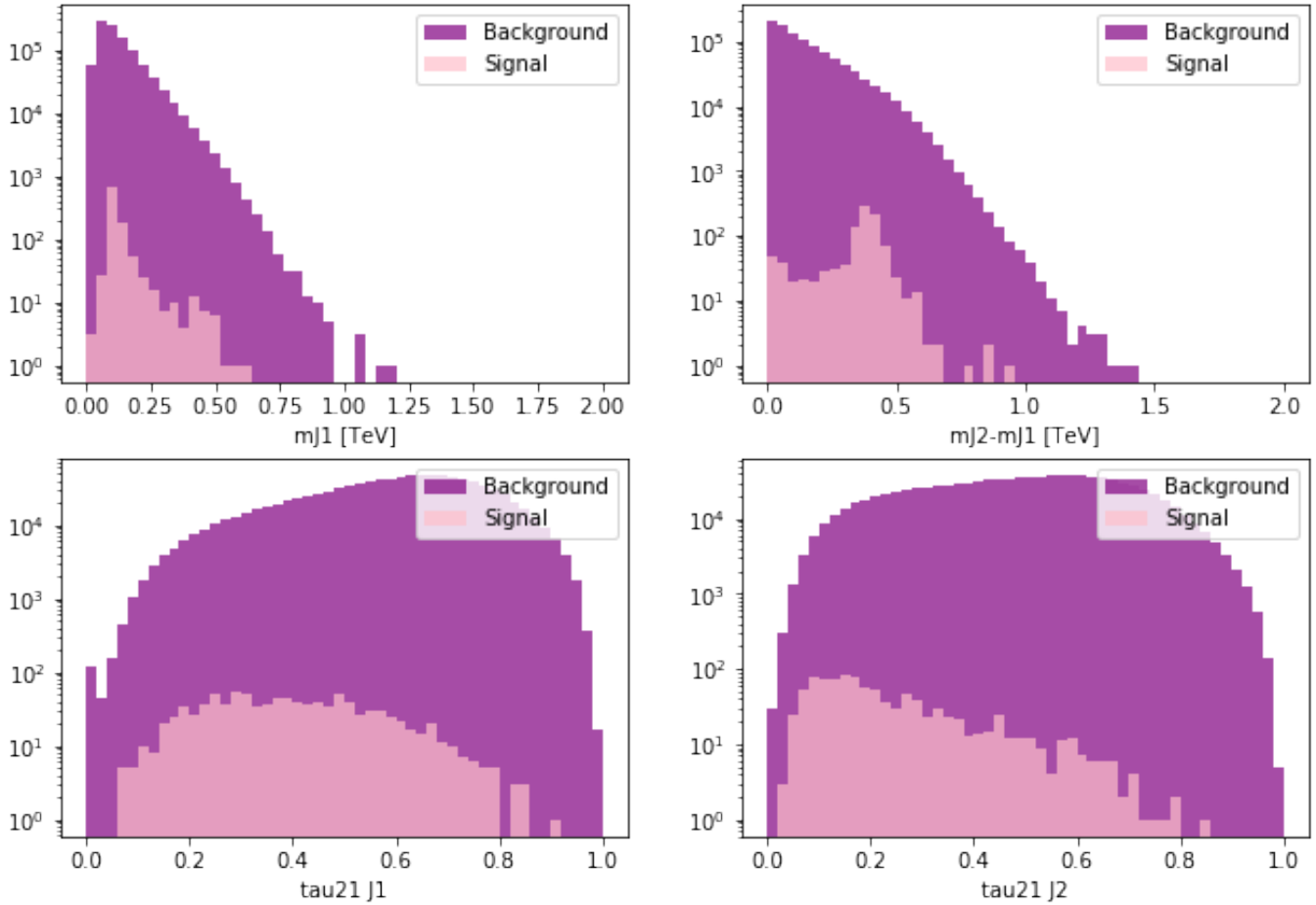
Dataset: features

Conditional feature:

- m_{JJ} – the total invariant mass of the two jets

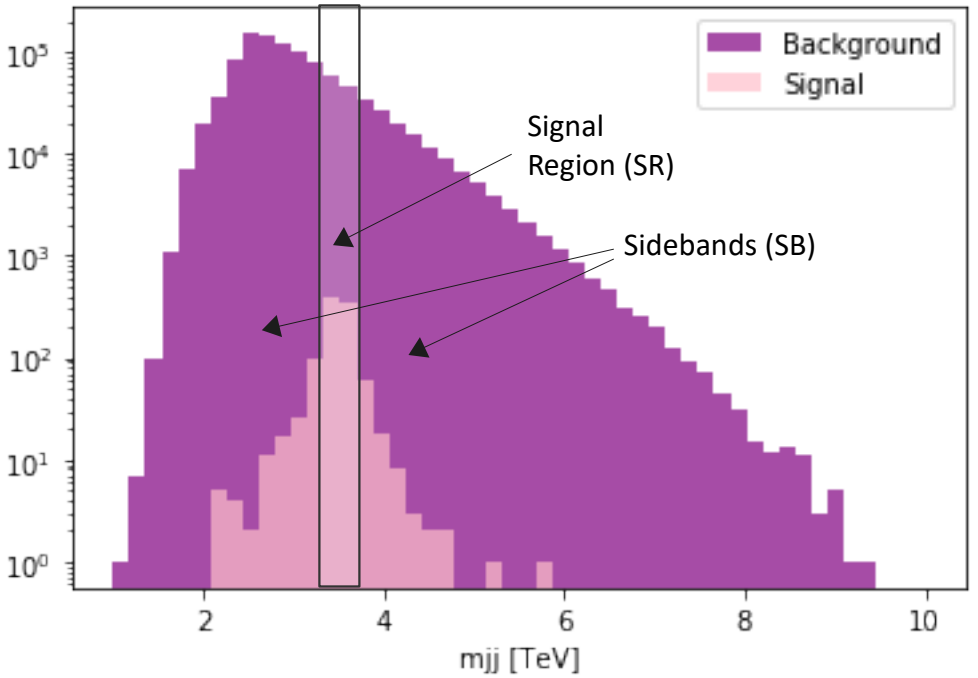
Auxiliary features:

- $m_{J1,J2}$ – the invariant masses of the individual jets
- $\tau^{21}_{J1,J2}$ – the n-subjettiness of the two jets



MAF training and sampling: training

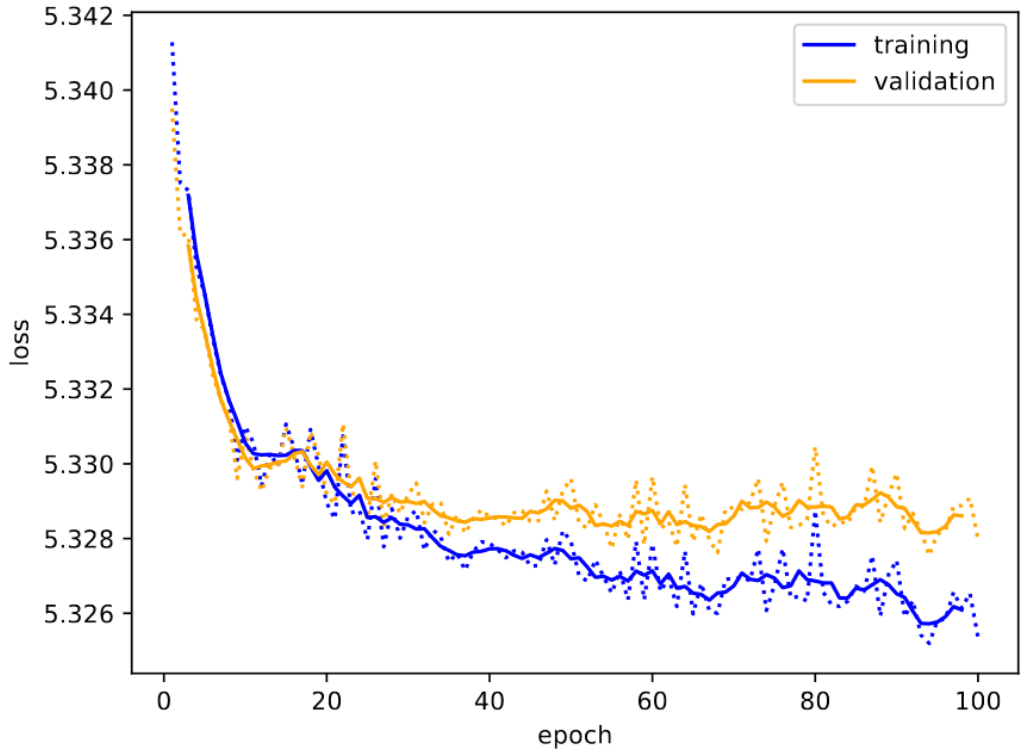
Train the MAF in the **sideband region** for 100 epochs.



MAF training and sampling: model selection

Train the MAF in the sideband region for 100 epochs.

Pick the 10 epochs with the **lowest validation loss**.



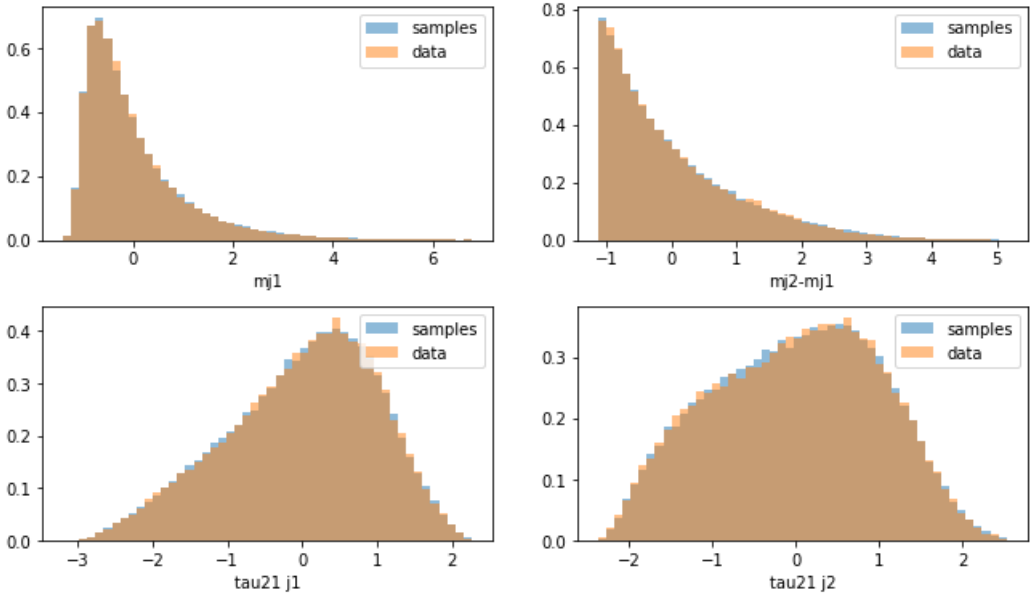
MAF training and sampling: sampling

Train the MAF in the sideband region for 100 epochs.

Pick the 10 epochs with the lowest validation loss.

Draw m_{JJ} values *in the signal region.*

Use these to **sample*** an equal number of events from each of the chosen epochs, and **combine** to one single sample. We may choose to **oversample**, generate more samples than data, which as we will see improves the performance.



Comparing samples (background model) to background in data

*We are using the MAF for sampling

Classification: getting the optimal anomaly detector

Train a classifier to distinguish between the samples we generated, and data in the signal region.

The data is a mixed dataset, as it has both signal and background. The sample will, per definition, only have background events. We know that the optimal classifier trained to distinguish between two mixed datasets (containing signal and background) is also the optimal classifier for distinguishing signal from background.

Classification with Keras

Train Keras with 3 hidden layers with 64 nodes each, ADAM as optimizer, for 100 epochs. Use class weights to re-balance the classes if oversampling has been used.

Pick the 10 epochs with the **lowest validation loss**, then **average** the predictions for each data point.

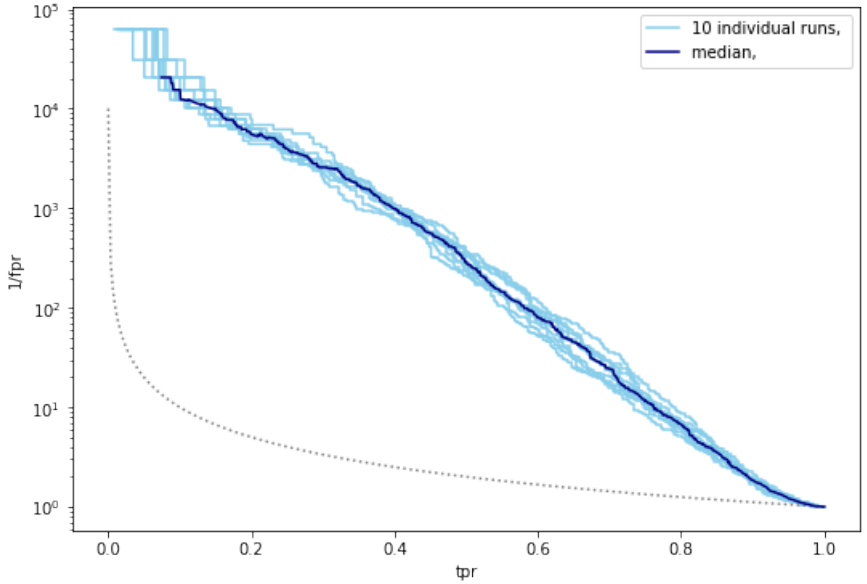
Calculate the true positive rate (TPR) and false positive rate (FPR) from the above average, and then the **significance improvement characteristic** ($\mathrm{SIC} = \mathrm{TPR}/\sqrt{\mathrm{FPR}}$).

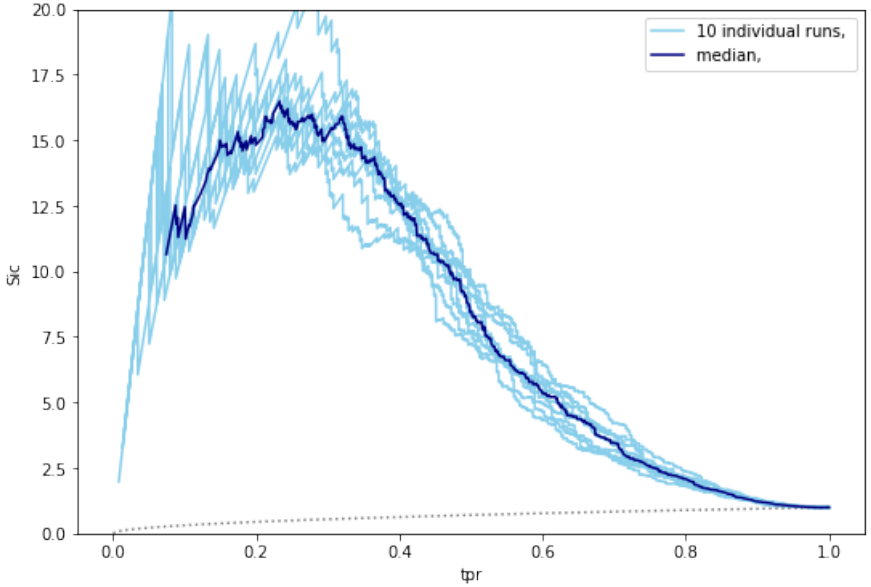
Results

Initial $S/B = 6 \times 10^{-3}$, $S/\sqrt{B} = 2.2$

Significance improvement with CATHODE: up to **15**

We also see that the classifier step is **robust**; low variability between independent runs.

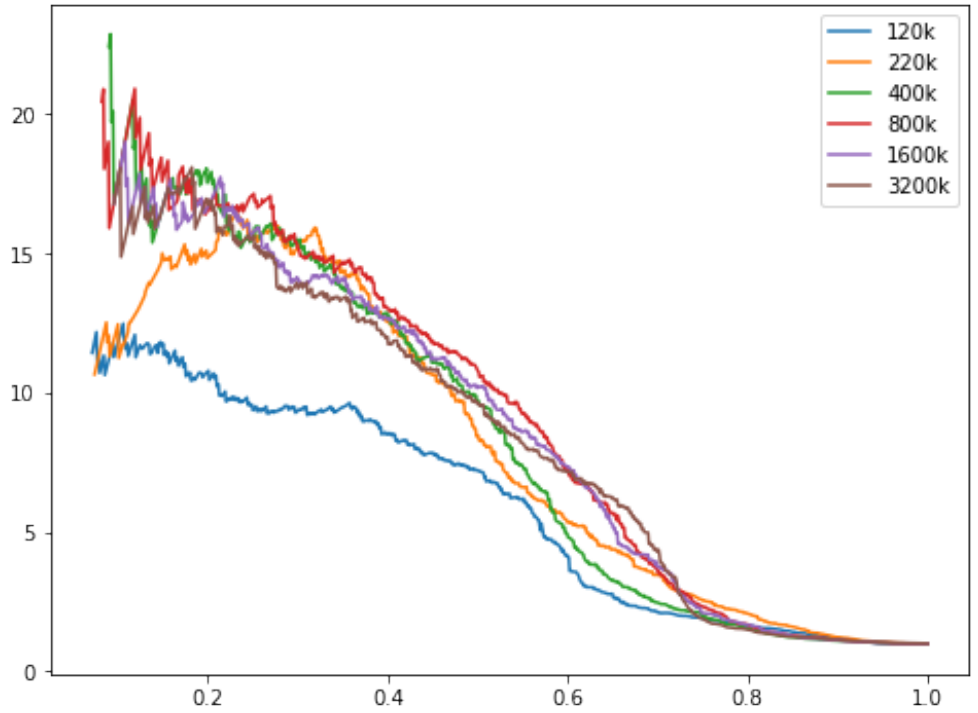




Results: oversampling

Data events: 120,000

Oversampling helps to a certain degree, as the classifier has more events to train on. Note that oversampling is not available for methods that rely only on the data.



Conclusions

- We have presented **CATHODE**: a new **model agnostic** search strategy for **resonant new physics** at the LHC .
- CATHODE learns the background density by training a Masked Autoregressive Flow in the sidebands and then interpolating to the signal region. This is a **data driven background estimation** that is robust to correlations.
- The background model is generated through **sampling** in the signal region. By oversampling we can create as much background as we wish, which improves the performance.
- The final step of CATHODE is to train a **classifier** to distinguish between data and samples.
- CATHODE can reach a **significance improvement** of up to **15** (!) – which means that an original significance of 2 will become 30.
- Further work and future directions
  - Performance vs initial signal injection
  - Background estimation
  - Other datasets (very strong results so far)
  - More or other auxiliary features

Don’t miss the next talk, coming up right after this one!

CATHODE 2: Robustness and comparison to other methods