13

Generative models for data-driven background modeling

Go with the flow. Force nothing. Let it happen, or not happen... trusting that whichever way it goes, it's for the best.

– Mandy Hale

As described in Chapter 11, QCD is 90% of the $4b$ background and needs to be estimated with data-driven techniques. In the Run 2 $4b$ analysis, reweighting maps were derived in dedicated CRs and then extrapolated into the SR. The uncertainty on this background estimate limits analysis sensitivity – in particular, the uncertainty from CRs extrapolating into the SR degrades the upper limit on μ_{SM} by 7.5% (Table 12.3).

This chapter proposes an alternative background estimate which relies on a different set of assumptions. Figure 13.1 shows that the mulit-jet background processes vary smoothly across the (m_{H1}, m_{H2}) massplane. This smooth variation of the underlying physics is used as a key inductive bias to train a generative model as a function of (m_{H1}, m_{H2}) to pose the background estimate as a high dimensional interpolation. This is done with a two-step procedure where the Higgs candidate masses and the HH kinematics (given by the vector x) are modeled separately. The joint probability distribution is decomposed using the chain rule of probability as:

$$p(x, m_{H1}, m_{H2}) = p(x|m_{H1}, m_{H2}) \cdot p(m_{H1}, m_{H2}), \tag{13.1}$$

where x is a six-dimensional vector that models the rest of the Higgs candidates' kinematics (as will be elaborated on in Chapter 13.2.2). The $p(m_{h1}, m_{h2})$ is fit with a Gaussian process (GP, described in Chapter 13.1) to constrain the predictions inside of the SR using the correlation with events outside of the SR. Since GP does not scale well to high dimensional inputs [164], the rest of the event kinematics

x are modeled with a conditional normalizing flow: $p(x|m_{h1}, m_{h2})$ (described in Chapter 13.2). Since the smoothly varying (m_{H1}, m_{H2}) is a key modeling assumption, the interpolation is derived before applying the $X_{Wt} > 1.5$ cut which induced structures in the massplane.

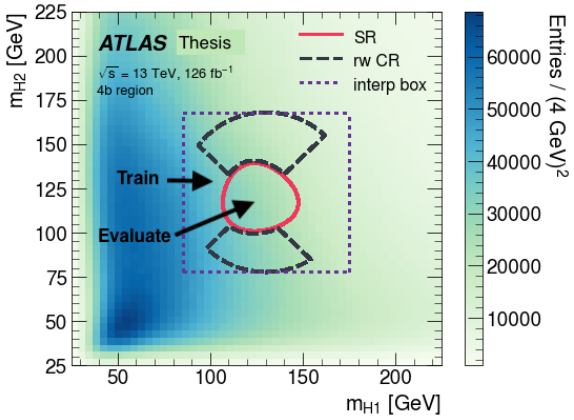


Figure 13.1: Illustration of the interpolation setup. The pink circle shows the nominal 4b SR, with the purple dotted line showing the bounding box used to train the interpolation. The quadrants for deriving the baseline reweighing method are shown in the navy blue dashed line crescents.

Here the ggF analysis background estimate is studied as this channel is more impacted by the background systematics than the VBF analysis which is dominated by statistical uncertainties. The selection here is identical to that outlined in Chapter 10, and the results for this interpolation are compared to the reweighting strategy in both the 4b SR, and the suite of validation regions.

13.1 Gaussian Processes

13.1.1 Introduction

The $p(m_{H1}, m_{H2})$ model fits a GP to the blinded 2d histogram, inspired by the boosted $4b$ analysis [114]. A GP is a Bayesian non-parametric ML method, which means that the model size grows with the number of input data points [165, 166]. It generalizes the notion of the multivariate Gaussian which describes distributions over random vectors, to describe the distributions over functions (to allow evaluations on arbitrary inputs) – where here we want to evaluate the distribution in the blinded SR. The GP specifies that the prior over any finite collection of points follows a multivariate Gaussian probability distribution, as specified by:

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(x_1) \\ \vdots \\ m(x_m) \end{bmatrix}, \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_m) \\ \vdots & \ddots & \vdots \\ K(x_m, x_1) & \cdots & K(x_m, x_m) \end{bmatrix} \right) \tag{13.2}$$

where $x_1, \ldots, x_m$ are the input points, and the f describes the prior over functions. As a pre-processing step, the targets are normalized to zero mean and unit variance, so the mean of the GP is set to zero. Therefore, the covariance function K defines the GP model. We will be using the squared exponential kernel for the covariance function: $K(x, x') = \exp\left(-\frac{(x-x')^2}{2l^2}\right)$, where l the length scale determines how strongly correlated close inputs are. The training data points be specified by the matrix X and the vector $\mathbf{y}$. Uncertainties on the i training observations can be included as:

$$y^{(i)} = f(x^{(i)}) + \varepsilon^{(i)}, \qquad i = 1, \ldots, m, \tag{13.3}$$

where ε is the noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, and the (i) superscript denotes one of the i training examples. Since the GP specifies a prior over functions, this length scale defining K is optimized by maximizing the log marginal likelihood on the training inputs. The log marginal likelihood is:

$$\begin{aligned} \log p(\mathbf{y}|X) &= \log \frac{1}{(2\pi)^{n/2}\left|K + \sigma^2 I\right|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{y}^T(K + \sigma^2 I)\mathbf{y}\right) \\ &= -\frac{1}{2}\mathbf{y}^T(K + \sigma^2 I)\mathbf{y} - \frac{1}{2}\log|K + \sigma^2 I| - \frac{n}{2}\log 2\pi, \end{aligned} \tag{13.4}$$

The term "marginal" means that the Gaussian distribution over the finite number of training data points doesn't change if a larger set of data points is considered [165]. Maximizing $\log p(y|X)$ gives the maximum likelihood estimator for the length scale, l.

The joint marginal over the training points and the test (or interpolation) points X_* and $\mathbf{y}_*$ also follows a multivariate Gaussian distribution:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \Big| X, X_* = \begin{bmatrix} f \\ f_* \end{bmatrix} + \begin{bmatrix} \vec{\varepsilon} \\ \vec{\varepsilon}* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*,) + \sigma^2 I \end{bmatrix}\right) \tag{13.5}$$

Marginalizing out the dependence over the training variables gives the "predictive posterior distribution" as the $(X_*, \mathbf{y}_*)$ also follows a Gaussian distribution, with the mean and covariance given by [165]:

$$\mu_* = K(X_*, X)(K(X,X) + \sigma^2 I)^{-1}\mathbf{y} \tag{13.6}$$

$$\Sigma_* = K(X_*, X_*) - K(X_*, X)(K(X,X) + \sigma^2 I)^{-1}K(X, X_*). \tag{13.7}$$

Instead of just having a point prediction at each query point, there is a distribution described by this multivariate Gaussian – which allows the model to also have a measure of its uncertainty.

13.1.2 Application for the $HH \to 4b$ analysis

The input to the GP is a 2d histogram of the massplane with a bounding box around the circle that defined the reweighting CRs, as shown by the dotted purple line in Figure 13.2(a). This box has an edge length of 90 GeV, and 25 bins for both the m_{H1} and m_{H2} axes. The (m_{H1}, m_{H2}) bin centers are used to predict the number of events in the bin. The Poisson errors on the bin entries are the uncertainties in the regression targets, and are included in the GP fit with the σ in Eq. 13.3. The targets are normalized to have zero mean and unit variance. Bins with any overlap with the SR are not included in the training process.

The GP is fit with scikit-learn [164] using the squared exponential kernel with two length scales for the m_{H1} and m_{H2} directions:

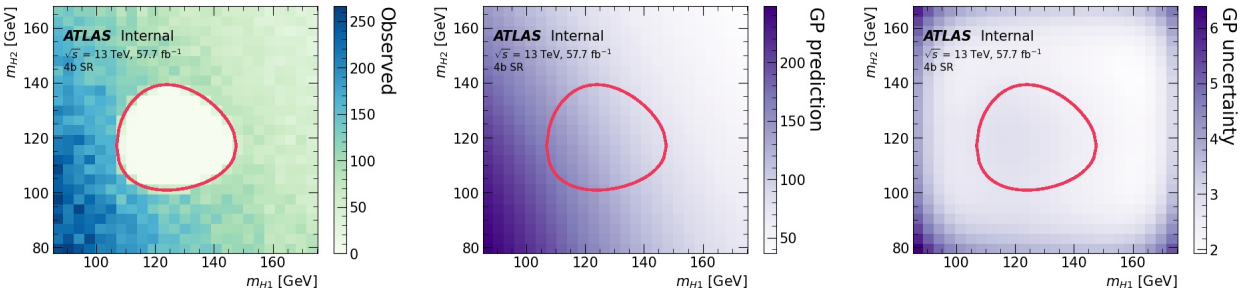
$$K\left(\begin{bmatrix} m_{H1} \\ m_{H2} \end{bmatrix}, \begin{bmatrix} m'_{H1} \\ m'_{H2} \end{bmatrix}\right) = \exp\left(-\frac{\left(m_{H1} - m'_{H1}\right)^2}{2l_1^2} - \frac{\left(m_{H2} - m'_{H2}\right)^2}{2l_2^2}\right). \tag{13.8}$$

To account for the differences in the triggers, a separate GP is fit for each year (2016, 2017, and 2018). Figure 13.2(b) shows the mean of predictive posterior for the GP fit to the blinded massplane. The predicted massplane gives a smoothed prediction of the input massplane, while also providing an interpolation for the bin entries into the SR since the fitted length scales (shown in Table 13.1) are larger than the radius of the SR. The GP predictions are constrained by nearby observed data points, so Figure 13.2(c) shows that the GP predicted error increases slightly as it interpolates into the SR. To quantify the error compared with the observed data, Figure 13.3 shows pulls defined by the GP mean prediction minus the observed yield, divided by the Poisson uncertainty on the observation (the square root of the bin yields). The pulls for these bins lying outside (purple) and inside (pink) the SR is shown in Figure 13.3(b). The mean and variance of these pulls in the SR are: $\mu = 0.08$, $\sigma = 0.93$ – which is close to zero mean and unit variance. This indicates the model is providing a good interpolation into the SR.

	m_{H1} [GeV]	m_{H2} [GeV]
2016	108.3	50.6
2017	106.1	64.4
2018	105.4	56.4

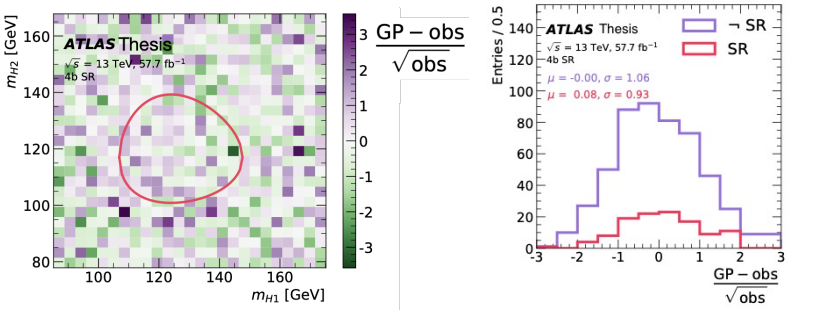
Table 13.1: The fitted length scales for 4b data.

Samples from the GP are used to condition the normalizing flow model's SR prediction. Inverse transform sampling [167] is used to draw the samples of the 2d histogram bin centers. The two-dimensional (m_{H1}, m_{H2}) GP prediction is reshaped to a 1d histogram which is used to construct the cumulative density function (CDF). Samples of $z \sim [0, 1]$ are mapped to the bins of the CDF to give samples of the (m_{H1}, m_{H2}) bin centers following the GP's $p(m_{H1}, m_{H2})$ probability. The (m_{H1}, m_{H2}) samples are then smeared uniformly with the bin width to give continuous predictions.



(a) observed data (b) GP fit (c) GP error

Figure 13.2: GP fits for the 2018 4b SR. All the massplane fits are before the X_{Wt} cut.



(a) (b)

Figure 13.3: Left: non-closure of the GP fits for the 2018 4b SR region. Right: 1d histogram of the pulls. The "¬ SR" ("not SR") purple line shows the bins used to fit the GP, while the "SR" pink line shows the bins that were blinded in the fit. These evaluation plots are before the X_{Wt} cut.

Since the number of samples drawn is arbitrary, the predicted SR event yields are rescaled using the observed and sampled entries from the interpolation bounding box with the blinded SR:

$$n_{SR} = n^{obs}_{\neg SR} \cdot \frac{n^{samples}_{SR}}{n^{samples}_{\neg SR}}. \tag{13.9}$$

13.2 Normalizing Flows

13.2.1 Intro

A normalizing flow is a generative model that transforms a known base density (such as a Gaussian) into a target density by a sequence of invertible transformations, here denoted f_i [168]. In the forward mode of the flow ($f = f_1 \circ \cdots \circ f_L$ in Figure 13.4) samples from the Gaussian are transformed to

give samples from the target distribution. Since each of the steps of the flow steps are invertible, the reverse mode of the flow f^{-1} lets us evaluate the probability of the samples using the chain rule of probability:

$$p_x(x) = p_z(z)\left|\frac{dz}{dx}\right| = p_z(f^{-1}(x))\left|\frac{df^{-1}}{dx}\right| \tag{13.10}$$

where $p_z(z)$ is the Gaussian base density, and x are the features we want to model. By estimating the probability of training data points, we can optimize the flow's parameters with maximum likelihood. We define a loss as the negative log-likelihood of the training data ($-\sum_i \log p_x(x_i)$) and minimize it by stochastic gradient descent.

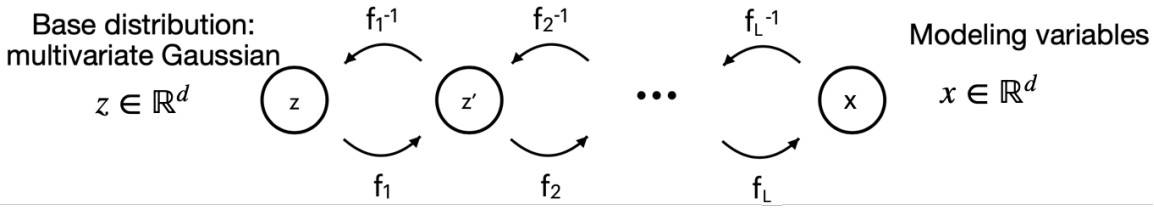


Figure 13.4: Visualization of the normalizing flow paradigm. The **forward mode** ($f = f_1 \circ \cdots \circ f_L$) generates samples. The **reverse mode** ($f^{-1} = f_L^{-1} \circ \cdots f_1^{-1}$) evaluates the density of a data point.

Since $x \in \mathbb{R}^d$, the f_i transformations need to keep both the Jacobian and the f_i^{-1} calculations tractable. Various methods for flow architectures to maximize the expressivity while preserving tractability are described in recent reviews [169, 170]. A trick is to use "coupling layers" which only transform half of the variables at a time. The Jacobian becomes block diagonal and the inverse operation goes from $\mathcal{O}(d^3)$ to $\mathcal{O}(d)$ since we just need to take the product over the diagonal entries. The RealNVP is a simple architecture with such coupling layers used in our early optimization studies, and described in Appendix G.1.

Here we show results for a rational quadratic neural spline flow (RQ-NSF) [171] with their `nflows` python package [172]. Since the only constraint needed to ensure the invertibility of f_i is that the f_i are monotonic functions, this method uses a monotonically increasing function parametrized by a spline with K bins between (B, B) as shown in Figure 13.5. Each "bin" on the domain is parametrized by a rational quadratic transformation given by:

$$f_{jk}(x_i) = \frac{a_{ijk}x_i^2 + b_{ijk}x_i + c_{ijk}}{d_{ijk}x_i^2 + e_{ijk}x_i + f_{ijk}}$$

where i is the dimension of the transforming variable, and j be the corresponding flow layer, and k is the bin. To predict one step of a single transforming variable involves $6K$ constants: $a_{ijk}, b_{ijk}, c_{ijk}, d_{ijk}, e_{ijk}, f_{ijk}$.

But applying the additional constraints:

- Boundary conditions:
  $(x_0, y_0) = (-B, -B)$ and $(x_K, y_K) = (B, B)$
- Continuity at the internal knots (which specify the bin boundaries):
  2(K-1) constraints
- Continuous derivatives at the knots:
  K-1 constraints

reduces the number of constants to
6K - [2(K-1) + 2K+1 + 2 + 2] = 3K - 1.

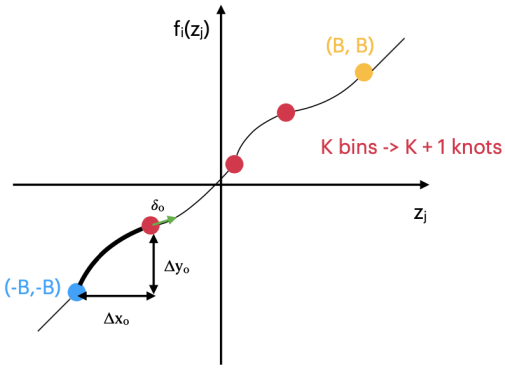


Figure 13.5: RQ-NSF parametrization

A NN then predicts the $2K$ widths and heights of the bins and the $K-1$ derivatives at the internal knots. The NN output is an unconstrained real vector in $\theta_i \in \mathbb{R}^{3K-1}$, and this the θ_i vector is partitioned into three pieces $[\theta_i^w, \theta_i^h, \theta_i^d]$. Since the spline characterizes a transformation from a domain $(-B, B)$ into a range $(-B, B)$, the θ_i^w and θ_i^h vectors are each passed through a softmax function (with a range (0,1)) and then multiplied by 2B to give the widths and heights of the shifts between the knot locations. Since the monotonicity of the f_j is crucial for its invertibility, the θ_i^d representing the knot derivatives are passed through a softplus[1] function to ensure the derivative stays positive. Outside of the range [-B,B] an identify transform is used. A batch norm layer [122] is used between each of the flow steps to keep the modeling variables in the range where the spline has its expressive power.

To allow mixing between the input variables, this architecture also uses a generalized permutation:

$$W = PLU, \tag{13.11}$$

where P is the permutation matrix, L is a lower triangular matrix, and U is an upper triangular matrix. The Jacobian stays tractable because $det(W)$ only takes $\mathcal{O}(d)$ time to compute. Inverting this step involves solving two triangular systems, which is a $\mathcal{O}(d^2)$ operation, which is the same order as inverting the spline transformations.

To solve the interpolation problem, we predict the conditional probability distribution (as shown in [171]) by passing (m_{h1}, m_{h2}) as additional variables to the NNs predicting the spline constants. This preserves the nice properties of the bijection for each of the flow steps while keeping the Jacobians tractable. The flow is trained inclusively over the years (2016, 2017, and 2018), and the year is passed as an additional input.

[1] Softplus$(x) = \log(1 + \exp(x))$: a smooth approximation to the function that ensures positive outputs [173].

13.2.2 Implementation details

We found that it was easier to use the flow to model the Higgs Candidates' 3-momentum instead of modeling m_{HH} and $\Delta\eta_{HH}$ directly. The top row of Figure 13.6 shows the HC kinematics. We predict the log of the HC p_{T}s instead of the HC p_{T}s directly since this gives a bell-shaped distribution which is easier to model for the flow which transforms a base Gaussian distribution. In Figure 13.6(c), the HC ϕs are shown. If the HC ϕ were predicted directly, then the flow would not be able to tell that $\phi = \pi$ and $\phi = -\pi$ correspond to the same event. But the boundary condition is not the only aspect that we need to encode as the problem can be further simplified with the azimuthal symmetry in our events. We don't need to predict both ϕ_{H1} and ϕ_{H2}, but just the difference $\Delta\phi_{HH}$ as both m_{HH} and $\Delta\eta_{HH}$ are symmetric with respect to azimuthal rotations of the event. Figure 13.6(d) shows $\Delta\phi_{HH}$, which is peaked at π for the frequently back-to-back the di-jet pairs. The sharp cutoff would be very hard for the flow to predict with the smooth transformations from a base Gaussian distribution. We instead consider $\pi - \Delta\phi_{HH}$, to get a steeply falling distribution, and then take the log to model $\log(\pi - \Delta\phi_{HH})$. This transformation is more Gaussian-like, as shown in Figure 13.6(e). This transformation also encodes a physicality requirement that the predicted $\Delta\phi_{HH}$ are always less than π. There's no corresponding guarantee that the predicted $\Delta\phi_{HH} > 0$, but since this is in the tail of the distribution, the flow predicts a negligible number of negative $\Delta\phi_{HH}$ values (see Figure 13.10(f)). All modeling variables are normalized to have zero mean and unit variance.

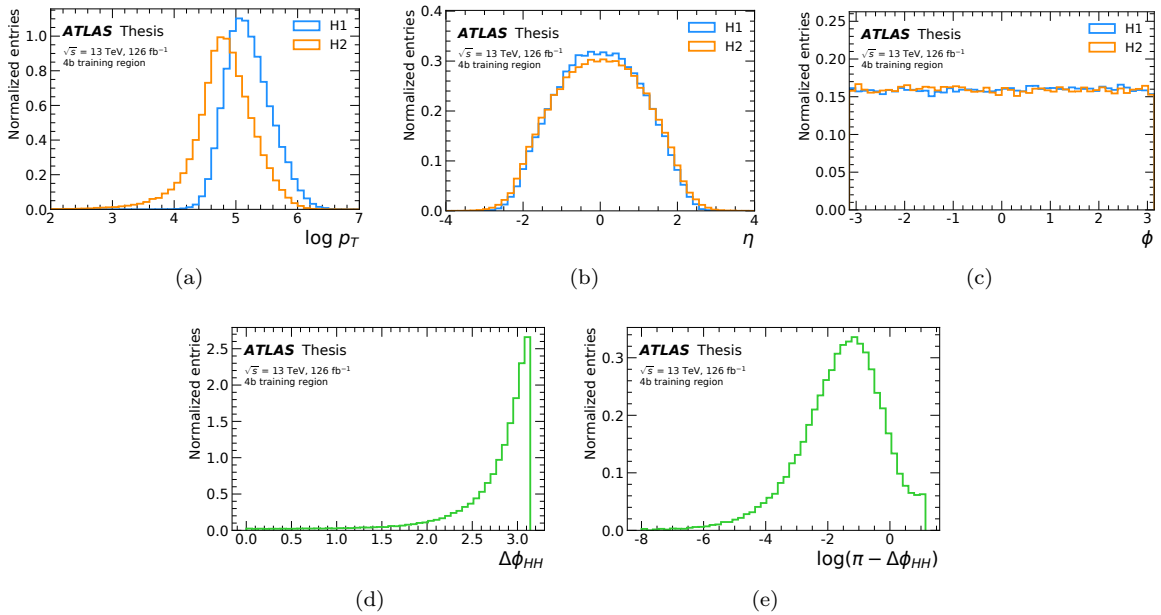


Figure 13.6: Motivation for Higgs Candidate variables modeled by the flow. The flow models the Higgs Candidates' p_{T}s (a) and ηs (b), the $\log(\pi - \Delta\phi_{HH})$ (e), and X_{Wt} (not shown here).

The a smooth massplane was a key modeling assumption, but the X_{Wt} cut induces structure in the massplane (see Figure 10.18). Thus, the flow is trained before applying the X_{Wt} cut, but X_{Wt} is passed as an additional modeling variable. When evaluating the SR predictions, the sampled events that have $X_{Wt} < 1.5$ are removed to compare to the reweighting background estimate.

We optimized the hyperparameters for this problem by looking at the modeling in the 2b SR, with Appendix G.2.3 giving the hyperparameter scan considered. We use a neural spline flow with 10 layers (or 10 f_i spline transformation steps). Each spline used $K = 4$ bins, and the spline transform is defined on the domain $(-B, B) = (-3, 3)$. The ResNets [174] are used to predict the spline constants. We use only needed a single ResNets block with 32 hidden units, and a dropout fraction of 10% [**dropout**] To avoid overfitting, we found that using an L2 regularization with $\beta = 1e-6$ was helpful, so the loss function is the sum of two terms:

$$\mathcal{L} = -\frac{1}{N} \sum_{x \sim \neg \text{SR}} \log p(x | m_{H1}, m_{H2}, \text{yr}) + \beta \sum_j w_j^2, \tag{13.12}$$

where the first term is the negative log-likelihood of the training data, and the second term is the L2 regularization for the w_j NN weights.

The flow was trained inclusively on all of the years using the data events in the interpolation bounding box outside of the SR conditioning on m_{H1}, m_{H2} and the year. We trained with the adam optimizer with a learning rate of 10^{-3}. Each flow is trained 25 times to assess an uncertainty from the variation in the NN trainings [125].

Figure 13.7 shows how flow proceeds in 10 steps to gradually transform the base Gaussian density into the Higgs candidate variables in the 4b SR.

Figure 13.8 visualizes the multi-dimensional density that the flow has learned by showing all of the $2d$ correlations. The grey scatter plots in the lower left triangular block show the correlations in the 4b SR data, while the pink scatter plots in the upper right triangular block show the analogous correlations predicted by the flow samples. The histograms along the diagonal compare the one-dimensional marginals. In the blue box the η_{H1}, η_{H2} correlation is emphasizes that the flow has learned to predict the $\Delta\eta_{HH} < 1.5$ cut.

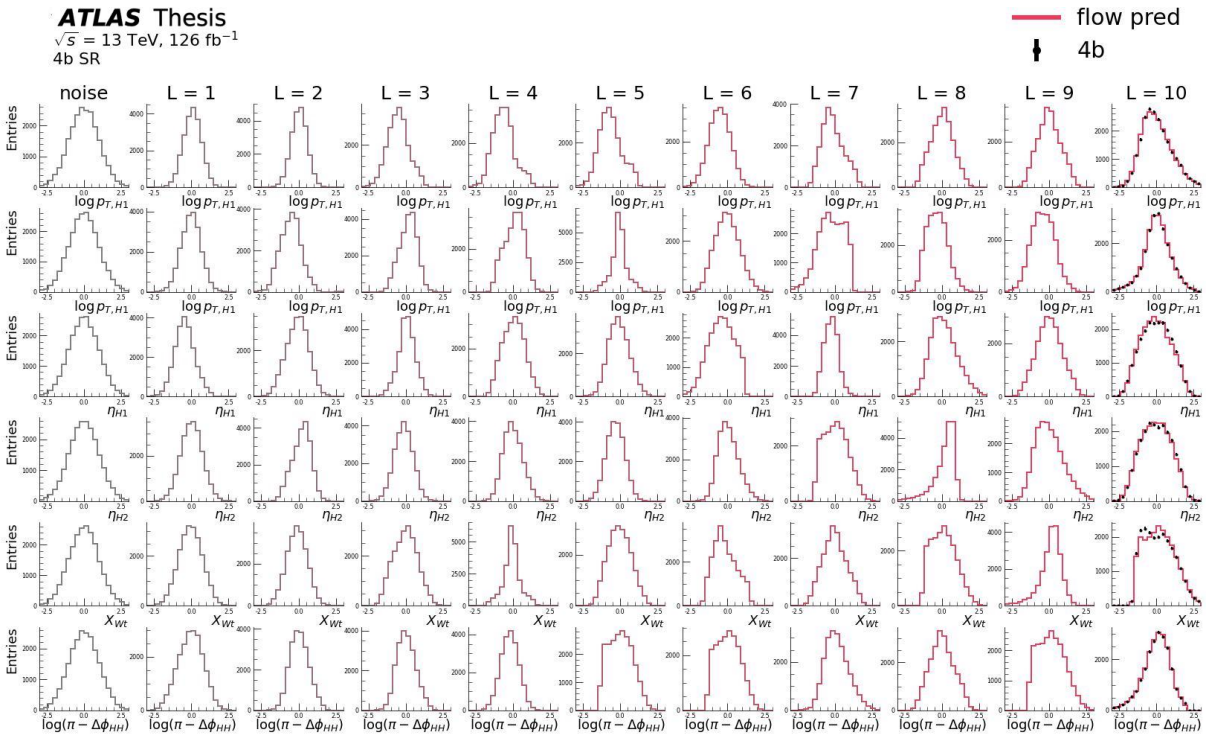


Figure 13.7: Demonstration of how the flow transforms unstructured noise into a structured prediction in the 4b SR. These samples are conditioned on the SR (m_{H1}, m_{H2}, yr) data from the 2016, 2017 and 2018 datasets. The grey histograms in the left column are samples from a 6d Gaussian. Each column to the right shows the transformation from one layer of the flow (i.e, a single invertible transformation), with the right-most column showing the prediction of the final flow. This is a single flow training, with the distributions are shown before applying the X_{Wt}. Variables have been scaled to zero mean and unit variance.

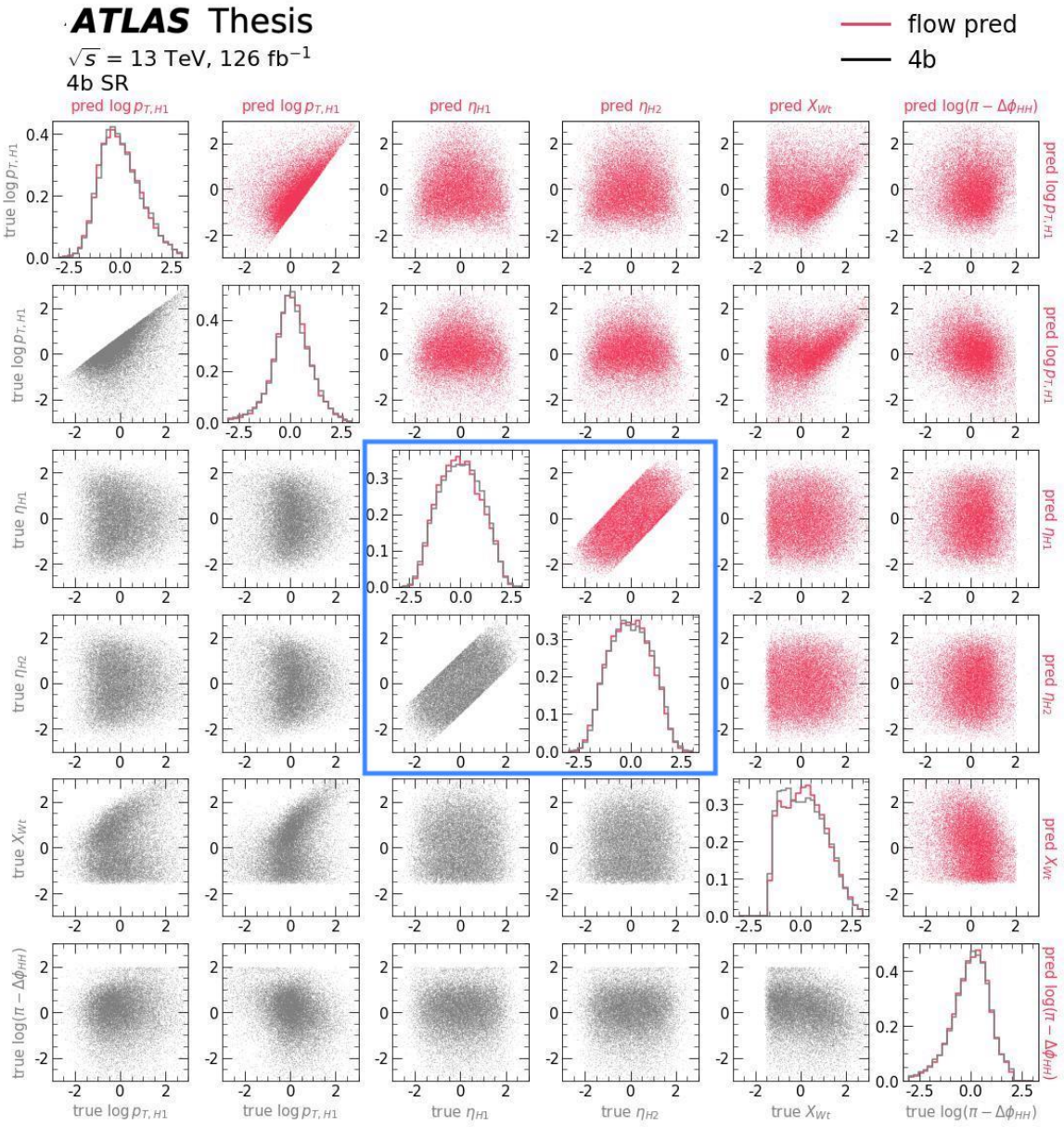


Figure 13.8: The correlation between the modeling variables for the SR data (grey) and the flow prediction (pink) in the 4b SR. The blue box emphasizes the η_{H1}, η_{H2} variables to show how the flow has learnt the $\Delta\eta_{HH} < 1.5$ cut. The marginal plots along the diagonal are normalized to unity. These samples are conditioned on the SR $(m_{H1}, m_{H2}, \text{yr})$ data from the 2016, 2017 and 2018 datasets. This is a single flow training, with the distributions shown before applying the X_{Wt} cut. Variables are scaled to zero mean and unit variance.

After predicting the vector $x = \left(\log p_{T,H1}, \log p_{T,H2}, \eta_{H1}, \eta_{H2} \log(\pi - \Delta\phi_{HH}), X_{Wt}\right)$ from the flow, we combine this with the m_{H1}, m_{H2} samples from the GP to reconstruct the HC 4-vectors:

$$p_{H1} = (p_{T,H1}, \eta_{H1}, 0, m_{H1})$$
$$p_{H2} = (p_{T,H2}, \eta_{H2}, \Delta\phi_{HH}, m_{H2}).$$

These HC 4-vectors are then summed to get the m_{HH}, and we also calculate $\Delta\eta_{HH} = |\eta_{H1} - \eta_{H2}|$.

Figure 13.9 is a summary of the background model presented in the previous two sections. In evaluating the background model, 100,000 events are sampled from the interpolation box for each of the GP massplanes for each year: $p_{16}(m_{H1}, m_{H2})$, $p_{17}(m_{H1}, m_{H2})$, $p_{18}(m_{H1}, m_{H2})$, where the subscripts denote the 2016, 2017, and 2018 GP fits, respectively (step 1a in Figure 13.9) . These samples set the SR normalization (step 1b and Eq. 13.9), and are used to condition the flow samples drawn in the SR (step 2). Although by modeling probability distributions both the GP and the flow quantify an uncertainty, for the $4b$ background modeling the dominant uncertainty was the Deep Ensembles uncertainty from the random initialization and optimization of the NNs ([125] and Table 12.3). For the following results, we show the NSF flow model trained 25 times. The same m_{H1}, m_{H2} samples from the GP SR are used to evaluate the flow prediction, and we cite the standard deviation of these 25 flow models as the interpolation error bar.

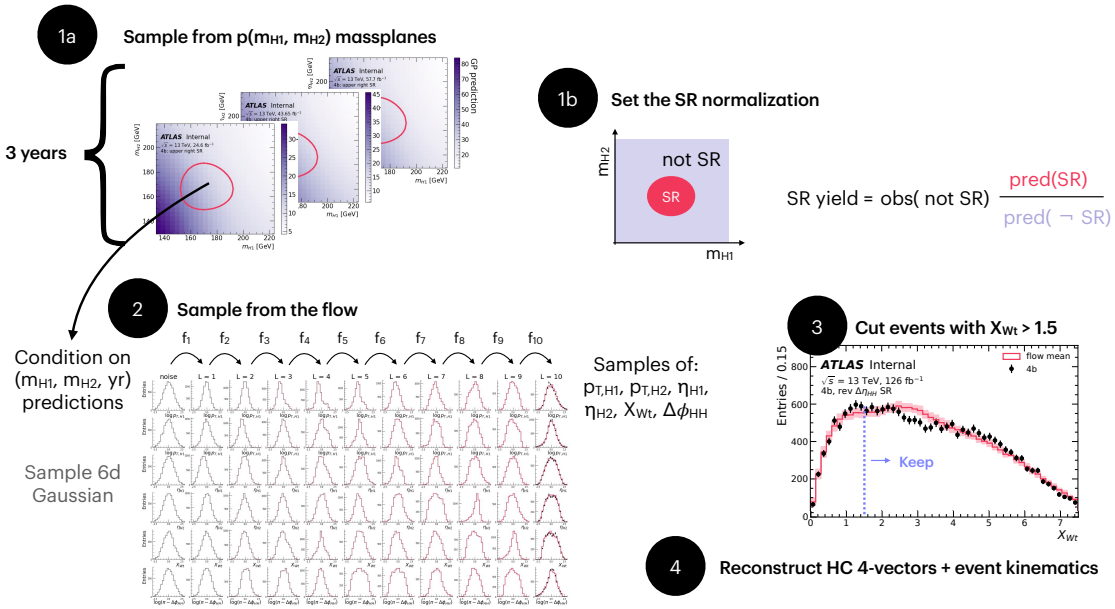


Figure 13.9: Demonstration of the background prediction algorithm for the upper right SR.

Comparisons are made to the reweighting method, and the reweighting error bars include the

deep ensembles error, the error on the CR1 / CR2 shape difference systematic, and the $2b$ statistical uncertainty.

13.3 Results

Figure 13.10 shows the six variables that the flow models and compares to the reweighting prediction. For the HC p_{T}s, the flow is doing a better job modeling the peak of the distribution. Through careful flow input optimizations, we have also learned non-trivial distributions, such as $\Delta\phi_{HH}$ in Figure 13.10(f). The interpolation also is better at modeling the decreasing X_{Wt} distribution in the range $X_{Wt} = (3, 6)$.

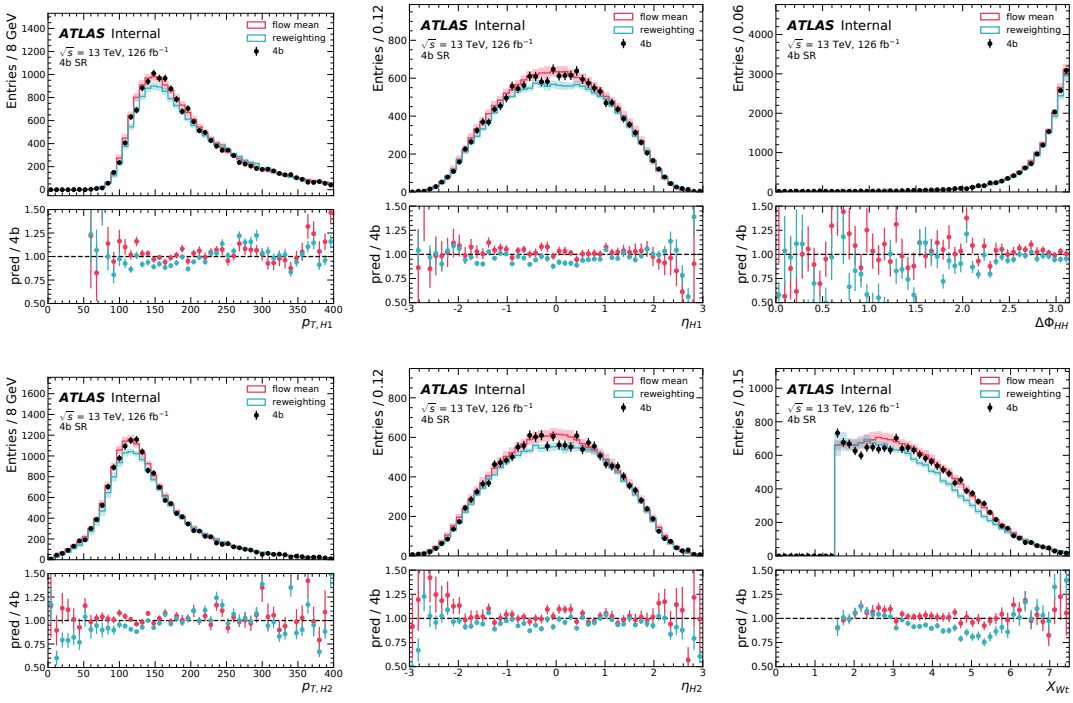


Figure 13.10: The flow training variables in the 4b SR (after the X_{Wt} cut).

Figure 13.11 shows the modeling for for the high level variables reconstructed from the HC kinematics. For both m_{HH} and $p_{T,HH}$ we see a better modeling in the peaks of these distributions. For $\Delta\eta_{HH}$, the flow sometimes predicts events with $\Delta\eta_{HH} > 1.5$, and in Figure 13.11(c) these overflow events are included in the highest $\Delta\eta_{HH}$ bin.

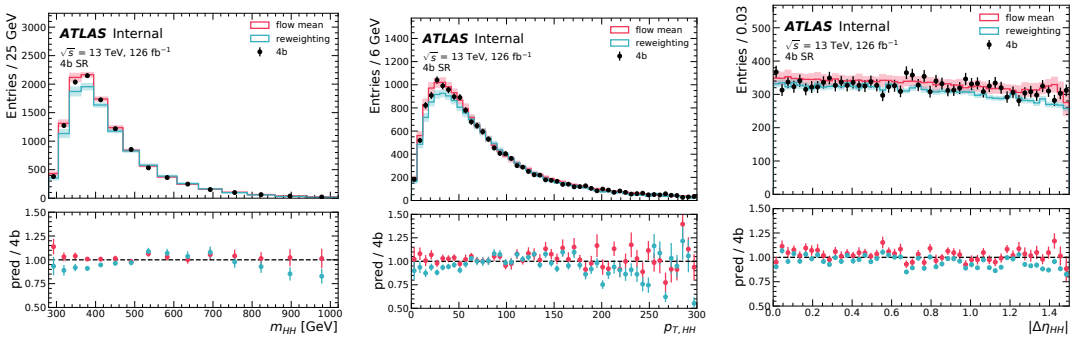


Figure 13.11: High level variables reconstructed from HC kinematics in the 4b SR (after the X_{Wt} cut).

Finally, Figure 13.12 shows m_{HH} with the $\Delta\eta_{HH}$ and X_{HH} categories used as the discriminating variable for setting the $4b$ analysis limits. Events in the overflow $\Delta\eta_{HH}$, m_{HH} bins are included in the highest bins here again. The flow again compares favorably here – making this a promising method to use for the Run 3 iteration of the $4b$ analysis.

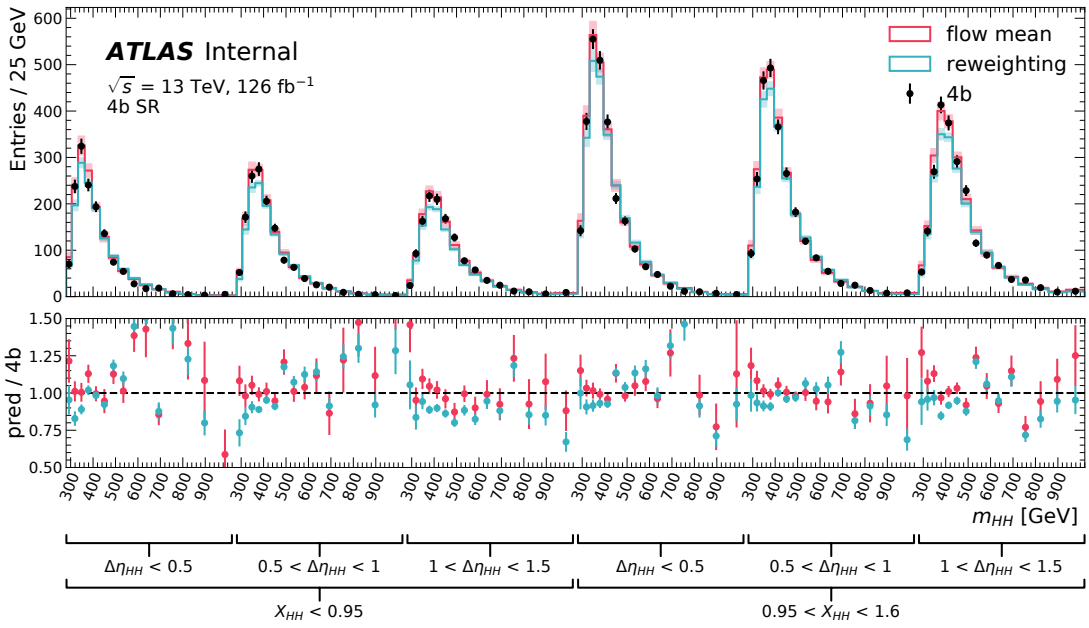


Figure 13.12: High dimensional discriminant in the 4b SR, after the X_{Wt} cut.

Results for other background validation regimes

Since one of the achievements of the recent $4b$ result was the tour de force effforts for testing our procedure in a suite of background validation regions, for the interpolation we also evaluated our procedure on this same set of regions: 3b1f, reversed $\Delta\eta_{HH}$, and the shifted SRs. Figure 13.13 shows the shifted SRs considered for the interpolation comparisons. Here we additionally also show results for the challenging "lower left" SR.

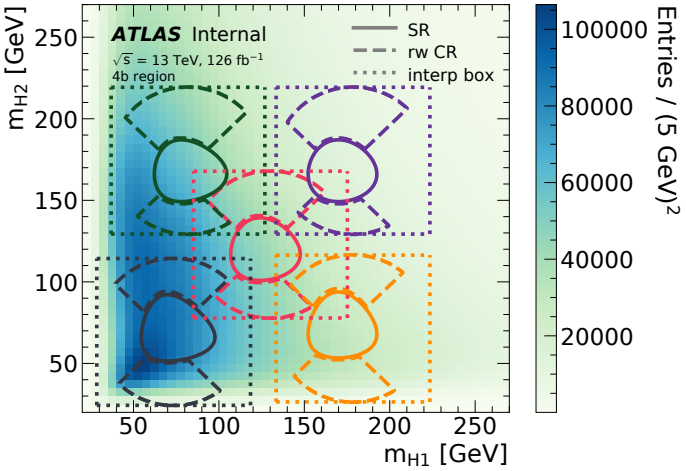


Figure 13.13: Illustration of the interpolation regimes. The pink (solid) circle shows the nominal 4b SR, with the pink dotted line showing the interpolation bounding box. The quadrants used to define the reweighing are also shown in the pink dashed crescents. The shifted regions used as validation tests of the method are shown in the blue, orange, green, and purple overlays.

Table 13.2 shows the observed yield, the reweighting, and the interpolation predictions. The flow enters in the interpolation's yield prediction by modeling X_{Wt} as these results are shown after X_{Wt} cut. In pink are highlighted the regions the flow is more accurate at predicting the normalization, while the turquoise indicates the regions the reweighting does better.

	obs	rw	flow	1 - rw / obs [%]	1 - flow / obs [%]
lower left	40578	48708.9	39252.2	-20.0	3.3
lower right	12377	14648.5	11982.7	-18.4	3.2
upper right	5751	5543.0	5825.9	3.6	-1.3
upper left	19075	19504.7	19833.4	-2.3	-4.0
3b1f	180044	175817.9	175416.8	2.3	2.6
rev $\Delta\eta_{HH}$	16113	16462.7	16185.9	-2.2	-0.5
4b	16171	15423.7	16564.8	4.6	-2.4

Table 13.2: Yields for the predictions after applying the X_{Wt} cut

The m_{HH} modeling for each of these validation regions is also shown in Figure 13.14. Figure 13.14(a) shows the lower left SR. In the reweighting validation, we excluded this from the reweighting tests because we believed that it was more challenging than the $4b$ SR – as Figure 13.13 shows that the lower left CR1 (in the dashed navy crescents) overlaps with the kinematic turn-on in the massplane. It is more difficult to derive a reliable background estimate when the underlying control region does not smoothly vary into the SR, evidenced in Figure 13.14(a) as the non-closure is much larger than the assessed error. The interpolation model, however, is showing a %-level error for the non-closure in this challenging extrapolation regime.

The lower right SR in Figure 13.14(b) also has a larger non-closure for the reweighting compared to the interpolation. In this region, we understood that the reweighting did have a good closure for CR2. Since the error bar on the reweighting includes the variation from the alternative CR2 model, we can see that the CR2 provides a good model as the lower edge of the reweighting error bar is close to the observed data. There is still a benefit with the interpolation not needing to assess as large of an error bar.

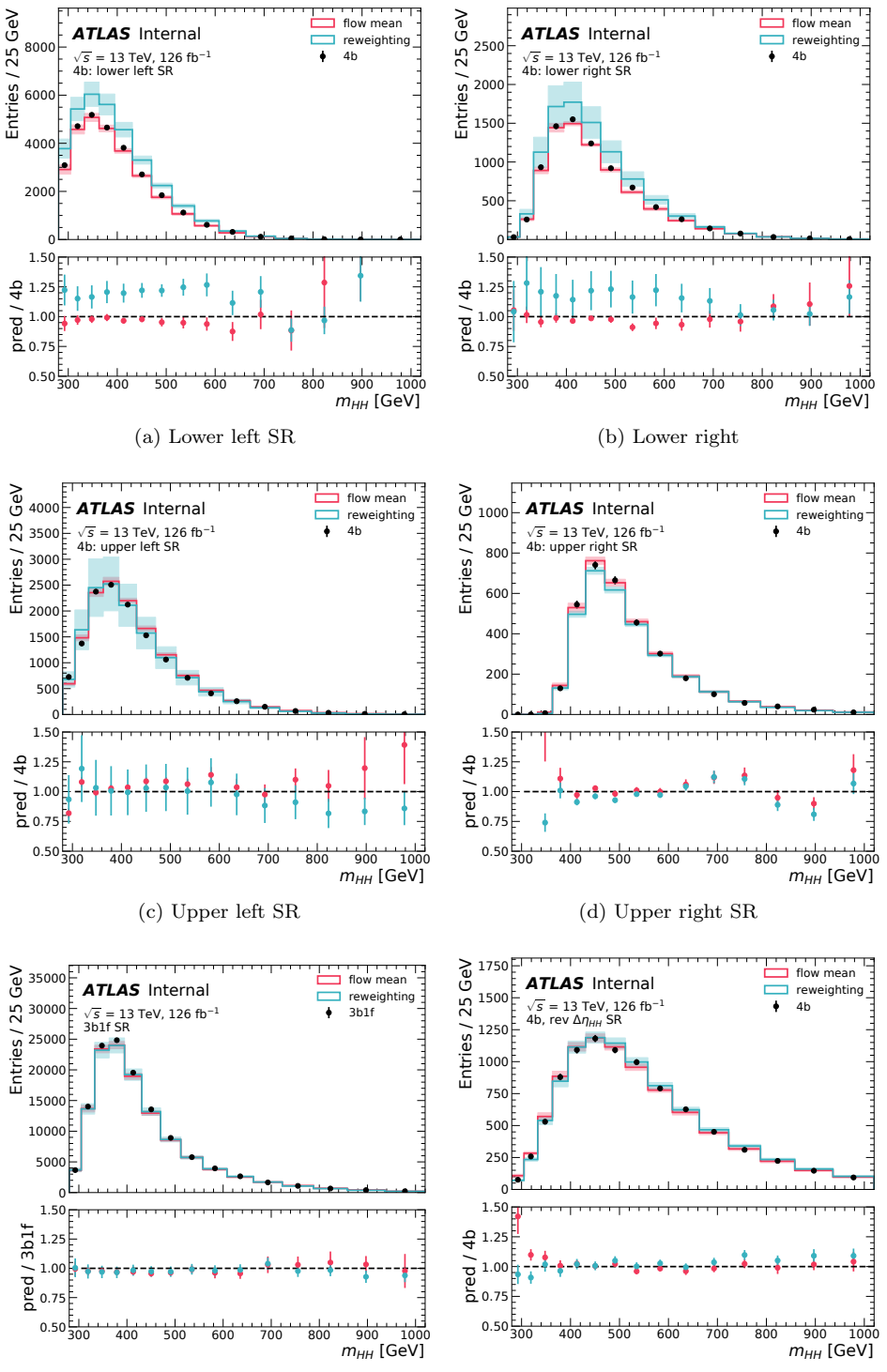
For the upper left SR in Figure 13.14(d), the reweighting has a better normalization prediction than the interpolation, but since the difference between CR1 and CR2 is quite large for this region, this is reflected in a correspondingly larger error bar for the reweighting.

The reweighting also has a slightly better normalization in the 3b1f region, but Figure 13.14(e) shows that both of these models give a good background model.

Finally, in Table 13.3 the χ^2 for the shapes of the kinematics are compared for the other validation regions. The prediction histograms are normalized to the observed event yield. The same highlighting scheme is used here as earlier, and overall the flow is showing good modeling iin these regions with respect to the strong baseline provided by the reweighting.

	lower left		lower right		upper right		upper left		3b1f		rev $\Delta\eta_{HH}$		4b	
	rw	flow	rw	flow	rw	flow	rw	flow	rw	flow	rw	flow	rw	flow
3d disc	5.52	2.19	1.32	0.98	1.35	0.93	2.54	1.60	1.85	2.24	1.37	1.25	2.14	1.34
m_{HH}	2.70	1.98	1.26	0.78	1.30	0.91	4.44	4.70	2.29	3.77	1.34	1.60	3.75	0.73
$\Delta\eta_{HH}$	6.95	1.37	1.52	1.07	1.51	1.63	1.99	0.98	1.38	1.94	1.16	3.90	1.19	0.99
$p_{T,H1}$	1.76	0.95	0.70	0.58	1.07	1.14	2.85	1.77	1.61	1.58	1.42	1.48	2.48	1.88
$p_{T,H2}$	2.81	1.83	2.83	1.17	1.82	1.06	0.92	1.49	2.64	2.50	1.74	1.51	2.24	1.00
η_{H1}	2.33	1.54	2.24	0.87	1.45	0.82	2.36	1.74	1.99	2.07	2.74	1.51	1.13	1.25
η_{H2}	2.62	1.35	4.93	1.11	0.85	0.75	1.33	1.16	1.37	1.36	2.07	1.17	1.26	1.16
$\Delta\phi_{HH}$	9.64	1.46	18.81	1.04	3.23	2.61	2.24	1.40	1.88	1.85	2.08	1.16	2.63	1.08
X_{Wt}	17.64	1.27	5.29	0.79	1.39	1.45	1.33	0.82	2.49	3.73	1.74	1.71	3.10	1.01
$p_{T,HH}$	6.06	1.13	1.76	1.00	1.50	1.23	1.80	0.80	2.54	0.93	2.35	0.87	2.00	0.79

Table 13.3: The χ^2 / (number of bins - 1) for the histograms after applying the X_{Wt} cut. In the top row, "3d discriminant" refers to the m_{HH} distribution with the $\Delta\eta_{HH}$ and X_{HH} categories.



(a) Lower left SR

(b) Lower right

(c) Upper left SR

(d) Upper right SR

(e) 3b1f SR

(f) reversed $\Delta\eta_{HH}$ SR

Figure 13.14: m_{HH} for the background validation regions, shown after the X_{Wt} cut.

Outlook

This new paradigm for a robust background model is encouraging for the future of the $4b$ analysis for Run 3 and beyond. Below As the interpolation models the (m_{H1}, m_{H2}) dependence, this removes the need for the CR12 extrapolation systematic which is currently one of our limiting analysis uncertainties (Table 12.3). The validation regions studied above could provide an additional systematic for the interpolation method when propagating into an analysis. Another benefit of the interpolation is that we can draw an arbitrarily large number of samples to construct these background templates. This is important as the CMS $4b$ resolved analysis is currently limited by the statistical precision of their background templates [175]. Furthermore, alleviating the necessity of reweighting from a $2b$ region also helps the HL-LHC prospects where the higher rates make it impossible to collect a $2b$ sample without draconian trigger thresholds. Finally, a more robust background model might allow us to use ML-methods (i.e, NNs) for signal versus background discrimination to improve our sensitivity in the future. These avenues of exploration that open up promise to make the future of $HH \to 4b$ quite bbbbright.