CURTAINs Flows For Flows: Constructing Unobserved Regions with Maximum Likelihood Estimation

Debajyoti Sengupta, Sam Klein, John Andrew Raine*, and Tobias Golling

University of Geneva
debajyoti.sengupta@unige.ch samuel.klein@unige.ch john.raine@unige.ch tobias.golling@unige.ch

8th May, 2023

Abstract

Model independent techniques for constructing background data templates using generative models have shown great promise for use in searches for new physics processes at the LHC. We introduce a major improvement to the CURTAINs method by training the conditional normalizing flow between two side-band regions using maximum likelihood estimation instead of an optimal transport loss. The new training objective improves the robustness and fidelity of the transformed data and is much faster and easier to train.

We compare the performance against the previous approach and the current state of the art using the LHC Olympics anomaly detection dataset, where we see a significant improvement in sensitivity over the original CURTAINs method. Furthermore, CURTAINsF4F requires substantially less computational resources to cover a large number of signal regions than other fully data driven approaches. When using an efficient configuration, an order of magnitude more models can be trained in the same time required for ten signal regions, without a significant drop in performance.

Contents

1 Introduction

The search for new physics phenomena is one of the cornerstones of the physics programme at the Large Hadron Collider (LHC). The unparalleled energy and intensity frontier provided by the LHC provides a huge range of phase space where new signatures may be observed. The ATLAS [1] and CMS [2] collaborations at the LHC perform a wide array of searches for new particles beyond the standard model (BSM) of particle physics. Many of these searches target specific models, however, due to the vast possibilities of models and particles, dedicated searches cannot be performed for all possible scenarios.

Model independent searches aim to provide a broad sensitivity to a wide range of potential BSM scenarios without targetting specific processes. A key technique used in many searches is the bump hunt. Under the assumption that a new BSM particle is localised to a certain mass value, a bump hunt scans over an invariant mass distribution looking for excesses on top of a smooth background. Bump hunts were crucial in the observation of the Higgs boson by the ATLAS and CMS collaborations [3,4]. However, despite the success at finding the Higgs boson, there is little evidence for any BSM particles at either experiment [5–10]. With advances in machine learning (ML) many new model independent methods have been proposed to enhance the sensitivity to BSM physics [11–29] including approaches which aim to improve the sensitivity of the bump hunt itself [30–37].

In this work we improve upon the CURTAINs approach [35] by replacing the optimal transport loss used to train a flow between two complex distributions with maximum likelihood estimation. In order to evaluate the likelihood of the complex distributions on either side of the normalizing flow, we use the *Flows for Flows* technique introduced in Ref. [38] and applied to physics processes in Ref. [39]. This new configuration is called CURTAINsF4F.

We apply CURTAINsF4F to the LHC Olympics (LHCO) R&D dataset [40], a community challenge dataset for developing and comparing anomaly detection techniques in high energy physics [23]. We compare it to the previous iteration of CURTAINs, as well as to a current state of the art data driven approach CATHODE [32]. We evaluate the performance both in terms of improved signal sensitivity, but also in the required computational time to train the background models for a number of signal regions.

2 Dataset

We evaluate the performance of CURTAINsF4F using the LHC Olympics R&D dataset.

The LHCO R&D dataset [40] comprises background data produced through QCD dijet production, with signal events arising from the all-hadronic decay of a massive particle to two other massive particles $W' \to X(\to q\bar{q})Y(\to q\bar{q})$, each with masses $m_{W'} = 3.5$ TeV, $m_X = 500$ GeV, and $m_Y = 100$ GeV. Both processes are simulated with `Pythia` 8.219 [41] and interfaced to `Delphes` 3.4.1 [42] for detector simulation. Jets are reconstructed using the anti-k_T clustering algorithm [43] with a radius parameter $R = 1.0$, using the `FastJet` [44] package. In total there are 1 million QCD dijet events and 100 000 signal events.

Events are required to have exactly two large radius jets, with at least one passing a transverse momentum requirement $p_{\mathrm{T}}^{J} > 1.2$ TeV. Jets are ordered by decreasing mass. In order to remove the turn on in the m_{JJ} distribution arising from the jet selections, we only consider events with $m_{JJ} > 2.8$ TeV. To construct the training datasets we use varying amounts of signal events mixed in with the QCD dijet data.

To study the performance of our method in enhancing the sensitivity in a bump hunt, we use the input features proposed in Refs. [30–32, 35]. These are

$$m_{JJ}, m_{J_1}, \Delta m_J = m_{J_1} - m_{J_2}, \tau_{21}^{J_1}, \tau_{21}^{J_2}, \text{and } \Delta R_{JJ},$$

where τ_{21} is the N-subjettinness [45] ratio of a large radius jet, and ΔR_{JJ} is the angular separation of the two jets in the detector $\eta - \phi$ space.

3 Method

CURTAINSF4F follows the same motivation and approach as the original CURTAINS method presented in Ref. [35]. In bump hunt searches, data are categorised into non overlapping signal (SR) and side-band (SB) regions on a resonant distribution (m_{JJ}). In CURTAINS, a conditional Invertible Neural Network (cINN) is trained to learn the mapping from data drawn from one set of m_{JJ} values to a target set of values. The cINN is trained using the SB regions and applied to transport data from the SB to the SR.

However, the approach improves upon CURTAINS by using a maximum likelihood loss on the transported data instead of an optimal transport loss between the batch of data and a batch of target data.

3.1 Flows for Flows architecture

A normalizing flow trained with maximum likelihood estimation requires an invertible neural network and a base distribution with an evaluable density. The standard choice for the base distribution is a standard normal distribution. The loss function for training the normalizing flow $f_\phi(z)$ from some distribution $x \sim X$ to the base distribution $z \sim p_{prior}$ is given from the change of variables formula

$$\log p_{\theta,\phi}(x) = \log p_\theta(f_\phi^{-1}(x)) - \log \left| \det(J_{f_\phi^{-1}(x)}) \right|,$$

where J is the Jacobian of f_ϕ. In the conditional case this extends to

$$\log p_{\theta,\phi}(x|c) = \log p_\theta \left(f_{\phi(c)}^{-1}(x|c) \right) - \log \left| \det(J_{f_{\phi(c)}^{-1}(x|c)}) \right|, \tag{1}$$

where c are the conditional properties.

In Eq. (1), the base density term $\log p_\theta \left(f_{\phi(c)}^{-1}(x|c) \right)$ introduces a problem for training CURTAINS with maximum likelihood estimation. As the network is trained between two regions sampled from some non-analytically defined distribution, the probability of the transformed data is unknown. As a result, an optimal transport loss was used instead.

However, the base density of a normalizing flow can be chosen as any distribution for which the the density is known. Therefore, we can train an additional normalizing flow to learn the conditional density $p_\theta\left(f_{\phi(c)}^{-1}(x|c)\right)$ of the target data distribution. This second normalizing flow, the base distribution, is trained in advance and is used to define the loss in Eq. (1) when training the normalizing flow on arbitrary target distributions. In CURTAINSF4F the conditional properties of the top flow are a function of the input (x) and target (y) conditional properties c_x and c_y. For the base distribution only a single conditional property c is

needed. The correspondence between the top normalizing flow and the base distributions in Flows for Flows is shown in Fig. 1.

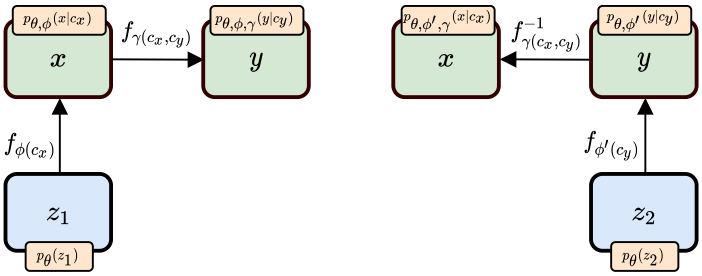

Figure 1: The Flows for Flows architecture for a conditional model [38]. Data x (y) are drawn from the initial distribution with conditional values c_x (c_y) and transformed to new values c_y (c_x) in a cINN $f_\gamma(c_x, c_y)$ conditioned on c_x and c_y. The probability of the transformed data points are evaluated using a second normalizing flow for the base distribution $f_{\phi'(c_y)}$ ($f_{\phi(c_x)}$). In the case where x and y are drawn from the same underlying distribution $p(x, c)$, the same base distribution f_ϕ can be used.

3.2 Training CURTAINSF4F

As with the original training method, CURTAINSF4F can be trained in both directions. The forward pass transforms data from low to higher target values of m_{JJ}, whereas the inverse pass transforms data from high to lower target values. When training between two distinct arbitrary distributions in both directions, a base distribution is required for each distribution.

In principle, CURTAINSF4F could be trained between data drawn from the low m_{JJ} SB (SB1) to the high m_{JJ} SB (SB2), as performed with CURTAINS. However, as data no longer need to be compared to a target batch, it is possible to train with both SBs combined in a simplified training.

Data are drawn from both SBs and target m_{JJ} values (m_{target}) are randomly assigned to each data point using all m_{JJ} values in the batch. Data are passed through the network in a forward or inverse pass depending on whether m_{target} is larger or smaller than their initial m_{JJ} (m_{input}). The network is conditioned on a function of m_{input} and m_{target} with the two values ordered in ascending order ($f(m_{jj}^{low}, m_{jj}^{high})$). This function could be, for example, the concatenation of or difference between m_{jj}^{low} and m_{jj}^{high}.

The probability term is evaluated using a single base distribution trained on the data from SB1 and SB2. The loss for the batch is calculated from the average of the probabilities calculated from the forward and inverse passes. A schematic overview is shown in Fig. 2.

Implementation

The CURTAINSF4F architecture comprises two conditional normalizing flows, the base distribution and the transformer flow. The base distribution learns the conditional density of the training data which is used to train the top flow. The top flow in turn transforms data from initial values of m_{JJ} to target values.

The base distribution is trained on the side-band data with a standard normal distribution as the target prior. It is conditioned on m_{JJ}. The top flow is trained between data drawn from the side-bands. The transformation is conditioned on $\Delta m_{JJ} = m_{JJ}^{high} - m_{JJ}^{low}$. The base distribution flow consists of ten autoregressive transformations using RQ splines, defined by four bins. The top flow consists of eight coupling transformations using RQ splines, defined by four bins. They are trained separately using the Adam optimiser and a cosine annealing learning rate scheduler. Each are trained for 100 epochs with a batch size of 256.

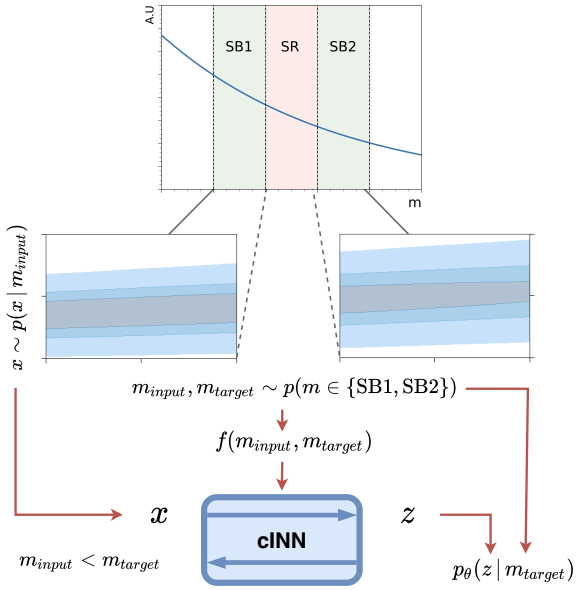

Figure 2: A schematic overview of the training procedure for CURTAINSF4F for an event where the target m_{JJ} value is greater than the input value. A single conditional normalizing flow is used for the base distribution, conditioned on the target m_{JJ} value m_{target}, to determine $p_\theta(z|m_{target})$. The top normalizing flow is conditioned on a function of the input (m_{input}) and target (m_{target}) m_{JJ} values. For the case where $m_{target} < m_{input}$, an inverse pass of the network is used and the conditioning property is calculated as $f(m_{target}, m_{input})$.

3.3 Generating background samples

To transform the data from the side-bands into the signal region, we assign target m_{JJ} values corresponding to the SR to the data in each side-band. The target values m_{target} are drawn from a function of the form

$$f(z) = p_1 (1-z)^{p_2} z^{p_3}, \tag{2}$$

where $z = m_{JJ}/\sqrt{s}$, with parameters p_i learned by performing a fit to the side-band data. Data from SB1 (SB2) are transformed in a forwards (inverse) pass through the top flow with Δm_{JJ}.

The background template can be oversampled by assigning multiple target m_{JJ} values to each data point. This has been found to improve the performance of CWOLA classifiers.

Due to the bidirectional nature of the cINNs, it is also possible to generate validation samples in regions further away from the SR. These outer-bands can be used to optimise the hyperparameters of the top flow in CURTAINSF4F.

3.4 Comparison to CURTAINS

CURTAINSF4F has a much simpler training procedure than in the original CURTAINS.

In CURTAINS, the Sinkhorn loss [46] was used to train the network, with the distance measured between a batch of data sampled from the target region and the transformed data. The target m_{JJ} values for the transformed data were chosen to match the values in the target batch. However, there was no guarantee that the minimum distance corresponded to the pairing of the transformed event with the event in the target batch with the corresponding m_{target} value. Furthermore, the loss itself ignored the m_{JJ} values as the input data and target data in the batch with the corresponding m_{JJ} target value are not necessarily events that should look the same for the same m_{JJ} value. Although successful, this approach introduced a lot of stochasticity, and required a large number of epochs to converge.

Due to the new loss, the training in CURTAINSF4F no longer needs to be between two discrete regions. This has the added benefit that it removes the need for splitting the SBs and

alternating between training CURTAINS between SB1 and SB2, and within each side-band.

Finally, in CURTAINS the network was trained and updated alternating batches in the forward and inverse directions. In CURTAINSF4F a single batch has both increasing and decreasing target m_{JJ} values. As such the network weights are updated based on the average of the loss in both the forward and inverse directions for each individual batch.

Due to the additional base distribution, CURTAINSF4F is no longer defined by a single model trained for each SR. This introduces an extra model which needs to be trained and optimised. We observe, however, that overall training both the base distribution and normalizing flow between SB data is less than required to train the cINN in CURTAINS.

The additional time required to train the base distribution can also be minimised. When training CURTAINSF4F for multiple SRs, a single base distribution can be trained using all available data for all possible m_{JJ} values. For each SR, the network would only be evaluated for values in SB1 and SB2, and no bias would be introduced from data in the SR. This reduces the overall computational cost incurred when evaluating multiple signal regions.

3.5 Comparison to other approaches

This approach is one of several using normalizing flows as density models for background estimation for extending the sensitivity of bump hunts. Many of these methods, including CURTAINS and CURTAINSF4F, produce background samples for use with CWOLA bump hunting [30, 47]. In CWOLA bump hunting, classifiers are trained on data from a hypothesised signal enriched region (the SR) and a signal depleted region (the SBs). Cuts are applied onto the classifier score to reduce the amount of background and, in the presence of signal, enhance the sensitivity.

- In ANODE [31], conditional normalizing flows are trained to learn the probability of the signal and background from data drawn from the SBs and SR respectively. The normalizing flows are conditioned on m_{JJ}, and the ratio of the probabilities is used as a likelihood test.

- In CATHODE [32], a conditional normalizing flow is trained on all SB data conditioned on m_{JJ}. Samples with m_{JJ} corresponding to the distribution of data in the SR (extrapolated from a side-band fit in m_{JJ}) are generated using the normalizing flow. These generated samples form a synthetic background sample which together with the SR data are used in a CWOLA approach [30, 47].

- In FETA [37], the *Flows for Flows* approach is used to train a conditional normalizing flow between background data in a monte carlo (MC) simulated sample and the data in the side-bands, as a function of m_{JJ}. This flow is used to transport the MC events from the SR to the data space, and account for mismodelling observed in the simulated distributions. A CWOLA classifier is trained on the transported MC and the SR data.

- LACATHODE [36] addresses the problem of distribution sculpting resulting from the choice of input features. Here the CWOLA classifier is trained on the base density of CATHODE, by first passing the SR data through the CATHODE conditional normalizing flow and using samples drawn from the prior base distribution. This approach is complementary to any method training a conditional normalizing flow, such as CURTAINSF4F and FETA.

- Although not applied in the context of bump hunt searches, ABCDNN [48] uses normalizing flows to extrapolate data from one region to another, similar to FETA. However, here the flows are trained with the maximum mean discrepancy loss, similar to the approach in CURTAINS though it does not interpolate to unknown conditional values.

4 Results

The main measure of performance for background estimation approaches is by how much they improve the sensitivity to a signal in a CWoLa bump hunt [30].

We define a SR centred on the signal process with a width of 400 GeV, which contains nearly all of the signal events. For CurtainsF4F and Curtains, we use side-bands 200 GeV either side of the SR to train the methods. Only these regions are used to train the base distribution for CurtainsF4F. For Cathode, the whole m_{JJ} distribution either side of the SR is considered as the SB region. This corresponds to side-bands of widths 500 GeV and 900 GeV.

Weakly supervised classifiers are trained to separate the generated background samples from data in the SR. For Curtains, CurtainsF4F, and Cathode, an oversampling factor of four is used to generate the backround samples in the SR, at which point the performance reaches saturation. In Curtains and CurtainsF4F this is with respect to the transported SB data, whereas for Cathode it is based on the yields in the SR.

As a reference, a fully supervised classifier trained to separate the signal from background in this region, and an idealised classifier trained with a perfect background estimation are also shown. The idealised classifier is trained for both equal numbers of background in each class (Eq-Idealised) and an oversampled background (Over-Idealised).

A k-fold training strategy with five folds is employed to train all classifiers. Three fifths are used to train the classifier, with one fifth for validation and the final fifth as a hold out set. The classifiers comprise three hidden layers with 32 nodes and ReLU activations. They are trained for 20 epochs with the Adam optimiser and a batch size of 128. The initial learning rate is 10^{-4} but is annealed to zero following a cosine schedule.

4.1 Comparison of performance

Figure 3 shows the background rejection and significance improvement for CWoLa classifiers trained using the different background estimation models as the cut on the classifier is varied. Here 3,000 signal events have been added to the QCD dijet sample, of which 2,214 fall within the SR. CurtainsF4F shows significant improvement over the original Curtains method, and now matches the Cathode performance across the majority of rejection and signal efficiency values. This is despite being trained on a much smaller range of data. Curtains still displays better significant improvement at very high rejection values, however this is in a region dominated by the statistical uncertainty.

In Fig. 4 the significance improvement for each method is calculated as a function of the number of signal events added to the sample. Here both the signal fraction and raw number of signal events in the SR are reported. The significance improvement is shown for two fixed background rejection values, rather than the maximum significance improvement, due to the sensitivity to fluctuations in the high background rejection regions where there are much lower statistics. The performance of CurtainsF4F is improved across all levels of signal in comparison to the original Curtains method, and performs equally as well as Cathode.

4.2 Dependence on side-band width

In CurtainsF4F, 200 GeV wide side-bands are used to train the networks and learn a local transformation. With leakage of signal into the side-bands or changing background composition, it could be beneficial to have narrower or wider SBs, and there is no set prescription for which is optimal. Figure 5 shows the impact on performance of varying the widths from 100 GeV up to all data not contained in the signal region (max width). For the 100 GeV wide side-bands a noticeable drop in performance is observed in the significance improvement and

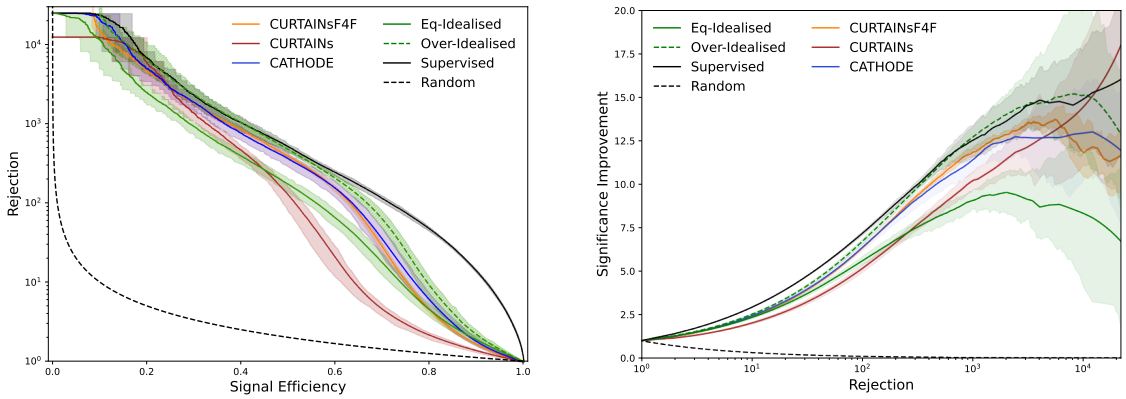

Figure 3: Background rejection as a function of signal efficiency (left) and significance improvement as a function of background rejection (right) for CURTAINS (red), CURTAINSF4F (orange), CATHODE (blue), Supervised (black), Eq-Idealised (green, solid), and Over-Idealised (green, dashed). All classifiers are trained on the sample with 3,000 injected signal events, using a signal region $3300 \leq m_{JJ} < 3700$ GeV. The lines show the mean value of fifty classifier trainings with different random seeds with the shaded band covering 68% uncertainty. A supervised classifier and two idealised classifiers are shown for reference.

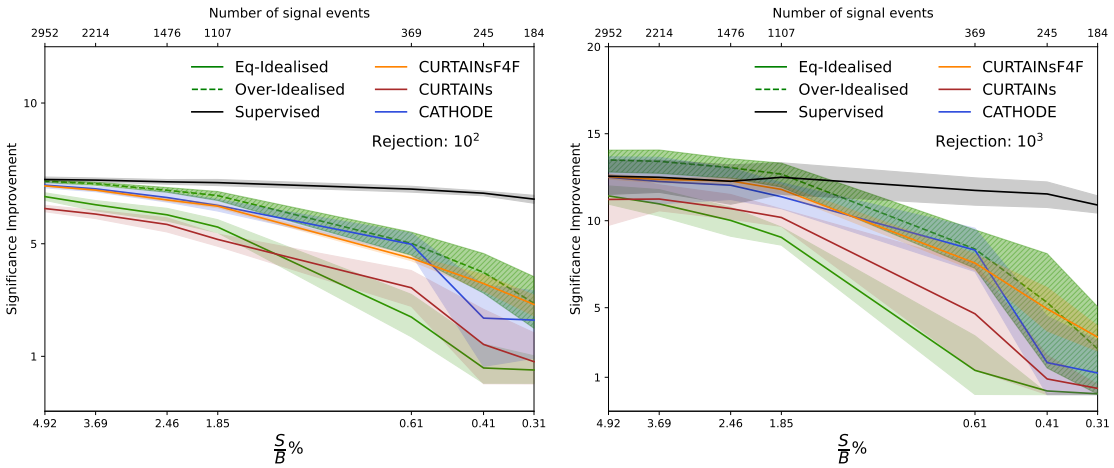

Figure 4: Significance improvement at a background rejection of 10^2 (left) and 10^3 (right) as a function of signal events in the signal region $3300 \leq m_{JJ} < 3700$ GeV, for CURTAINSF4F (orange), CURTAINS (red), CATHODE (blue), Supervised (black), Eq-Idealised (green, solid), and Over-Idealised (green, dashed). The lines show the mean value of fifty classifier trainings with different random seeds with the shaded band covering 68% uncertainty. A supervised classifier and two idealised classifiers are shown for reference.

ROC curves. However at a background rejection of $\sim 10^3$ all other side-band widths have similar levels of rejection. At higher levels of background rejection training on larger side-bands, and thus more data, results in better performance than the default CURTAINSF4F model with widths of 200 GeV. It should be kept in mind that as the width of the side-bands increase, the required training time increases. For these comparisons no hyperparameter optimisation has been performed and the default values are used for all models.

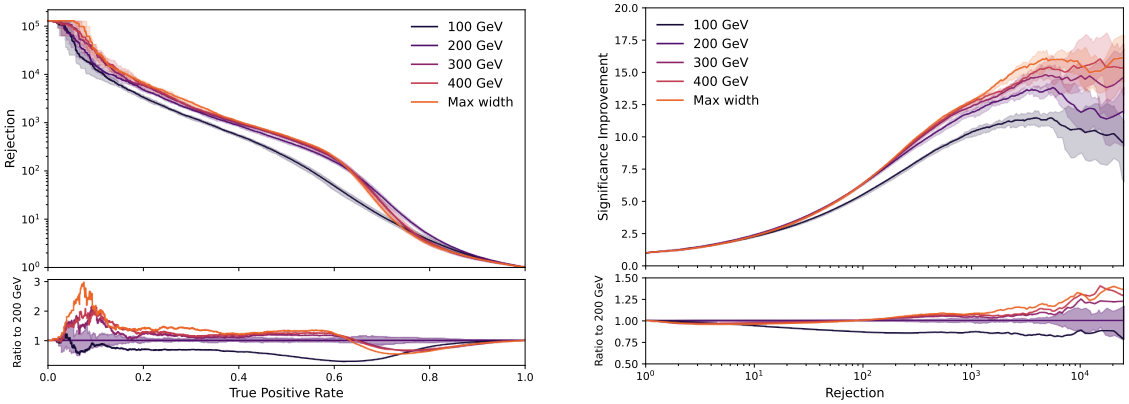

Figure 5: Background rejection as a function of signal efficiency (left) and significance improvement as a function of background rejection (right) for CURTAINSF4F trained with varying width side-bands, ranging from 100 GeV to the maximum width possible (SB1: 500 GeV, SB2: 900 GeV). All classifiers are trained on the sample with 3,000 injected signal events, using a signal region $3300 \leq m_{JJ} < 3700$ GeV. The lines show the mean value of fifty classifier trainings with different random seeds with the shaded band covering 68% uncertainty.

In Fig. 6 the performance of CURTAINSF4F and CATHODE are shown for the case where each model is trained on either 200 GeV wide SBs or the max width. For CURTAINSF4F the difference in performance is mostly at high background rejection whereas CATHODE has a drop in performance at all values.

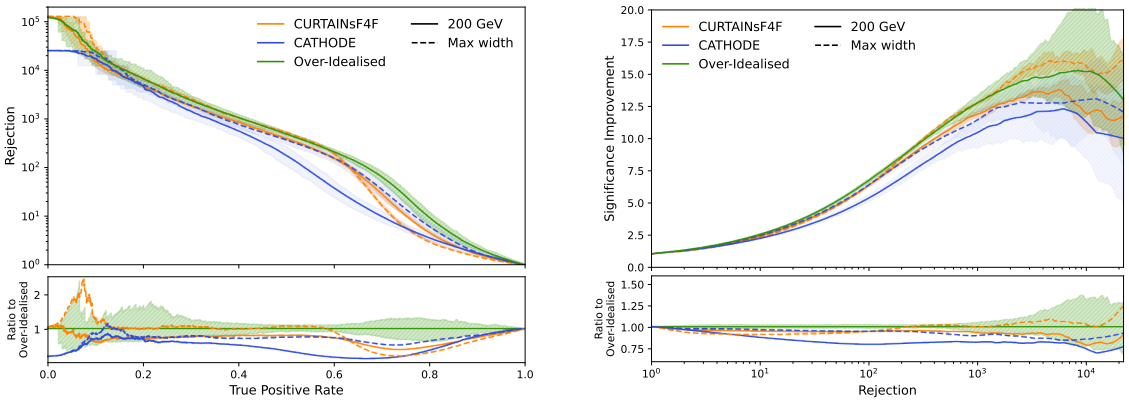

Figure 6: Background rejection as a function of signal efficiency (left) and significance improvement as a function of background rejection (right) for CURTAINSF4F (orange) and CATHODE (blue). Two side-band widths are used to train the two methods, 200 GeV side-bands (solid) and the maximum width (dashed, SB1: 500 GeV, SB2: 900 GeV). All classifiers are trained on the sample with 3,000 injected signal events, using a signal region $3300 \leq m_{JJ} < 3700$ GeV. The lines show the mean value of fifty classifier trainings with different random seeds with the shaded band covering 68% uncertainty. The Over-Idealised classifier (green) is shown for reference.

4.3 Required training time

For a bump hunt or sliding window search, a large number of models need to be trained which can result in a high demand on computing resources. As a result, the granularity of a search may be restricted in line with overall computational time. Therefore, a key measure of methods like CURTAINSF4F and CATHODE is how quick the models are to train.

In Table 1 the require time to train the two approaches for one SR are shown for convergence and for one epoch. CATHODE has an advantage over CURTAINSF4F in that only one normalizing flow is trained. The total training time required for CURTAINSF4F is much reduced in comparison to CURTAINS and is slightly faster than CATHODE for the default configurations.

Table 1: Comparison of the required time to train CURTAINS, CURTAINSF4F, and CATHODE. All models are trained on the same hardware with epoch and total training time representative of using an NVIDIA® RTX 3080 graphics card. For CURTAINSF4F two numbers are shown for the epoch time and number of epochs due to the two normalizing flows which need to be trained. Default side-band widths are used for all models, around the nominal signal region.

	Time / epoch [s]	N epochs	Total time [min]
CURTAINS	10	1000	167
CATHODE	78	100	129
CURTAINSF4F	32/32	100/100	107

4.4 Reducing computational footprint

When applying the models to multiple signal regions in a bump hunt, new models need to be trained for each step. For CATHODE this involves training a complete model each time. However, due to the modular nature of CURTAINSF4F, if the base distribution is trained on the whole spectrum, only the top flow needs to be trained with each step. As such, as soon as more than one SR is considered, CURTAINSF4F requires substantially less computational resources for a similar level of performance.

Additionally, the transformation learned by the top flow in CURTAINSF4F is known to be a smaller shift than for the base distribution or in CATHODE. The top flow can thus also be optimised for speed without sacrificing as much performance and does not require the same expressive architecture as used by default.

The default CURTAINSF4F configuration is compared to an efficient implementation in which a single base distribution is trained on all data, and the top flow is optimised for speed. The base distribution has the same architecture as the default configuration. The efficient top flow comprises two coupling transformations using RQ splines, rather than eight, with each now defined by six bins instead of four. The top flow is trained for 20 epochs with a batch size of 256. All other hyperparameters remain unchanged and side-bands of 200 GeV are used to train the top flow and produce the background template in the SR. The potential reduction in computation time for using CURTAINSF4F in a sliding window search is presented in Table 2. With the efficient configuration more than one hundred signal regions can be evaluated with CURTAINSF4F transformers for the same computational cost as ten with the default configuration.

In Fig. 7 the significance improvement when using the efficient configuration is compared to the default CURTAINSF4F model for 3000 injected signal events. The performance as a function of the number of injected signal events is shown in Fig. 8. No significant decrease in performance is observed.

Table 2: Comparison of the required time to train the base distribution and top flow in CURTAINsF4F. The default configuration comprises the base distribution and top flow trained on 200 GeV side-bands. The efficient configuration has a single base distribution trained on all data, and a top flow trained on 200 GeV side-bands and optimised for the fastest training time. All models are trained on the same hardware with epoch and total training time representative of using an NVIDIA® RTX 3080 graphics card. An extrapolation of the required total time to train a complete CURTAINsF4F model for one and ten signal regions are also shown for the two configurations. The extrapolated time for 125 signal regions is also shown for the efficient configuration, requiring less time than ten signal regions with the default configuration.
† Timing is for the nominal side-bands, this would vary as the signal region changes due to total number of training events.

	Time / epoch [s]	N epochs	Total time [min]
	Default		
Base	32.4†	100	54
Top flow	31.5†	100	53
		One Signal Region	107
		(Extrapolated†) Ten Signal Region	1070
	Efficient		
Base	104.2	100	174
Top flow	21.3†	20	7
		One Signal Region	181
		(Extrapolated†) Ten Signal Region	244
		(Extrapolated†) 125 Signal Region	1049

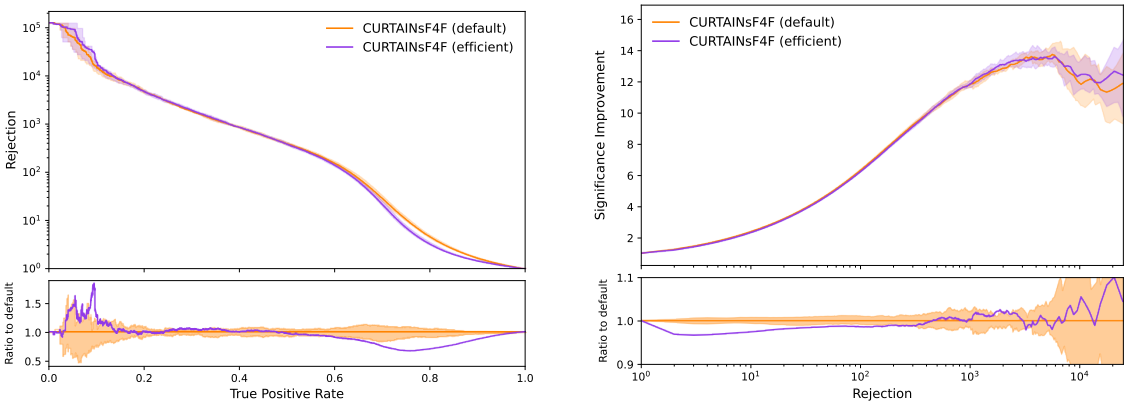

Figure 7: Background rejection as a function of signal efficiency (left) and significance improvement as a function of background rejection (right) for CURTAINsF4F using the default (orange) and efficient (purple) training configurations. All classifiers are trained on the sample with 3,000 injected signal events, using a signal region $3300 \leq m_{JJ} < 3700$ GeV. The lines show the mean value of fifty classifier trainings with different random seeds with the shaded band covering 68% uncertainty.

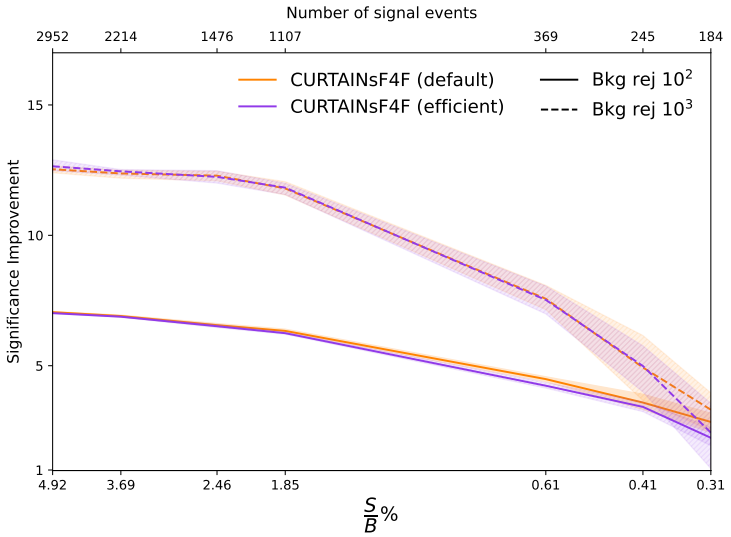

Figure 8: Significance improvement at a background rejection of 10^2 and 10^3 as a function of signal events in the signal region $3300 \leq m_{JJ} < 3700$ GeV for CURTAINsF4F using the default (orange) and efficient (purple) training configurations. The lines show the mean value of fifty classifier trainings with different random seeds with the shaded band covering 68% uncertainty.

5 Conclusions

In the original CURTAINs method, a distance based optimal transport loss was used to train a conditional invertible neural network. In this work we have shown that the performance can be improved significantly by moving to a maximum likelihood estimation loss, using the *Flows for Flows* methodology. The performance levels reached by CURTAINsF4F are state-of-the-art, and can do so training on less data from narrower side-bands than the previous state of the art.

By only modifying the training procedure, other advantages of CURTAINs are preserved. Additional validation regions further away from the signal region can be used to optimise the hyperparameters of both the normalizing flow and classification networks.

Furthermore, in order to address background sculpting resulting from the classifiers, the latent approach introduced in LACATHODE can be performed using the base distribution. With the original CURTAINs method, an additional normalizing flow would need to be trained on the signal region data for each signal region.

Finally, for a single signal region CURTAINsF4F requires similar computing resources as other leading approaches, with almost half the required training time in comparison to the original CURTAINs method. However, when moving to a sliding window bump hunt, the overall computing resources required for CURTAINsF4F is reduced by a large factor. On the LHCO R&D dataset over one hundred signal regions can be trained for the same computing resources as otherwise required for ten signal regions. This could be of particular interest for large scale searches which are limited by the computational cost to cover a larger number of signal regions, such as those in Refs. [49, 50] amongst others.

Acknowledgements

We would like to thank David Shih and Matt Buckley for valuable discussions at the ML4Jets 2022 conference at Rutgers, New Jersey, in particular on results of interest and potential ap-

plications. In addition, we would like to thank Radha Mastandrea, Ben Nachman and Kees Benkendorfer for useful discussions concerning the performance evaluation.

The authors would like to acknowledge funding through the SNSF Sinergia grant called "Robust Deep Density Models for High-Energy Particle Physics and Solar Flare Analysis (RODEM)" with funding number CRSII5_193716 and the SNSF project grant 200020_212127 called "At the two upgrade frontiers: machine learning and the ITk Pixel detector".

References

[1] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST **3**, S08003 (2008), doi:10.1088/1748-0221/3/08/S08003.

[2] CMS Collaboration, *The CMS Experiment at the CERN LHC*, JINST **3**, S08004 (2008), doi:10.1088/1748-0221/3/08/S08004.

[3] ATLAS Collaboration, *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, Phys. Lett. B **716**, 1 (2012), doi:10.1016/j.physletb.2012.08.020, https://arxiv.org/abs/1207.7214.

[4] CMS Collaboration, *Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC*, Phys. Lett. B **716**, 30 (2012), doi:10.1016/j.physletb.2012.08.021, https://arxiv.org/abs/1207.7235.

[5] ATLAS Collaboration, *SUSY Summary Plots June 2021*, Available at https://cds.cern.ch/record/2771785, accessed 14th February 2023 (2021).

[6] ATLAS Collaboration, *Summary Plots from ATLAS Searches for Pair-Produced Leptoquarks*, Available at https://cds.cern.ch/record/2771726, accessed 14th February 2023 (2021).

[7] ATLAS Collaboration, *Summary Plots for Heavy Particle Searches and Long-lived Particle Searches - July 2021*, Available at https://cds.cern.ch/record/2777015, accessed 14th February 2023 (2021).

[8] CMS Collaboration, *CMS EXO sumary plots at 13TeV*, Accessed Available at https://twiki.cern.ch/twiki/bin/view/CMSPublic/SummaryPlotsEXO13TeV, a4th February 2023 (2022).

[9] CMS Collaboration, *CMS B2G physics results summary*, Available at https://twiki.cern.ch/twiki/bin/view/CMSPublic/PhysicsResultsB2G, accessed 14th February 2023 (2022).

[10] CMS Collaboration, *CMS SUS physics results summary*, Available at https://twiki.cern.ch/twiki/bin/view/CMSPublic/PhysicsResultsSUS, accessed 14th February 2023 (2022).

[11] J. A. Aguilar-Saavedra, J. H. Collins and R. K. Mishra, *A generic anti-QCD jet tagger*, JHEP **11**, 163 (2017), doi:10.1007/JHEP11(2017)163, https://arxiv.org/abs/1709.01087.

[12] J. Hajer, Y.-Y. Li, T. Liu and H. Wang, *Novelty Detection Meets Collider Physics*, Phys. Rev. D **101**(7), 076015 (2020), doi:10.1103/PhysRevD.101.076015, https://arxiv.org/abs/1807.10261.

[13] T. Heimel, G. Kasieczka, T. Plehn and J. M. Thompson, *QCD or What?*, SciPost Phys. **6**(3), 030 (2019), doi:10.21468/SciPostPhys.6.3.030, https://arxiv.org/abs/1808.08979.

[14] M. Farina, Y. Nakai and D. Shih, *Searching for New Physics with Deep Autoencoders*, Phys. Rev. D **101**(7), 075021 (2020), doi:10.1103/PhysRevD.101.075021, https://arxiv.org/abs/1808.08992.

[15] O. Cerri, T. Q. Nguyen, M. Pierini, M. Spiropulu and J.-R. Vlimant, *Variational autoencoders for new physics mining at the large hadron collider*, Journal of High Energy Physics **2019**(5) (2019), doi:10.1007/jhep05(2019)036.

[16] T. S. Roy and A. H. Vijay, *A robust anomaly finder based on autoencoders* (2019), https://arxiv.org/abs/1903.02032.

[17] A. Blance, M. Spannowsky and P. Waite, *Adversarially-trained autoencoders for robust unsupervised new physics searches*, Journal of High Energy Physics **2019**(10) (2019), doi:10.1007/jhep10(2019)047.

[18] P. Jawahar, T. Aarrestad, N. Chernyavskaya, M. Pierini, K. A. Wozniak, J. Ngadiuba, J. Duarte and S. Tsan, *Improving Variational Autoencoders for New Physics Detection at the LHC With Normalizing Flows*, Front. Big Data **5**, 803685 (2022), doi:10.3389/fdata.2022.803685, https://arxiv.org/abs/2110.08508.

[19] R. T. D'Agnolo and A. Wulzer, *Learning new physics from a machine*, Physical Review D **99**(1) (2019), doi:10.1103/physrevd.99.015014.

[20] A. D. Simone and T. Jacques, *Guiding new physics searches with unsupervised learning*, The European Physical Journal C **79**(4) (2019), doi:10.1140/epjc/s10052-019-6787-3.

[21] M. Letizia, G. Losapio, M. Rando, G. Grosso, A. Wulzer, M. Pierini, M. Zanetti and L. Rosasco, *Learning new physics efficiently with nonparametric methods* (2022), https://arxiv.org/abs/2204.02317.

[22] R. T. D'Agnolo, G. Grosso, M. Pierini, A. Wulzer and M. Zanetti, *Learning multivariate new physics*, Eur. Phys. J. C **81**(1), 89 (2021), doi:10.1140/epjc/s10052-021-08853-y, https://arxiv.org/abs/1912.12155.

[23] G. Kasieczka *et al.*, *The LHC Olympics 2020 a community challenge for anomaly detection in high energy physics*, Rept. Prog. Phys. **84**(12), 124201 (2021), doi:10.1088/1361-6633/ac36b9, https://arxiv.org/abs/2101.08320.

[24] T. Aarrestad *et al.*, *The Dark Machines Anomaly Score Challenge: Benchmark Data and Model Independent Event Classification for the Large Hadron Collider*, SciPost Phys. **12**(1), 043 (2022), doi:10.21468/SciPostPhys.12.1.043, https://arxiv.org/abs/2105.14027.

[25] B. M. Dillon, L. Favaro, F. Feiden, T. Modak and T. Plehn, *Anomalies, Representations, and Self-Supervision* (2023), https://arxiv.org/abs/2301.04660.

[26] M. F. Chen, B. Nachman and F. Sala, *Resonant Anomaly Detection with Multiple Reference Datasets* (2022), https://arxiv.org/abs/2212.10579.

[27] G. Kasieczka, R. Mastandrea, V. Mikuni, B. Nachman, M. Pettee and D. Shih, *Anomaly detection under coordinate transformations*, Phys. Rev. D **107**(1), 015009 (2023), doi:10.1103/PhysRevD.107.015009, https://arxiv.org/abs/2209.06225.

[28] B. M. Dillon, R. Mastandrea and B. Nachman, *Self-supervised anomaly detection for new physics*, Phys. Rev. D **106**(5), 056005 (2022), doi:10.1103/PhysRevD.106.056005, https://arxiv.org/abs/2205.10380.

[29] T. Finke, M. Krämer, M. Lipp and A. Mück, *Boosting mono-jet searches with model-agnostic machine learning*, JHEP **08**, 015 (2022), doi:10.1007/JHEP08(2022)015, https://arxiv.org/abs/2204.11889.

[30] J. H. Collins, K. Howe and B. Nachman, *Extending the search for new resonances with machine learning*, Phys. Rev. D **99**(1), 014038 (2019), doi:10.1103/PhysRevD.99.014038, https://arxiv.org/abs/1902.02634.

[31] B. Nachman and D. Shih, *Anomaly Detection with Density Estimation*, Phys. Rev. D **101**, 075042 (2020), doi:10.1103/PhysRevD.101.075042, https://arxiv.org/abs/2001.04990.

[32] A. Hallin, J. Isaacson, G. Kasieczka, C. Krause, B. Nachman, T. Quadfasel, M. Schlaffer, D. Shih and M. Sommerhalder, *Classifying anomalies through outer density estimation*, Phys. Rev. D **106**(5), 055006 (2022), doi:10.1103/PhysRevD.106.055006.

[33] A. Andreassen, B. Nachman and D. Shih, *Simulation Assisted Likelihood-free Anomaly Detection*, Phys. Rev. D **101**(9), 095004 (2020), doi:10.1103/PhysRevD.101.095004, https://arxiv.org/abs/2001.05001.

[34] K. Benkendorfer, L. L. Pottier and B. Nachman, *Simulation-assisted decorrelation for resonant anomaly detection*, Phys. Rev. D **104**(3), 035003 (2021), doi:10.1103/PhysRevD.104.035003, https://arxiv.org/abs/2009.02205.

[35] J. A. Raine, S. Klein, D. Sengupta and T. Golling, *CURTAINs for your Sliding Window: Constructing Unobserved Regions by Transforming Adjacent Intervals*, Front. Big Data **6**, 899345 (2023), doi:10.3389/fdata.2023.899345.

[36] A. Hallin, G. Kasieczka, T. Quadfasel, D. Shih and M. Sommerhalder, *Resonant anomaly detection without background sculpting* (2022), https://arxiv.org/abs/2210.14924.

[37] T. Golling, S. Klein, R. Mastandrea and B. Nachman, *FETA: Flow-Enhanced Transportation for Anomaly Detection* (2022), https://arxiv.org/abs/2212.11285.

[38] S. Klein, J. A. Raine and T. Golling, *Flows for flows: Training normalizing flows between arbitrary distributions with maximum likelihood estimation*, doi:10.48550/ARXIV.2211.02487 (2022).

[39] R. Mastandrea and B. Nachman, *Efficiently Moving Instead of Reweighting Collider Events with Machine Learning*, In *36th Conference on Neural Information Processing Systems* (2022), 2212.06155.

[40] G. Kasieczka, B. Nachman and D. Shih, *R&D Dataset for LHC Olympics 2020 Anomaly Detection Challenge*, doi:10.5281/zenodo.4536377 (2019).

[41] T. Sjöstrand, S. Mrenna and P. Z. Skands, *A Brief Introduction to PYTHIA 8.1*, Comput. Phys. Commun. **178**, 852 (2008), doi:10.1016/j.cpc.2008.01.036, https://arxiv.org/abs/0710.3820.

[42] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens and M. Selvaggi, *DELPHES 3, A modular framework for fast simulation of a generic collider experiment*, JHEP **02**, 057 (2014), doi:10.1007/JHEP02(2014)057, https://arxiv.org/abs/1307.6346.

[43] M. Cacciari, G. P. Salam and G. Soyez, *The anti-k_t jet clustering algorithm*, JHEP **04**, 063 (2008), doi:10.1088/1126-6708/2008/04/063, https://arxiv.org/abs/0802.1189.

[44] M. Cacciari, G. P. Salam and G. Soyez, *FastJet User Manual*, Eur. Phys. J. C **72**, 1896 (2012), doi:10.1140/epjc/s10052-012-1896-2, https://arxiv.org/abs/1111.6097.

[45] J. Thaler and K. Van Tilburg, *Identifying boosted objects with n-subjettiness*, Journal of High Energy Physics **2011**(3) (2011), doi:10.1007/jhep03(2011)015.

[46] M. Cuturi, *Sinkhorn distances: Lightspeed computation of optimal transportation distances* (2013), 1306.0895.

[47] E. M. Metodiev, B. Nachman and J. Thaler, *Classification without labels: Learning from mixed samples in high energy physics*, JHEP **10**, 174 (2017), doi:10.1007/JHEP10(2017)174, https://arxiv.org/abs/1708.02949.

[48] S. Choi, J. Lim and H. Oh, *Data-driven Estimation of Background Distribution through Neural Autoregressive Flows* (2020), https://arxiv.org/abs/2008.03636.

[49] D. Shih, M. R. Buckley, L. Necib and J. Tamanas, *via machinae: Searching for stellar streams using unsupervised machine learning*, Mon. Not. Roy. Astron. Soc. **509**(4), 5992 (2021), doi:10.1093/mnras/stab3372, https://arxiv.org/abs/2104.12789.

[50] D. Shih, M. R. Buckley and L. Necib, *Via Machinae 2.0: Full-Sky, Model-Agnostic Search for Stellar Streams in Gaia DR2* (2023), https://arxiv.org/abs/2303.01529.

A Additional results

In Fig. 9 the maximum significance improvement for the default models is shown, rather than at fixed background rejection values.

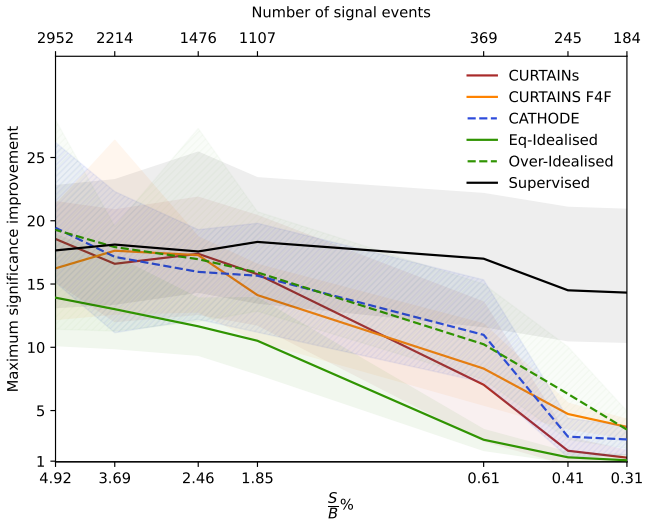

Figure 9: Maximum significance improvement as a function of signal events in the signal region $3300 \leq m_{JJ} < 3700$ GeV, for CURTAINs (red), CURTAINsF4F (orange), CATHODE (blue), Supervised (black), Eq-Idealised (green, solid), and Over-Idealised (green, dashed). The lines show the mean value of fifty classifier trainings with different random seeds with the shaded band covering 68% uncertainty. A supervised classifier and two idealised classifiers are shown for reference.

An investigation on the sensitivity of CURTAINsF4F to the amount of oversampling is shown in Fig. 10. At a factor of four (default) the performance saturates.

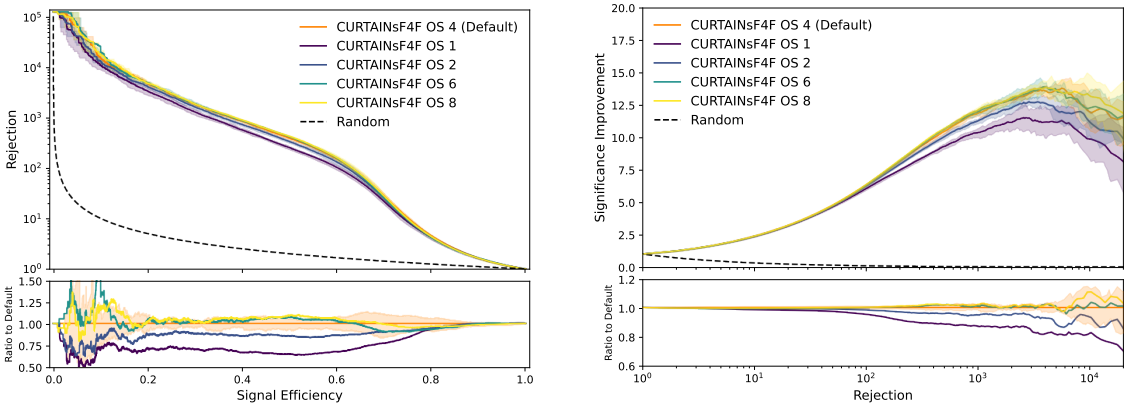

Figure 10: Background rejection as a function of signal efficiency (left) and significance improvement as a function of background rejection (right) for CURTAINsF4F trained with varying amounts of oversampling using 200 GeV side-bands All classifiers are trained on the sample with 3,000 injected signal events, using a signal region $3300 \leq m_{JJ} < 3700$ GeV. The lines show the mean value of fifty classifier trainings with different random seeds with the shaded band covering 68% uncertainty.

In Table 3 the extrapolated times are computed using the faster top flow but with a new base flow for each signal region. Although there is a significant time improvement over the default configuration, the efficient implementation still almost three times faster for ten signal regions, and a factor of seven more signal regions can be trained in just over 1000 minutes.

Table 3: The required time to train the base distribution and top flow in CURTAINSF4F using the faster top flow but a base distribution for each signal region. The base distribution and top flow trained on 200 GeV side-bands. The models are trained on the same hardware with epoch and total training time representative of using an NVIDIA® RTX 3080 graphics card. An extrapolation of the required total time to train a complete CURTAINSF4F model for one and ten signal regions are also shown for the two configurations. †Timing is for the nominal side-bands, this would vary as the signal region changes due to total number of training events.

	Time / epoch [s]	N epochs	Total time [min]
	Faster		
Base	32.4†	100	54
Top flow	21.3†	20	7
		One Signal Region	61
	(Extrapolated†)	Ten Signal Region	610
	(Extrapolated†)	17 Signal Region	1037

B Hyperparameters

Table 4: Hyperparameters for training the flows in CURTAINSF4F.

	Base distribution	Top flow (default)	Top flow (efficient)
Number of RQ splines	10	8	2
Number of bins per spline	4	4	6
Transformation	Autoregressive	Coupling	Coupling
Blocks per spline	2	2	6
Hidden nodes per block	128	32	64
Number of epochs	100	100	20
Batch size	256	256	256
Optimiser	Adam	Adam	Adam
Initial learning rate	1e-4	1e-4	1e-4
Cosine annealing	True	True	True